Dynamic Sell-side Revshare on AdX

Date: May 9, 2014
Authors: David Pal <dpal@google.com>, Max Lin <whlin@google.com>
With inputs from jimgiles@, mpal@, nirmaljayaram@
go/adx-ss-revshare
Tracking bug: b/14681323

Goal

We would like to allow dynamic sell-side revshare on AdX so that transactions that would have not been cleared because of fixed sell-side revshare can be cleared. This is for increasing matching rate on AdX and revenue. The expected revenue impact is ~$128MM with overall AdX margin at 19%. See Nirmal's original doc for more analysis.

This document describes the AdX serving change for running dynamic sell-side revshare experiments.

Background

Currently AdX takes a fixed 20% revshare from transactions for all buyers (AdWords or RTB buyers). The 20% AdX revshare is called **sell-side** revshare, in contrast to additional 14% **buy-side** revshare Google take for AdWords buyers. Dynamic buy-side revshare has been launched with positive outcome. This launch focuses on sell-side revshare.

[redacted]

PTX1040
1:23-cv-00108

Suppose there are two bids from AdWords (with their buy-side revshare already taken out) and one bid from RTB buyers.

Original bids enter AdX: AW1 $2, AW2 $1, RTB $1.5
Take out 20% sell-side revshare: AW $1.6, AW2 $0.8, RTB $1.2

If the minimum reserve price is $1.8, all bids are below the reserve price and no ad is matched. However, if AdX is willing to take a lower sell-side revshare, we can clear the transaction at the reserve price and pays publisher $1.8, and charges advertiser $2, resulting in effectively 10% sell-side revshare (= 1 - 1.8/2).

[redacted]

We can maintain a desired sell-side minimal margin by probabilistic throttling. Buyers who abuse the dynamic rev-share can be throttled with high probability, that is, most of the time their queries are transacted using fixed minimal sell-side revshare. In the above example, when throttled, we will not clear the transaction if minimal reserve price is higher than $1.8.

AdX Auction with Dynamic Sell-side Revshare

We illustrate where dynamic sell-side revshare differs from existing fixed sell-side revshare in AdX auction in two places: filtering bids for reserve price, and choose clearing price.

	AdWord 1	AdWord 2	RTB
Bid entering AdX	2	1	1.4
In **fixed sell-side revshare** (20%), bids are taken out revshare before comparing with reserve price.	1.6	0.8	1.12
Auction outcome with reserve price of 1.5 under **fixed sell-side revshare**	Winner (the clearing price 1.5, publisher gets 1.5, advertiser pays 1.88, AdX takes 20% margin)	FILTERED	FILTERED
Auction outcome with reserve price of 1.7 under **fixed sell-side revshare**	FILTERED	FILTERED	FILTERD
In **dynamic sell-side revshare** , bids are NOT taken out revshare before comparing with reserve price.	2	1	1.4
Auction outcome with reserve price of 1.5 under **dynamic sell-side revshare**	Winner (the clearing price is 1.5, publisher gets 1.5, advertiser pays 1.88, AdX takes 20% margin)	FILTERED	FILTERED
Auction outcome with reserve price of 1.7 under **dynamic sell-side revshare**	Winner (the clearing price is 1.7, publisher gets 1.7, advertiser pays 2, AdX takes 15% margin)	FILTERED	FILTERED
Auction outcome with reserve price 1.1 under **dynamic sell-side revshare**	Winner (the clearing price is 1.12, publisher gets 1.12, advertiser pays 1.4, AdX takes 20% margin)	FILTERED	SECOND BID

	AdWord 1	AdWord 2	RTB
Bid entering AdX	2	1	1.9
In **dynamic sell-side revshare** , bids are NOT taken out revshare before comparing with reserve price.	2	1	1.9
Auction outcome with reserve price 1.1 under **dynamic sell-side revshare**	Winner (the clearing is 1.52, publisher gets 1.52, advertiser pays 1.9, and AdX takes 20% margin).	FILTERED	SECOND BID

When a request from AdX seller comes, AdX solicits bids from AdWords and RTB buyers.

AdWord bid: $2
AdWord bid 2: $1
RTB bid: $1.4

Post-revshare bids below minimum reserve price are filtered before auction. With **fixed sell-side revshare** (e.g., 20%), all bids are taken out sell-side revshare and compare with minimum reserve price.

Post-revshare AdWord bid 1: $1.6
Post-revshare AdWord bid 2: $0.8
Post-revshare RTB bid: $1.12

If the minimum reserve price is $1.5, top AdWord bid can enter the auction. The clearing price is $1.5, AdX pays the publisher $1.5 and charges the buyer $1.875 (=1.5 / 0.8), taking fixed 20% margin (= 1 - 1.5 / 1.875).

If the minimum reserve price is $1.7, all bids are removed before auction and no transaction happens. However, we may clear the transaction if AdX is willing to take less than 20% sell-side revshare.

With **dynamic sell-side revshare**, we will assume sell-side revshare may go as low as zero, and will filter bids effectively based on original bids.

Post-revshare AdWord bid 1: $2
Post-revshare AdWord bid 2: $1
Post-revshare RTB bid: $1.4

If the minimum reserve price is $1.5, top AdWord bid enters the auction. The clearing price is $1.5, AdX pays the publisher $1.5 and charges the buyer $1.875 (= 1.5 / 0.8), taking fixed 20% margin (= 1 - 1.5 / 1.875). The outcome of the auction, running with 0% sell-side revshare, is the same as the outcome from the fixed 20% sell-side revshare.

If the minimum reserve price is $1.7, top AdWord bid enters the auction. The clearing price is $1.7, AdX pays the publisher $1.7 and charges the buyer $2, taking dynamic 15% margin (= 1 - 1.7 / 2). In practice, after winner is chosen, we compare the effective revshare based on payout and bid and maximum sell-side revshare (20%), and choose the small one as the final sell-side revshare.

Cases

[redacted]

Programmatic Reservation and Fixed CPM deals

Fixed CPM deals (preferred deals, private auctions) are not supported by dynamic sell-side revshare yet. Their auction outcome will not be changed after dynamic sell-side revshare is enabled for second priced candidates.

	First placed bid	Second placed bid	third-party reserve price
Pre sell-side revshare (original bids)	2 (programmatic reservation)	1 (open auction)	2.125
Post sell-side revshare	1.9 (= 2 * 0.95)	0.8 (= 1 * 0.8)	1.7 (= 2.125 * 0.8)

We clear at the programmatic reservation bid, charge advertiser $2, and take default 5% margin. The auction outcome is the same as that when dynamic sell-side revshare is not enabled.

	First placed bid	Second placed bid	third-party reserve price
Pre sell-side revshare (original bids)	1.75 (programmatic reservation)	1 (open auction)	2.125
Post sell-side revshare	1.66 (= 1.75 * 0.95)	0.8 (= 1 * 0.8)	1.7 (= 2.125 * 0.8)

There is no winner because highest bid is below third-party reserve (1.66 < 1.7). The auction outcome is the same as that when dynamic sell-side revshare is not enabled.

In the future, we may choose to enable dynamic revshare for fixed cpm deals, and clear at the third party reserve price, charge advertiser 1.75, and takes 3% (< usual 5% margin).

[redacted]

Self-bought Inventory

Self-bought ad (buyer and seller are the same entity, that is, publisher) takes a smaller than usual sell side revshare. The underlying implementation is done through buy-side revshare. The buy-side revshare times sell-side revshare is the desired revshare for self-bought. Therefore, self-bought inventory is a special case of regular second priced ad with fixed buy-side revshare.

	First placed bid	Second placed bid	third-party reserve price
Pre sell-side revshare (original bids)	2 (self-bought)	1 (open auction)	2.125
Post sell-side revshare	1.98 (= 2 * 0.99)		1.7 (= 2.125 * 0.8)

We clear at the third-party reserve price, charge advertiser $1.72, and takes default 1% margin. The outcome is the same as that with dynamic sell-side revshare disabled.

	First placed bid	Second placed bid	third-party reserve price
Pre sell-side revshare (original bids)	2 (self-bought)	1 (open auction)	2.49
Post sell-side revshare	1.98 (= 2 * 0.99)		1.99 (= 2.49 * 0.8)

This example will not be cleared with fixed sell-side revshare (1.98 < 1.99). However, with dynamic sell-side revshare, we will clear at third-party reserve price, charge advertiser $2, with 0.4% margin (< regular 1% margin).

Passback Chains

[redacted]

AdX auction supports auction with passback tags and can return a sequence of winners instead of one winner ad. We do not support dynamic sell-side revshare for passback chains yet and need to resolve the following two design points:

- There are multiple ad winners with different revshare (for example, publishers take more revshare from managed tags than AdWords ads). We need to decide how to adjust revshare for each ad in the chain: should AdX take less revshare uniformly across the chain, proportionally (publishers takes less revshare from ads with larger social value), or sequentially (exhausting revshare from first ad before following ads)?
- The sell-side revshare for managed tag candidates are implemented through the combination of buy-side and sell-side revshare. This is different from regular second

priced candidates that sell-side revshare is only through sell-side revshare. This breaks the assumption that we can take zero sell-side revshare before sorting by setting sell-side revshare to 1.0.

AdX Serving Changes

The key change in the AdX serving stack is to allow sell-side revshare on the AdX vary and control behind an experimental flag.

This table summarizes the auction changes, in terms of revshares, before and after dynamic revshare is implemented. Suppose the highest bid is $2 with no buy-side revshare with a third-party reserve price at $1.5:

	Bids	Third-party reserve price	Pricing
Fixed revshare	**Post** buy and sell side revshare, e.g., $2 * 0.8 = $1.6	**Post** revshare, e.g., $1.5 * 0.8 = $1.2	Clear at post-revshare third-party reserve price $1.2. Pay the publisher $1.2
Dynamic revshare	Post buy side revshare but pre sell-side revshare, e.g., $2.	Pre revshare, e.g., $1.5.	Clear at pre-revshare third-party reserve price $1.5. Take out 20% margin and pay the publisher $1.2.

Sell-side revshare

Right before AdX auction begins, sell-side revenue share is set based on the WebPropertyPageSettings protobuf constructed in the supermixer. When calling CreateRevShareInfo(), we will pass in a "minimal sell-side revshare" (value coming from the experimental flag) to the RevshareInfo object. This minimal sell-side revshare controls revshare for all buyers on AdX, including both AdWords and RTB buyers. The flag setting and experimental state are already available from supermixer's auction flow.

When ComputeRevenueShare() is called during auction to calculate revenue share, we will use the minimal sell-side revenue (say, 10%) instead of regular sell-side revshare (20%) and times the buy-side revenue share to return the final revshare. We will do this only for open auction (that is, transaction type is SECOND_PRICE_AUCTION). The sell-side revshare for other types of transactions (e.g., deals) is not changed.

AdX Simulation

Due to the newly added minimal sell-side revshare, we will need to record the minimal sell-side reveshare so that forecasting simulation can continue to simulate auction faithfully. We will record the minimal sell-side revshare in the RevshareInfo::Serialize() function.

Reserve Price seen by AdX Buyers

With reduced sell-side revshare, it's possible for AdX to clear a transaction when AdX buyer's bid, post regular sell-side revshare, is lower than reserve price. For example, AdX buyers may start to see sometimes they can clear $1 reserve price call with $1.1 bid instead of at least $1.25 (20% sell-side revshare). See three possible options in Nirmal's doc. We will run experiments to evaluate these options.

Launch Plan

Before full launch we will evaluate the change offline and online.

For offline analysis, after serving changes are complete, we will run simulators with different minimal sell-side revshare (e.g., 0%, and 10%), and confirm the revenue impact from simulations.

For online experiments, we will start with low-percentage traffic and evaluate impact using Rasta. We will run various experiments to determine final minimal sell-side revshare, whether we should install throttling before full launch.

Throttling

To prevent dynamic sell-side revshare from eroding AdX margins too much, we plan to throttle dynamic sell-side revshare to maintain AdX margins. Before AdX auction we will flip a coin to decide if the request should run under fixed sell-side revshare or dynamic sell-side revshare. We call this throttling probability, ranging between 0 (no throttling) and 1 (all transactions are throttled and run in fixed revshare). By adjusting this throttling probability, we may maintain AdX margins, which we will run experiments to validate the assumption.

We will calculate actual AdX margins regularly from logs and measure the difference between actual AdX margin and target AdX margin. Suppose we decide to maintain overall AdX margin at 19% (target margin), while actual margin can vary between 20% (fixed revshare, that is, no dynamic revshare) and 0%. When actual margin is below 19%, we will increase throttling probability so that less dynamic revshare and move AdX margin toward 20%. In other words, we observe the difference between actual margin and target margin, and adjust throttling probability to keep actual margin from deviating away from target margin.

Alternatively, we may consider limiting the range of AdX revshare to guarantee AdX margin. For example, we can limit the range of AdX dynamic revshare to be [10%, 20%]. Hence no transactions can be cleared below 10% margin, and we can guarantee that AdX's margin stays above 10%. However, this is sub-optimal for maximizing publishers' revenue. Limited range will exclude queries that are below the range, which are lost opportunities. For example, if 50% of queries can be cleared at 20% AdX margin and 50% at 0% AdX margin. Limited range [10%, 20%] would clear queries with 20% margin, while AdX can actually clean all queries and keep the AdX query at (average) 10%.

The probability throttling here is used similarly for Bernanke. Unlike the probability throttling in Bernanke, we directly measure *AdX margin* instead of *first price rate* and use that in the control loop. Because our goal is to maintain AdX margin, not first price rate, directly observing AdX margin should perform better.

Measuring AdX margin

Effective AdX margin is one minus *publisher payout* divided by *post-buyside-revshare bid* (that is, AdX profit divided by AdX revenue). We will accumulate AdX payout and AdX revenue over all AdX impressions for each web property (by web_property_id) and each buyer (by buyer_network_id).

Example: Without buyside margin (e.g., RTB on AdX)

Advertiser Charge / Bid	Publisher payout	Publisher revshare	Buyside margin	AdX margin
11	10	91%	0%	9%

Publisher payout is $10, and post-buyside-revshare bid is $11. AdX margin is hence 9% (=1 - $10 / $11).

Example: With positive buyside margin (e.g., AdWords on AdX)

Advertiser Charge / Bid	Publisher payout / third-party reserve price	Publisher revshare	Buyside margin	AdX margin
12	10	83%	15%	2%

Publisher payout is $10, and post-buyside-revshare bid is $10.2 (=$12 * (1 - 15%)). AdX margin is hence 2% (=1 - $10 / $10.2).

Example: With negative buyside margin (e.g., AdWords with Bernanke on AdX)

Advertiser Charge / Bid	Publisher payout / third-party reserve price	Publisher revshare	Buyside margin	AdX margin
6	10	166%	-100%	17%

Publisher payout is $10, and post-buyside-revshare bid is $12 (=$6 * (1 - -100%)). AdX margin is hence 17% (=1 - $10 / $12)

The corresponding fields in the logs are the following:

- Advertiser charge: `{unviewed_}impression_set.impression_cost`
- Publisher payout: `adx_query_fields.winner_publisher_revenue_share` * advertiser charge.

- AdX revenue
  - `{unviewed_}impression_set.impression_cost` if `adx_query_fields.buyer_network_id` is not 1 (AdWords),
  - `min(adx_bid_adjustments_info.bidding_parameters.min_percentage * adx_query_fields.max_cpm_usd_micros, max(adx_query_fields.second_highest_cpm_usd_micros, 1.25 * adx_query_fields.seller_reserve_price_usd_micros)` if `adx_query_fields.buyer_network_id` is 1 (AdWords),.

Probabilistic Throttling

We will throttle publishers probabilistically and force them to run auctions with fixed sellside revshare if the average margin of an AdX publisher's webproperty goes below the target AdX margin due to dynamic sellside revshare. We plan to throttle AdX buyers selectively if needed in the future. An AdX auction will either run with dynamic sell-side revshare enabled (that is, the AdX publisher's webproperty is not throttled), or without (that is, the AdX publisher's webproperty is throttled).

The throttling probability per web property is default 0, that is, no throttling. After AdX margin is calculated in the log processing pipeline, we will output a table of throttling probability, with each row for an AdX publisher's webproperty. Throttling probabilities is calculated as

throttling probability = 1 - [(1 - previous throttling probability) * observed AdX margin / AdX target margin]

Throttling probability will be capped between [0, 1]. Previous throttling probability comes from the output of the previous log processing run.

The throttling probability update rules have the following properties

- If observed AdX margin is the same as AdX target margin, throttling probability is not changed (that is, in a steady state).
- If observed AdX margin is smaller than target margin, throttling probability will be increased. Similarly, if observed margin is larger than target margin, throttling probability will be decreased.
- If observed AdX margin goes to 0%, throttling probability becomes 1.0, that is, completely throttled.

This is a simple control loop with AdX margin as the process variable, target AdX margin as the setpoint, and the throttling probability as control variable.

For example, if the target AdX margin is 19% (pub's share is 81%, more than fixed revshare 80%), observed pub's average margin is 12%, and previous throttling probability is 40%, the updated throttling probability will be 63% (= 1 - [(1 - 40%) * 12% / 19%]).

The throttling probability table will be copied to supermixers using experiment file copier and read into supermixer memory through an experiment flag.

Throttling on a per buyer basis

To throttle on a per buyer basis, we test whether a transaction will be cleared on dynamic revshare, and then check whether they winner is probabilistically throttled.

To throttle on a per publisher basis, we check whether the incoming query should be probabilistically throttled.

Throttling during ramp-up

To support throttling during percentage experiments, we plan to support calculating margins and probabilities on a per experiment basis. When we collect AdX revenue and payout on a per publisher or buyer basis, we will filter by a given experiment id. We remove traffic not in the experiment and limit the universe to only traffic in the given experiment. The 1% experiment will be throttled based on statistics from the 1% traffic, which gives a clear picture on what will happen post-launch (100% experiment based on throttling from 100% traffic).

Alternative design is to set a tighter threshold but continue to aggregate AdX margins on all traffic. For example, we can set a target AdX margin at 19.99% for 1% experiment, and reduce the target as we ramp up the traffic percentages. The tighter margin, however, will be likely to introduce more noise during the low percentage experiments, because there are AdX pubs and transaction types that are not in the standard AdX margin and will be throttled.