

COLUMNS

Understanding mobile tracking with UDIDs and cookies

April 25, 2012



By [Michael Dewhirst](#)

[Sign up now](#)

Luxury Daily

There are numerous problems that still need to be solved in the mobile advertising world: patchy cookie support and understanding how they actually work; device identifiers – or, as they are otherwise known, UDIDs – and their inconsistency; and the lack of ad tracking information pass-through within the Apple App store, to name just a few.

The good news? There are offerings on the market right now that allow advertisers and agencies to very precisely track and attribute downloads, conversions and even in-application events such as frequent use, purchases and game-level completion.

In the first of two articles, I will take a look at the problems facing the mobile world, and how they can be overcome.

App download tracking and attribution is a reality right now and the information is usable by media buyers to plug into automated or programmatic buying solutions such as demand side platforms (DSPs) to understand and adjust their buying strategy.

Goal and benefits of tracking

By plugging app download tracking data into a DSP, its learning mechanisms can automatically adjust the buying strategy to buy more of the traffic that has the same profile

as that which delivered the best results.

The DSP effectively finds a “pattern” of the traffic that is most likely to result in a download by looking at many variables such as device types, versions, location, time of day, day of the week and the site the ads are shown on.

When there are enough statistics for the events that the advertiser wants to buy such as a download and install, the DSP will find the same recurring set of variable values and make a note of them.

It will then bid higher for such traffic and buy more impressions with such a profile – all on an impression-by-impression basis. And we are talking 10,000-plus impressions per second here. That is 20 billion-plus impressions a month.

A typical traffic “profile” might look something like this: iPhone or Samsung Galaxy S, Sunday and Saturday lunch time, on sites from publisher XYZ and apps from game maker ABC and so on. We actually use about 30 different variables to determine this pattern.

Think of this as a large number of criteria to distinguish between different people, such as personality, hobbies, height, weight, eye color and hair color, as used on a dating site.

The more parameters you use, the more precisely you can identify a person that is right for you.

The result is then a much more focused buy, which has massively larger yields than that of a blind buy that is done in bulk and not on an impression by impression basis.

This way, the advertiser knows exactly who drove the most effective traffic, and can adjust the spend in the right direction accordingly.

What if there is a “missing link”?

How does a DSP – or anybody for that matter - track and attribute impressions and clicks to a download or subscription if there is “dead space” between the “entrance” and “exit” of the mobile app advertising workflow?

The answer is not a single solution, but a combination of approaches and technical implementations, which together deliver the required result.

UDIDs – set up to fail from the start?

Currently, when a DSP or an ad network buys a mobile banner that runs on a mobile media exchange – or a supply side platform, also known as SSP – the latter’s code in the app passes the UDID back, often one way encrypted, also known as hashed.

So, many ad networks quickly adopted a “match the ID” approach where they would record the UDID of the click and ask the advertiser to send the UDID of the device which installed the app to their server, where they would check to see if that UDID is in the click records – attributing the download if there was a match.

But there are several problems here right off the bat:

- Typically, an advertiser will use many different banners to advertise an app and the

above approach does not precisely pinpoint which banner drove the download. And no, you cannot always argue that last view or last click won.

- Secondly, the exchange may be sending the UDID hashed using one algorithm (e.g. SHA1) and the advertiser may be sending it hashed using another (e.g. MD5) – so there will never be any match. A bit like trying to match a phone number and a ZIP code. They are just never going to match, even if they are for exactly the same house.

- Thirdly, the Android operating system has several IDs available to app developers – AndroidID, IMEI, MEID or ESN. Exchanges and advertisers often will use and send different ones again. Combine that with different hashing algorithms and it is a right mess.

- Lastly, Apple is deprecating the UDID and it will no longer be available within apps to be pulled out by the SSP SDK, so this free ride will end soon. There will be other IDs one can use on the device, such as the Mac address of the WiFi card, but this may not be very reliable either for several reasons.

There is often a lack of clarity when setting up a campaign between the advertisers, agency, media buyer and the exchange and again, the wrong IDs are often compared and therefore never match. Many attributions are missed and do not go back into the mix, driving up the cost of the download and reducing their number too.

One more reason still to ditch the UDID

If this is not enough to convince you, the last and one of the main other problems with this approach is that only media that is running inside an app – and not media seen on mobile sites in a mobile Web browser – can be bought if UDIDs are to be used for download attribution.

This is because a UDID can only be obtained when you have access to the operating system API, i.e. you are some code running inside an app.

If you are a simple meat-and-potatoes HTML page or even a fancy one with some JavaScript, you just cannot get the UDID of the device you are being shown on.

It is like arriving in a building blindfolded and only being able to go into one room with no windows to the outside world. You will not be able to work out where you are unless you can peak outside and glance at a street sign.

This is a problem because there are a huge amount of mobile-optimized sites which could be driving punters to the app store to download that app. Hundred percent more, in fact, in addition to the app traffic.

The net effect is that app inventory becomes more sought-after and, in turn, the price, yield and cost of a download goes up. Yes, this keeps the app developers happy – which is not necessarily a bad thing – but does not benefit anybody else.

“But if the app stores are a dead-end where advertising tracking dies, how do you tie an impression to a download without a device ID?” you ask.

Surprisingly it is back to the old veteran – the cookie.

Back to the cookie

Cookies actually work fine on most mobile devices, especially on all the main ones that have app stores and apps, such as iPhones, iPads, Androids, BlackBerry phones and Nokia devices.

If they did not, many sites where you have to log in such as Gmail, eBay and Facebook would be pretty hard to use as the site would not remember who you were from page to page.

Those that do not work are some feature phones, where you cannot install apps anyway, so who cares.

So a cookie is a bit like that ultraviolet stamp that a bouncer would put on your hand so they know who you are after you pop out for some fresh air or a cigarette and decide to go back in.

The only issue that exists with cookies is on the Apple operating system, iOS – but only with the setting and not reading – of third-party cookies.

If you are not sure what I mean by first- and third-party cookies, I simply refer to the domain of the page you look and the domain of the server where the tracking cookie was set from.

For example, let us say you are looking at a page on Amazon.com and there is a tracking pixel pointing to strikead.com, which tries to set a cookie. This strikead.com cookie will be classed as a third-party one as it is not from the same domain as the page you are on.

Cookies from Amazon.com, however, are first-party since they are from the same domain as the page.

Setting a third-party cookie in iOS Safari, the iPhone browser, will not work. The attempt to set a cookie will be blocked by Safari.

However, reading both first- and third-party cookies is just fine on iOS Safari on the iPhone, iPad and iPod touch.

Getting to cookies from an iPhone app

That is all fine and dandy, you say, but how do I read that cookie from inside the app?

The simple answer is – you cannot, since Safari is just another app and apps on iOS cannot share data for security reasons. The longer answer is you can, but there are a few workarounds to be done.

The problem with reading cookies from inside an app is that they are set in the device's browser, and only it has access to them.

You can, however, launch one app from another. For example, you could launch the

device's Web browser, such as Safari, from inside the downloaded app.

When you do so, you tell the server to go to the same tracking page where the cookie was set earlier during the click.

You can also pass the server any necessary information for the download to be attributed to the click and this information gets passed to the server.

The server then tells the browser to go back to the app it came from and the loop is complete. No more "dead space."

It may sound like magic, but this is possible and it works very well.

The only negative effect of this process is a brief "flash" of the browser window whilst you are in the app – a few hundred milliseconds – blink and you definitely will miss it.

Many advertisers are already using this method and have found that it does not affect user experience and only gives them much better insight into the app life.

Think back to the desktop computer and the standard browser – pop-ups are still common there and nobody really cares about them.

What about Google Play?

Good old Google, being deeply steeped in advertising, understands that certain things need to happen for the advertising machine to keep turning its wheels and provide free app developers with a source of income.

So, Google Play – the former Android Market – actually has a mechanism which allows variables to be passed to it from a click on a banner and from there to be passed to the app. The app can then pass this information back to the media buyer for attribution and optimization.

So for those advertisers who really cannot have a pop-up in their app – at least on Android – there is a way.

Technology is here now

This technology is a reality now.

It is clear that there are solutions out there to make mobile app advertising more successful and cost-effective, but it will take the triumvirate of the advertiser, agency and media to adopt them.

Without all three parties understanding the options and using them, nothing will happen – or at least not easily and not quickly.

Tomorrow: Understanding mobile downloads: Tracking conversions and solving the download lag

Michael Dewhirst is chief technology officer of [StrikeAd](#), London. Reach him at mike@strikead.com.

© Napean LLC. All rights reserved.

Luxury Daily is published each business day. Thank you for reading us. Your **feedback** is welcome.