Google vs Bing

Redacted@, Redacted@, Redacted@, Redacted@, Redacted@, Redacted@

go/google-vs-bing

June 2017

Summary of Findings

Looking specifically at mobile queries on browsers, Bing consistently serves search results faster than Google today:

1. Bing results arrive ~300ms faster
   Not including differences due to SSL, Bing search results start to arrive and render ~300ms faster than Google search results.
2. Google has a larger latency penalty for logging-in than Bing
   Results for logged-in user queries on Google arrive ~350ms later than queries by logged-out users. Logged-in queries on Bing are only ~100ms slower.
3. Bing is faster in part due to our server-side latency increasing since Q3 2016
   Latency trends show that, while user connections are gradually improving, server-side latency is getting worse at a faster rate. Roughly ~150ms of the gain comes from Confidential and ~80ms from Confidential itself.
4. Bing has more granular streaming
   The Google SRP comes down in four distinct chunks (header, body, footer, late footer). Bing delivers their SRP in many more granular chunks.
5. Bing has a smaller payload size
   On average, the Bing /search page is half the size of Google a google /search page (~200kb vs ~100kb uncompressed, excluding external resources).
6. Bing and Google's client-side rendering times are comparable
   While there may be room for optimization, the client-side rendering times for both Bing and Google are roughly the same.
7. Bing does not use SSL by default
   Using SSL incurs an amortized ~26ms loss per query relative to Bing without SSL.
8. Bing does not support HTTP/2 and QUIC
   This may cause a higher response time on high-latency networks for Bing due to the TLS negotiation process on every query.
9. Bing is more adversely affected by poor networks
   Under poor network conditions, Google is actually faster than Bing.
10. Google's mobile traffic incurs more server-side latency than desktop traffic
    The latency is spread evenly between Confidential and differs by ~50ms.

Ex. No.
UPX2022
1:20-cv-03010-APM

Redacted

Looking at the Bing App on Android and iOS:

11. Bing uses native rendering
    The SRP is implemented with native widgets on Android and (most likely) on iOS as well and shows approximately the same ~300ms difference in performance.
12. Bing implements infinite scrolling
    The Bing App implements tap-for-more-results infinite scrolling.

Furthermore, ramiroguerra@ has done an analysis of Bing Mobile web.

Background

As part of the Folly effort, it was observed that today Bing appears to serve search results faster than Google. In one of the worst cases found, the query "san diego to lax train" takes 3.69s for content download on Google versus only 565ms for Bing (see transport query discussion). We set out to quantify the difference and try to drill down into possible reasons as to why this could be.

2014 Latency Lab study

This question has come up before. In 2014 latency lab studies, Bing was faster than Google mainly due to its lack of SSL. However, in 31 of the 100 queries, Google was also slower than Bing with SSL. The reasons identified are listed here. It is possible that since then Bing has made even more performance gains (e.g. Windows Server 2016 and improved .NET yielded 16% Bing search latency reduction).

Findings

For more details, see the additional information in the Appendix.

Bing results arrive ~300ms faster

Using 1000 random queries to get a more realistic sample of what real traffic would look like (rather than using "sloth" queries or common queries) and an automated query tool we gathered latency data for both search engines on different networks. Both search engines return the first byte quickly and at about relatively the same time (chart).

However, we see a large difference when we look at the time that the first byte of the *search results* arrives:

Comment [1]: Are there lessons to be learned if we look at specifically the queries of sloth?

Comment [2]: You will likely find pages where Google has one or more heavy features and Bing has a feature-light page. I don't think the Bing-vs-Google context is adding more insight rather than just looking at Google sloth queries independently.

Comment [3]: Would this be a useful latency metric to track for SRP?

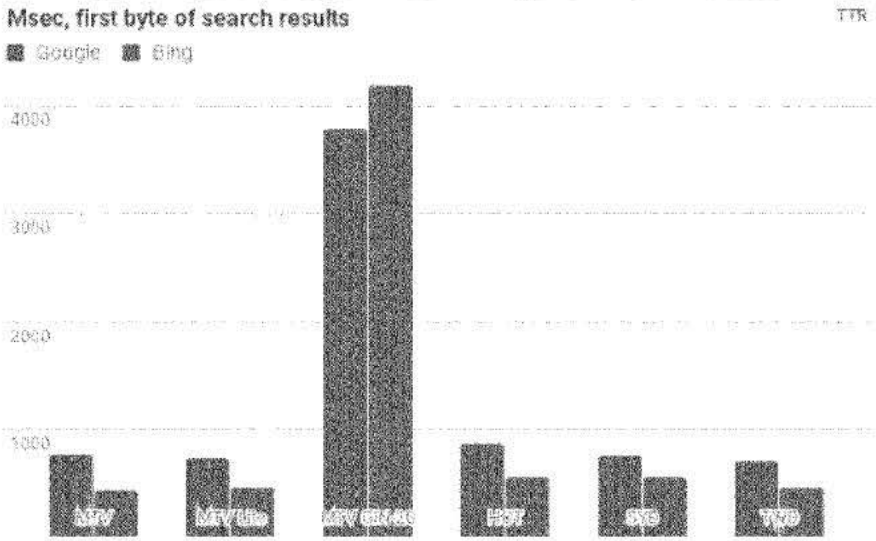
Comment [4]: Maybe but any gains here will reflect in AFT directly.

Comment [5]: I wonder if there's some useful analysis to be done on the gap between TTFB and TTFR. The page is perceived as faster when the first result comes in more quickly, but AFAIK we don't have quantitative data on the effects. If we're going to expend effort closing the TTFR gap, this might give us some idea of the headroom.

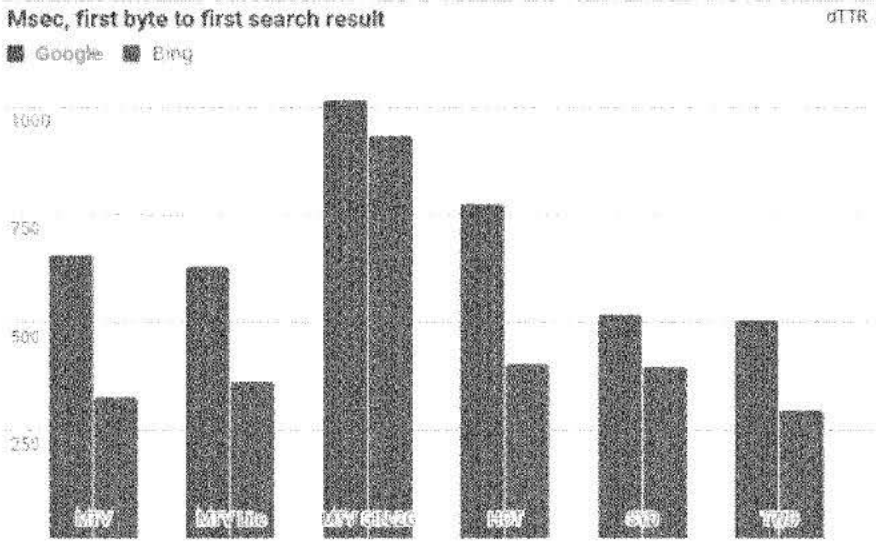
go/searchlatencyprimer seems to indicate that we collect chunk boundary metrics, but the doc referenced is from 2013 -- do we still collect these? https://docs.google.com/document/d/1Rf7nGYfl23S5eD07U3BzdKLcGQLI3VK4kqqA4TW_IkU/view

+Redacted@google.com +Redacted@google.com +Redacted@google.com

Comment [6]: I can't find the InstrumentationService in our codebase anymore. Not sure how difficult it would be to implement cl/41166371 in our current state



When the connection is fast, Bing results arrive 120-370ms earlier depending on the region. This effect is more easily seen when looking at the difference between the time of the first byte ("header chunk") and the first byte of the search results ("body chunk"):



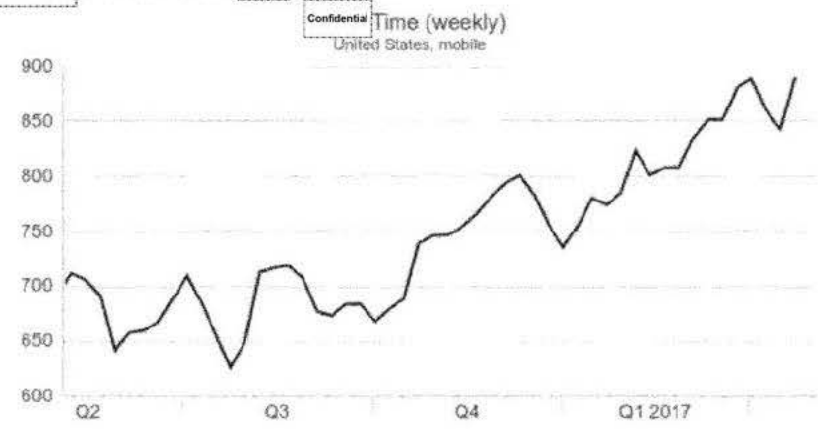
Redacted

The largest single source of server latency appears to be from Confidential in particular, rather than within Confidential itself. Full data is available here.

The Google header is 9.5kb compressed on Tier 1 and renders in under 100ms on even the slowest devices. The time between the header and the first byte of the results is a particularly pernicious place to have a latency gap because even the slowest of client devices will be sitting idle while waiting for the search results bytes to arrive.

Bing is faster in part due to our server-side latency increasing since Q3 2106

The server-side latency difference between Bing and Google is quite significant and this difference seems to be a recent development. Much of the latency that Google lags behind Bing has been gained over the previous two quarters. Looking at mobile traffic, roughly ~150ms from Confidential and ~80ms from Confidential Time itself:



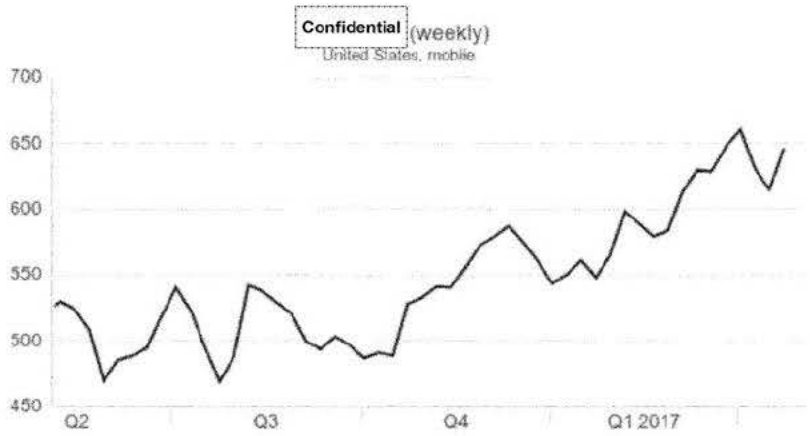
Comment [7]: I don't get to this conclusion from the chart and your earlier words. It seems like you're saying that there's a X00ms latency gap between first byte and first search result. But, what's happening during that time? Is it _waiting_ on [redacted] or some other server? Is it possibly that we're transmitting a lot of bytes between these points (ie. that it just takes more time to reach the first byte)? Could it be that the browser is doing something with those earlier bytes (eg, perhaps there's inline javascript that's being executed)?

Comment [8]: Redacted@google.com Yes, the charts show a X00ms gap between first byte and first result. Confidential Confidential These queries were done with an automated query tool that has no browser and looks for a particular string in the network response.

Comment [9]: To answer all of Corey's questions directly:
1. Yes, the browser sits idle.
2. Yes, it is waiting on bytes from Confidential

Confidential

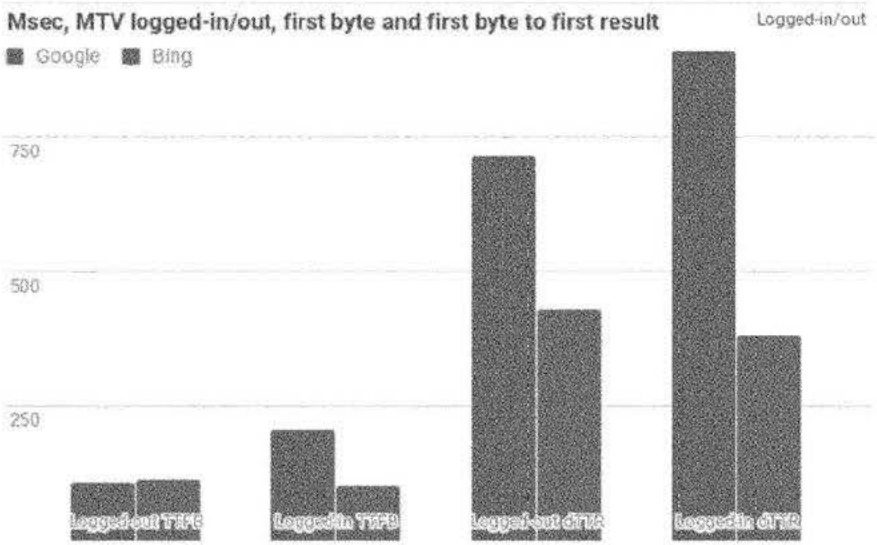
Comment [10]: Michael, do you want to add additional conclusions to the text here?



Much of these losses have been masked by increasingly fast user connections.

Google has a significant logged in penalty that Bing does not have

On Google, for logged-in user queries, the first byte of the header arrives ~100ms later than for logged-out users. Additionally, the first byte of the results will arrive ~240ms later. This gap is server-side processing. Not clear what exactly but this latency shows up on the end-user latency dashboards split between Confidential Response size is about the same. For Bing, no significant difference between logged-in and logged-out queries was observed.



These delays show up in our end-user latency dashboards and appear to affect **Confidential** and even header time. See the appendix for more data.

Bing has more granular streaming

We investigated the loading behavior of both Google and Bing search and noticed differences in how results are streamed to the client. Using a tool to analyze the time that bytes arrive, the Google SRP comes down in four distinct response chunks (header, body, footer, late footer). However, Bing delivers their SRP in many more granular chunks.

Bing's streaming approach may allow better latency with poor network connectivity but does not appear to impact latency on good connections. During testing using the corp network and with a home cable connection, the average time between the first result byte and the last byte of the SRP was ~30ms for both search engines.

Bing has a smaller payload size

Bing's /search pages are significantly smaller in size than Google's. We captured byte-size and page-load times for some of the suggested queries from the project slides as well as byte-size breakdowns for a set of 1000 random queries. For the /search page, Bing loads much fewer bytes than Google does. On average, the Bing /search page is half the size of Google a google /search page (~207kb vs ~112kb uncompressed, ~66kb vs ~35kb compressed).

Comment [11]: Why else might Bing have finer granularity? Did you observe that the chunks lined up with any particular boundaries? How do browsers treat chunks – is there something to be gained, related to browser processing, by using finer or more coarse chunks?

Comment [12]: Looking at Wireshark, there doesn't seem to be any logical segmentation beyond being 8kb chunks. Redacted@google.com anything else to add there?

Comment [13]: It didn't seem like explicit boundaries to me and the amount of chunks varies quite a bit. Browsers will render earlier data they receive earlier. Beyond that there is no effect.

A byte-breakdown comparison between Bing and Google further shows that Bing SERPs are much smaller in most byte categories. Of the roughly -95kb difference going from Google to Bing, about -33kb comes from HTML markup, +19kb comes from HTML text (quite a bit of this is in fact CDATA containing JS), -19kb comes from style blocks, and -62kb comes from script blocks. Of the difference due to script blocks, -31kb comes from JS inlined images. Of the difference due to style blocks, -2kb comes from CSS inlined images.

Bing loads fewer bytes of external resources than Google: 168kb versus 403kb of XJS, and 16kb versus 37kb of external images. However, Google does much fewer XJS fetches (exactly one) than Bing, which does about 15.8 XJS fetches on average.

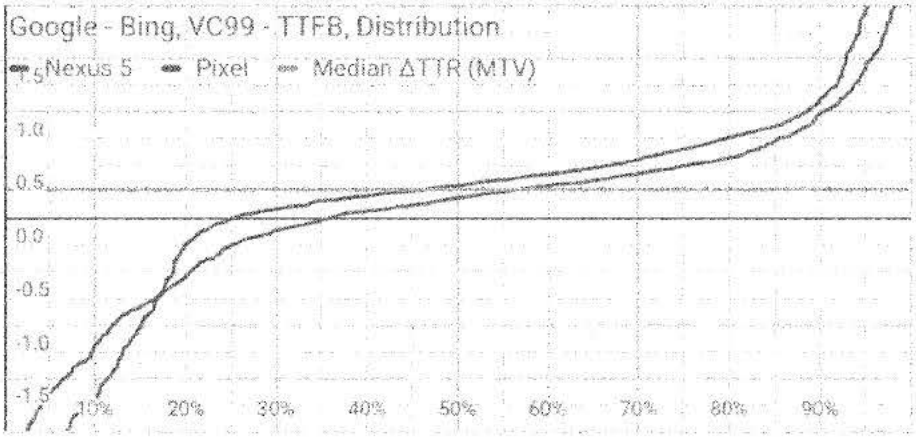
Comment [14]: Do we have a measure of how much time the browser takes to process this js for bing v/s google?

Comment [15]: We can see this in Chrome performance profiles but all of the XJS processing time is after AFT.

Bing and Google's client-side rendering times are comparable

Although it depends on the device used, the client-side rendering latency of Google and Bing are largely comparable. We gathered aggregate data on overall latency using Latency Lab by running 1000 random queries on both Google and Bing on a slower Nexus 5 phone and a fast Pixel phone and recording the time that each search page reached 99% visual completion above the fold.

The following graph plots the latency difference distribution for each phone:



Each blue point in the graph plots the difference in time to 99% visually complete minus time to first byte (to control for network effects) between Google and Bing for a particular query run on a Nexus 5. The red line plots the same for a Pixel phone. Points above the horizontal axis indicate that Google is slower and points below indicate that Bing is slower. This graph shows that today 25% of queries in this set are faster on Google on a Nexus 5 and 35% are faster on a Pixel phone.

The orange line highlights the median server-side latency gap (as measured in MTV) to highlight the fact that, had the search results reached the client at the same time for both Google and Bing, Google latency would actually be just slightly worse than Bing on a Nexus 5 (45% faster) and in fact *better* than Bing's on a Pixel (55% faster).

An in-depth look at client-side latency for "queries of sloth" (queries specifically chosen because they are slow on Google) actually showed that the above-the-fold area rendered *later* for Bing in most cases.

Bing uses more images

This can likely be attributed to more prolific use of images on Bing above the fold. The following data is for ~600 random English queries:

	Google	Bing
Average images above the fold:	4.82	6.94
Average number of image pixels above the fold:	70k	138k
Average total images:	24.54	14.58

Note that this analysis included a doodle running on the day the data was collected which accounts for ~10k of above-the-fold pixels on Google.

Bing does not use SSL by default

Bing does not use SSL by default; Google does. SSL traditionally required one or two extra round trips, but protocols such as QUIC have sped up SSL, and in practice only 11% of HTTPS queries require a SSL handshake, delaying the search by ~240ms[1] when a handshake is required. This amortizes to 26ms per query.

Bing does not support HTTP/2 and QUIC

Bing does not appear to support HTTP/2 (test) or use QUIC (as evidenced by the lack of UDP traffic at the packet level). This may cause a higher response time on high-latency networks due to the TLS negotiation process on every query. In contrast, abou[redacted] of Google's traffic uses HTTP/2, and [redacted] QUIC.

For Google, using HTTP/2 appears to give only a minor benefit to latency for new connections on a simulated 3G network:

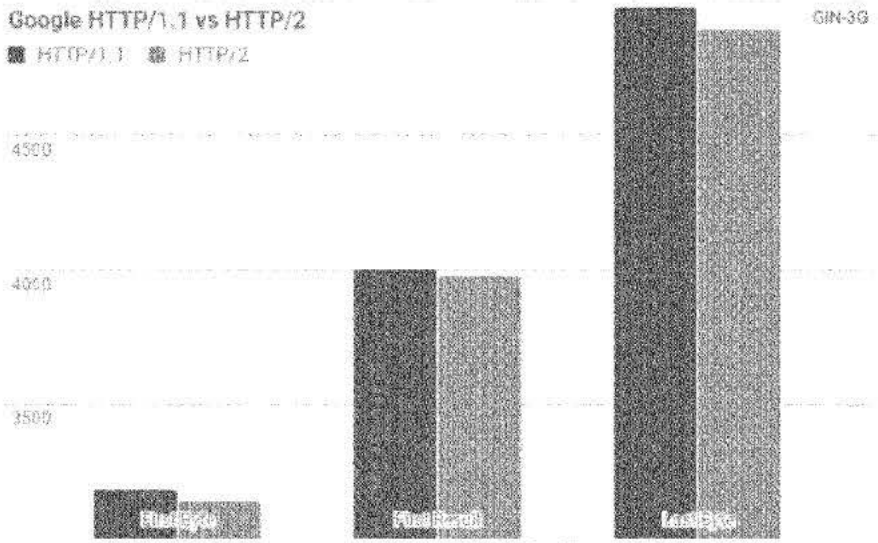
[1] As measured by the Navigation Timing API: connectEnd - secureConnectionStart in https://www.w3.org/TR/navigation-timing/#processing-model

Comment [16]: I'd also like to investigate latency impact of http->https server redirects, which wouldn't be included in these metrics.

Comment [17]: [Redacted]@google.com do you have data for how frequent that is? I though we looked into it...

Comment [18]: Volume with our RDXT metric (obtained as redirectEnd-redirectStart in https://www.w3.org/TR/navigation-timing/#processing-model) is very low, less than 1%. I'm not sure though whether HTTP to HTTPS redirects are included since according to that page, the source and destination have to have the same origin.

Comment [19]: I think amortising arithmetic is detrimental to our users interests. It'd be more correct to say that we had an overall reduction SSL handshakes with QUIC, but an increase in latency variability of 240ms in 11% of queries.



It may however affect repeat connections and allow benefits not measured in this test such as server push.

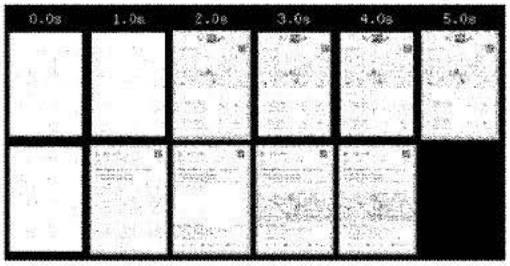
Comment [20]: We had at one point done H2 holdbacks in the wild to measure impact and it might be interesting to do that again.

Bing is more adversely affected by poor networks

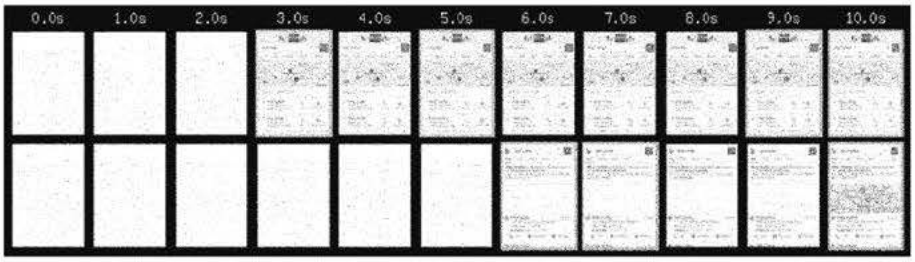
Perhaps because Bing and Google use a different method of SSL negotiation, HTTP protocol, and a different number of images and other resources, the two search engines do not appear to be proportionately affected by increasing network latency. It seems that in general, Bing appears to be much more adversely affected by poor network connections in terms of both time to first byte and loading images. This effect can be seen in the GIN 2G time-to-first-result data presented below.

For illustration, the following film strips were taken with different amounts of network throttling:

[philz coffee] with no network throttling

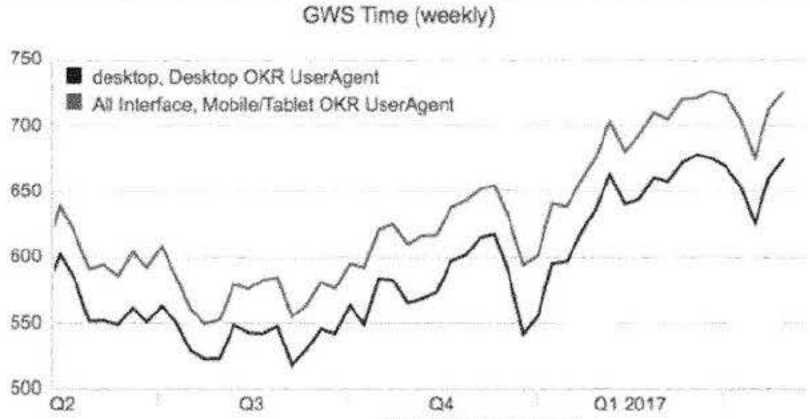


[philz coffee] with simulated 3G



Google's mobile traffic incurs more server-side latency than desktop traffic

Server side latency is ~50ms larger for mobile:



This latency is about evenly spread between **Confidential** This difference can be measured when issuing the same query on mobile and Desktop which suggests that the latency has an infrastructure or feature source rather than a being due to query mix.

Comment [21]: Have you dug into why this might be the case?

Comment [22]: My guess is # of features. Server latency is also much higher/growing faster over last 2 quarters for US/UK than for many other locales

Comment [23]: The strange thing here actually is that tablet is even slower than mobile. I think it may be due to the query mix.

Comment [24]: Is the 50ms latency difference seen when looking at the same query on Mobile and desktop?

Comment [25]: Seems you most definitely can. I just ran the query "lemons" 10 times using a mobile and a Destkop UA (via curl, logged out) and see ~200ms of difference in time to first result.

The Bing App uses native rendering

The Bing App only uses WebViews for rendering results pages and using the Hierarchy Snapshot Viewer we can see that the Bing App SRP itself uses native widgets:

It is not as easy to do this kind of analysis on iOS but the visual and functional similarity between the two apps suggests that both are natively implemented. One confirmation that Bing does use native rendering on iOS is that they use Apple Maps in results, which is native only.

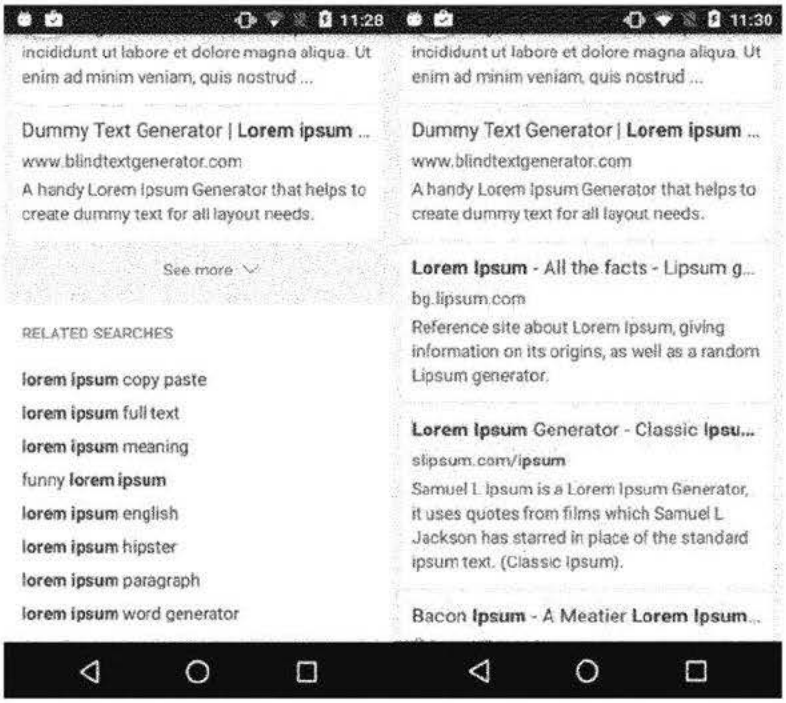
The Bing app renders results faster than the GSA on Android and iOS by 300-400ms, which is about equal to the server-side latency difference. See the side-by-side videos on Android and iOS for an example query (note the longer App startup time for Bing on both platforms).

The Bing App implements infinite scrolling

While the Web interface of Bing does not have infinite scrolling, native scrolling is easier to implement with a native widget based interface so it's not surprising that this is a feature of the Bing App:

Comment [26]: What's fun is it looks like bing simply appends the new cards data then re-renders all the cards it has received to date. After paginating 5+ times the render speed on my phone got very slow (15+ seconds whole app locking up) and RAM went to a max of 130MB. They haven't solved for recycling, unlike Facebook.

Comment [27]: So I'd say infinite given infinite RAM and CPU :)



The Bing App still requires the user to tap to see additional results rather than loading them as the user scrolls.

Appendix

Server-side latency analysis

User connections are steadily improving

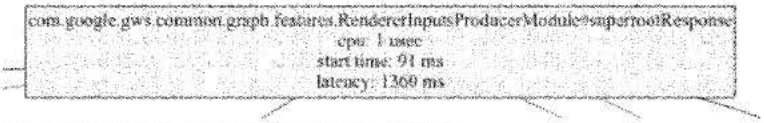
This chart shows SRT over non-GSA browsers as an approximation of time to first byte:

Comment [28]: The trend is actually even stronger than this, since our Confidential Confidential increased 15ms in the same period.

SRT is the time from query commit until javascript in the header chunk is executed. The chart excludes GSA because the WebView does not receive the header data (and record SRT) until the body chunk arrives.

Largest source of latency

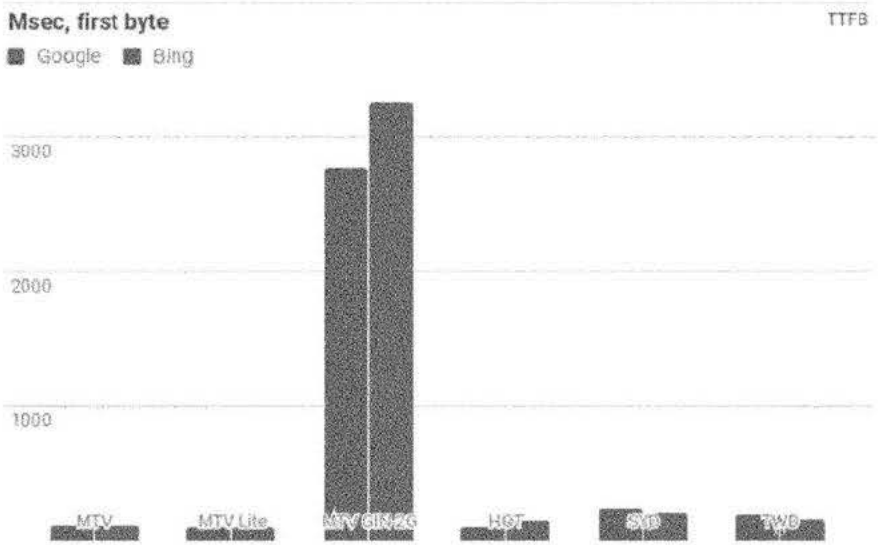
The following screenshot is from the /producerz graph for the query "22 jump street release date" (which has an AFT greater than 2s on the Google corp network!):



The same query is answered by Bing in ~500ms.

Time to first byte on various networks

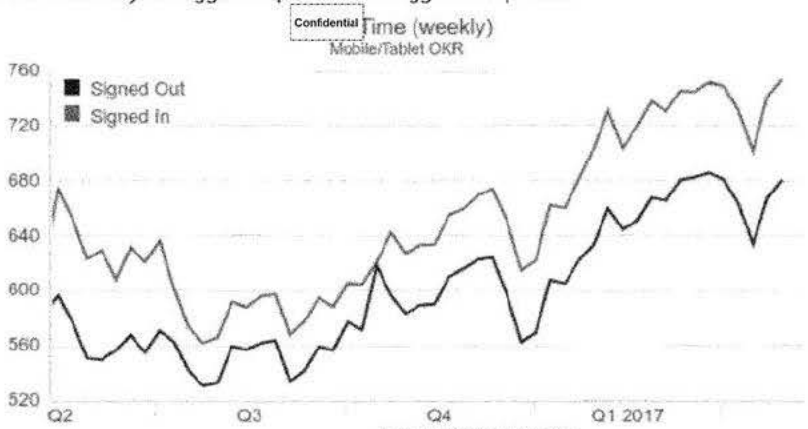
The time of the first byte is consistently about equal for both search engines across different networks:



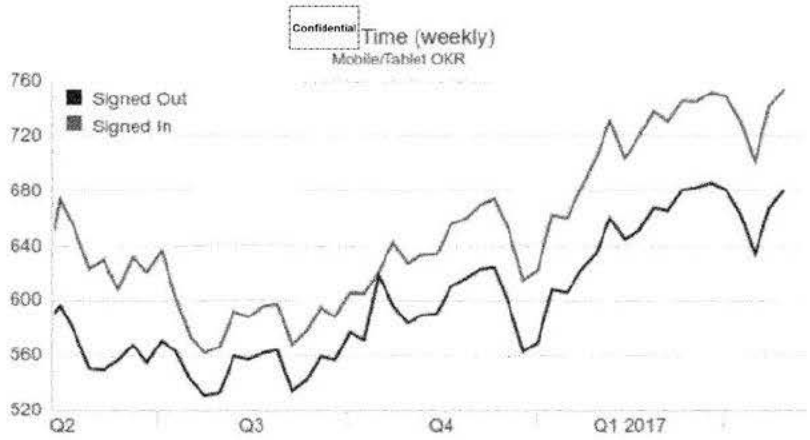
The only exception is the GIN-2G run. It's not clear whether Google is actually faster on slow networks or that this is an artifact of the GIN WiFi networks.

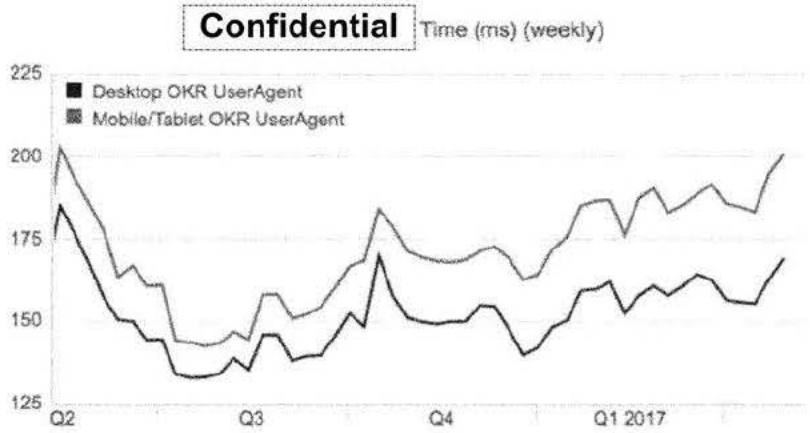
Logged-in/out query analysis

Looking at end-user latency dashboards we see that there is an overall ~70ms increase in server-side latency for logged-in queries over logged-out queries:

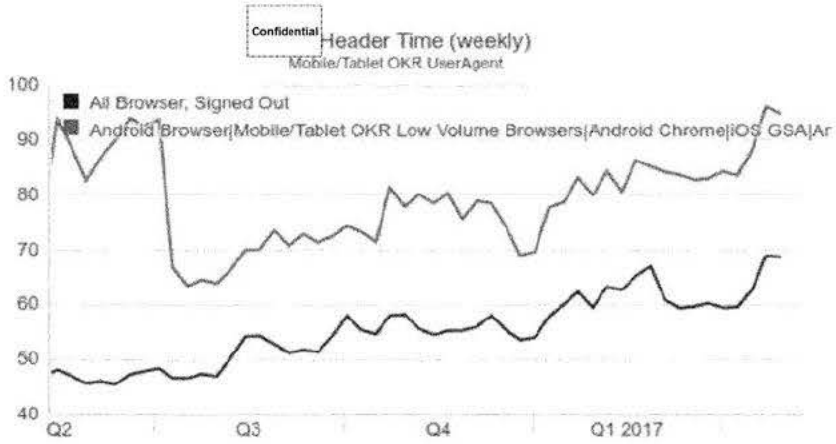


This latency is divided about evenly between Confidential.



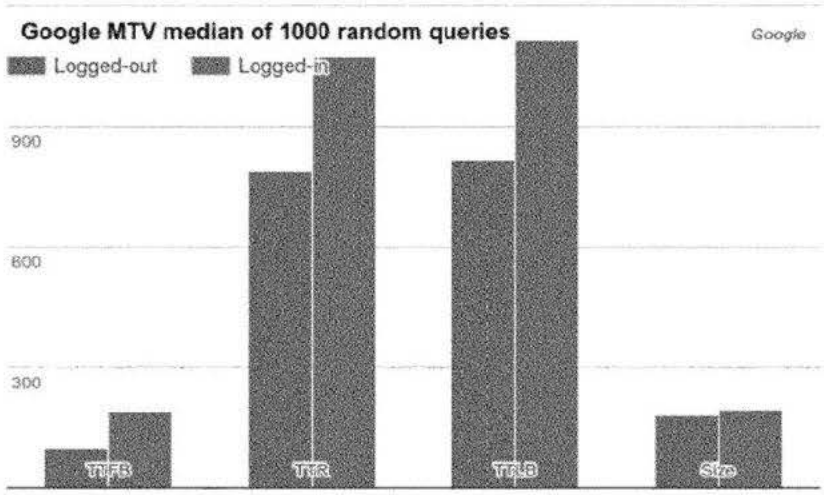


Additionally there is a ~25ms increase in the time to render the header:

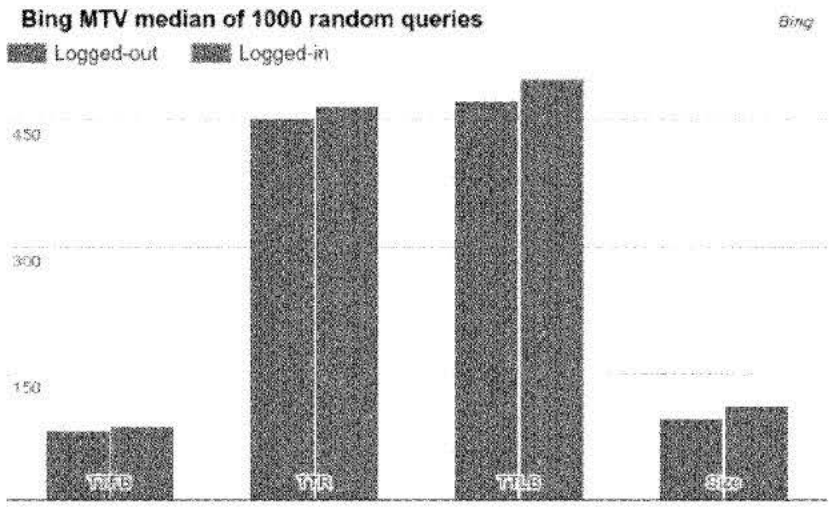


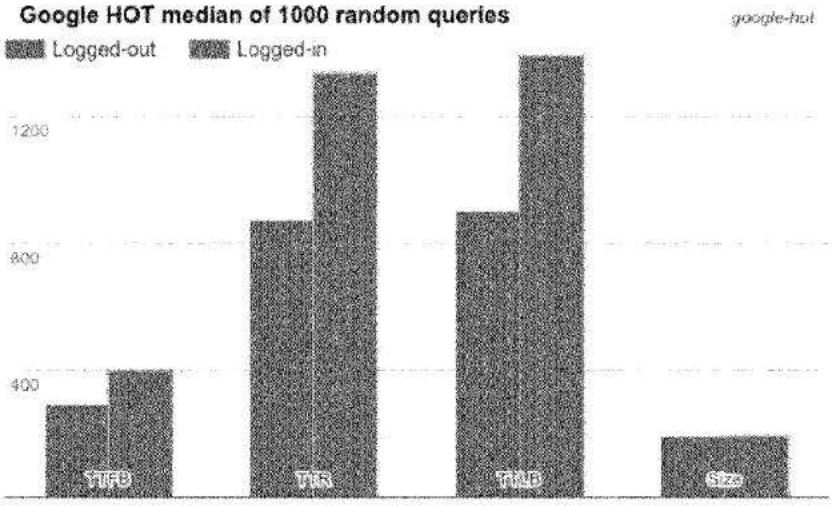
Note that this graph excludes AGSA since AGSA header time was severely affected by the native SRP QBT rollout.

Using 1000 random queries, in MTV and HOT datacenters:

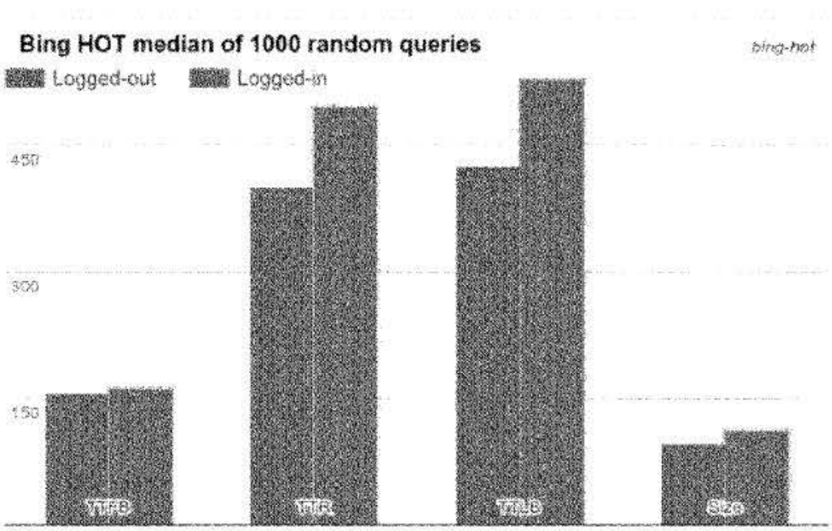


Redacted





Redacted



Full data is available here.

Bing uses more granular streaming

Using a Python script that records the time that bytes arrive, we can easily see the four Google SRP chunks (header, body, footer, late footer):

```
$ curl -s "https://www.google.com/search?q=cassini+space+probe" -A "Mozilla/5.0
(iPhone; CPU iPhone OS 9_1 like Mac OS X) AppleWebKit/601.1.46 (KHTML, like Gecko)
Version/9.0 Mobile/13B137 Safari/601.1" | data_rate.py
  78ms: ████████████████████████████████████████████████████████████ 52k
 564ms: ████████████████████████████████████████████████████████████ …
128k
 575ms: ██████████████████████████████████████████████ 40k
 587ms: ████████████████████████████████████████████████████████████ … 65k
```

Bing appears to stream results in many more chunks than this:

```
$ curl -s "https://www.bing.com/search?q=cassini+space+probe" -A "Mozilla/5.0
(iPhone; CPU iPhone OS 9_1 like Mac OS X) AppleWebKit/601.1.46 (KHTML, like Gecko)
Version/9.0 Mobile/13B137 Safari/601.1" | data_rate.py
 134ms: ██████████████ 12k
 157ms: ███████████████████████ 20k
 180ms: █████████ 8k
 407ms: ██████████████████ 16k
```

Comment [29]: Redacted@google.com do you think we can do a breakdown on how this visually looks on the page? I think that understanding the user-facing impact that streaming more chunks may yield (if any).

Comment [30]: I can look into it but my hunch is that the chunk at 407ms here for Bing would contain all of the ATF content after the header.

Bing's streaming approach may allow better latency with poor network connectivity but does not appear to impact latency on good connections. During testing using the corp network and with a home cable connection, the average time between the first result byte and the last byte of the SRP was ~30ms for both search engines.

Queries of "slothier than Bing"

We can sort the list of 1000 queries by the most negative delta between Bing and Google first-byte to first-result time to find the slowest queries in the query set relative to Bing:

lata mangeshkar song	-1452
22 jump street release date	-1173
азия	-1091
Ghana news	-1012
james songs	-1003
chatham	-943
bj's	-933

Mobile vs Desktop server-side latency

Graphs from the End-User Latency dashboard:

Comment [31]: The [star trek cast] query (which has a comparatively v slow /search HTTP response - see link below - possibly due to TopicServer fanout bug) doesn't appear in this sorted list, I guess because not in the random sample set, which suggests to prioritize effort we may want to also analyse differences use traffic-weighted go/queries-of-sloth.

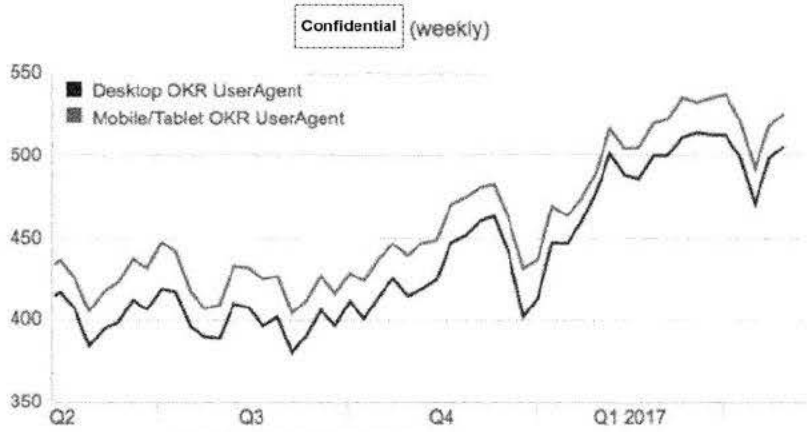
https://docs.google.com/document/d/1Ng-2-ba6ZRyBZp6s_mqg5h03FSGBcNV55xLcPEp5tfo/edit#heading=h.lsawfokydpr8

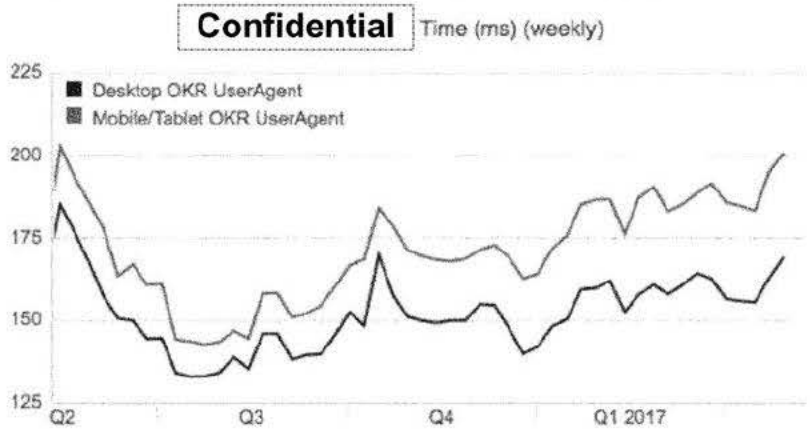
Comment [32]: Yes the query set we used isn't going to have the worst cases but I would argue that picking random queries is already giving us traffic-weighted sampling.

Although I do wonder, can a query appear more than once in the sample?

Redacted@google.com

Comment [33]: I'm not sure. uniq doesn't seem to work well with the foreign characters, but at a quick glance I didn't see any.





Bing does not appear to pre-compute SRPs

Timings taken of randomly generated 7-word tail queries (usually getting only one or two dozen results) did not show any significant difference from extremely head queries ("facebook"). If Bing does precompute any part of the SRP, the latency effect is too small to make any difference.

Bing has a smaller payload size

Byte-size and page-load times for some of the suggested queries from the project slides. Byte-breakdown comparison between Bing and Google for 1000 random queries is available in this cwtool report. Bing seems to be much smaller in most byte categories excepting HTML text and

Redacted

inlined CSS images, and head JS. In particular, Bing loads a smaller amount of external resources (XJS and images) than Google. On average Bing does about 15 more XJS fetches, and about one less image fetch than Google.

CDATA

A Bing SERP has much more HTML text than a Google SERP (25.4kb versus 6.3kb). Much of Bing's HTML appears to come from embedding JS wrapped within a CDATA section and placed as a comment in a non-displaying div. For example:

```
<div style="display:none"><!--//<![CDATA[var PushPin=…;//]]>--></div>
```

This seems to have something to do with XML parsing athough it's not clear what.

Comment [34]: Maybe it helps the document to parse as XML? http://stackoverflow.com/questions/66837/when-is-a-cdata-section-necessary-within-a-script-tag

Click tracking

Google uses a combination of click tracking methods that take roughly 50 to 100 compressed bytes per url. Bing's click tracking mechanism is much shorter than Google's and appears to take just a few bytes. It appears to rely on JavaScript to pick up extra information stored in the "h" attribute. Below we compare the markup for the wikipedia link from the "trump" SRP.

Bing on Desktop

```
<a href="https://en.wikipedia.org/wiki/Donald_Trump" h="ID=SERP,5138.1">
  <strong>Donald Trump</strong> - Wikipedia
</a>
```

(https://screenshot.googleplex.com/akcGsh5rEH7)

Google on Desktop (HREF rewrite clicktracking)

```
<a href="https://en.wikipedia.org/wiki/Donald_Trump"
   onmousedown="return rwt(this,'','','','12',
                           'AFQjCNHbjSG94byvo78Dafly1Q7_0QlYKw',
                           'CE3IvyViDcUTuPjiYCjSQA',
                           '0ahUKEwjxlqTGgMjTAhWpiFQKHS7sCPwQFgheMAs',
                           '','',event)">
  Donald Trump - Wikipedia
</a>
```

(https://screenshot.googleplex.com/pKccEZ2QBLz)

Bing on Mobile

```
<a href="https://en.m.wikipedia.org/wiki/Donald_trump" h="ID=SERP,5246.1">
  <strong>Donald Trump</strong> - Wikipedia
</a>
```

(https://screenshot.googleplex.com/i2rw3iLWsDW)