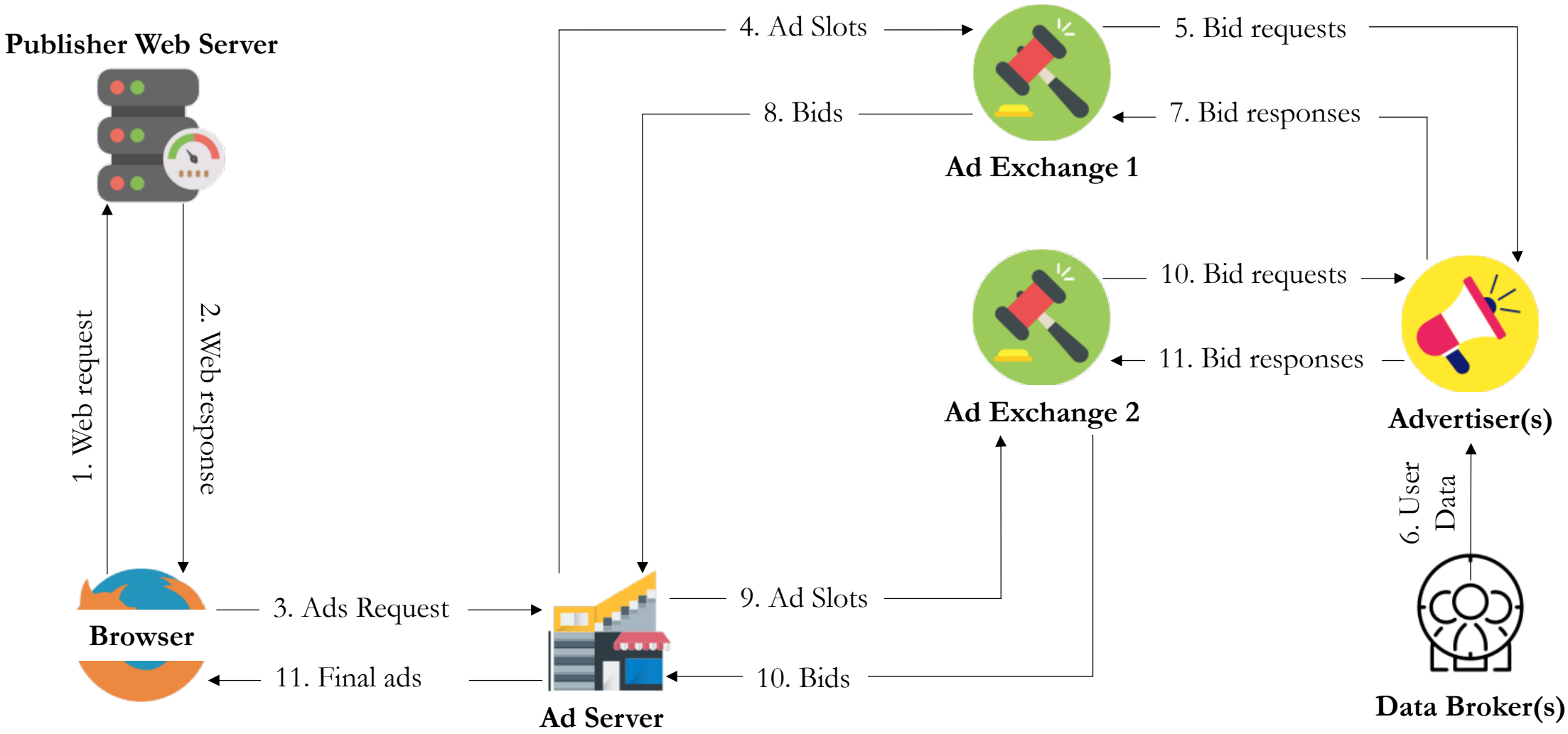
Untangling Header Bidding Lore

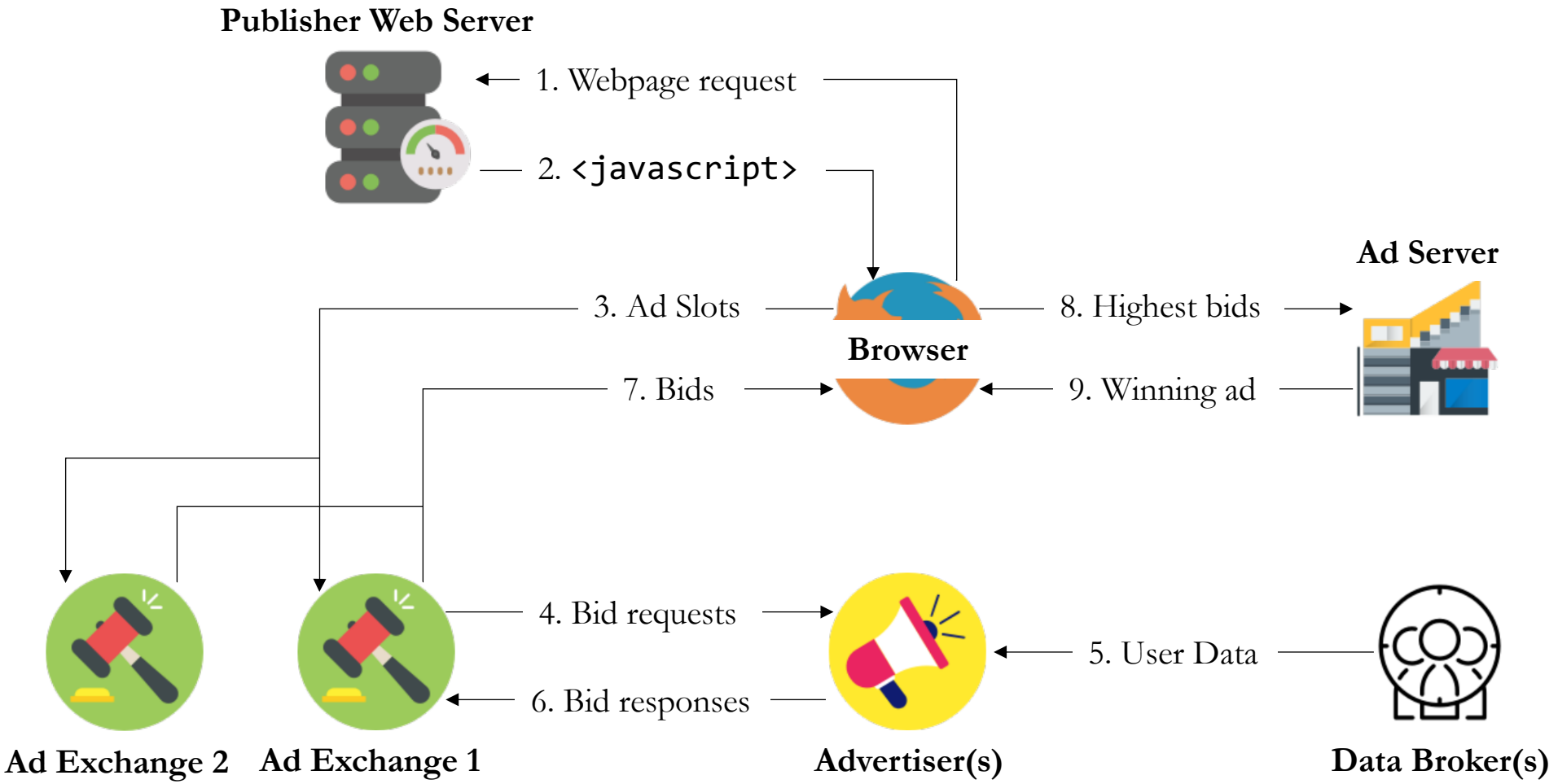
Some myths, some truths, and some hope

Waqar Aqeel, Debopam Bhattacherjee, Balakrishnan Chandrasekaran, Philip Brighten Godfrey, Gregory Laughlin, Bruce M. Maggs, and Ankit Singla

How (traditional) Real-Time Bidding Works



How Header Bidding Works



Header Bidding Background

- Started in 2013 to take wrestle control back from big players (Google)
  - Waterfall model used to favor particular exchanges
  - Parallel process guarantees fairness for all
  - May increase revenue because more buyers can bid
- 80.2% adoption among top 1K publishers
- Online advertising is a $300 billion industry
- Latency-critical process

Previous work

- Only one measurement study on header bidding:
  - Scraping instead of real user data
  - Single vantage point
  - Unrealistic bids
  - Less focus on latency

"~~Non-Viable Performance Overheads~~"

Using real data and a deeper dive into latency, we show that latency overheads are not fundamental

What was measured? How?

Browser extension[1] for Firefox and Chrome measures:

- Prebid.js library logs for ad slots, exchanges and bids
- PerformanceTiming API for timing breakdown of bid requests and responses
- WebExtensions API for IP addresses of ad exchanges
- Domain name of page visited
- Users' city-level location

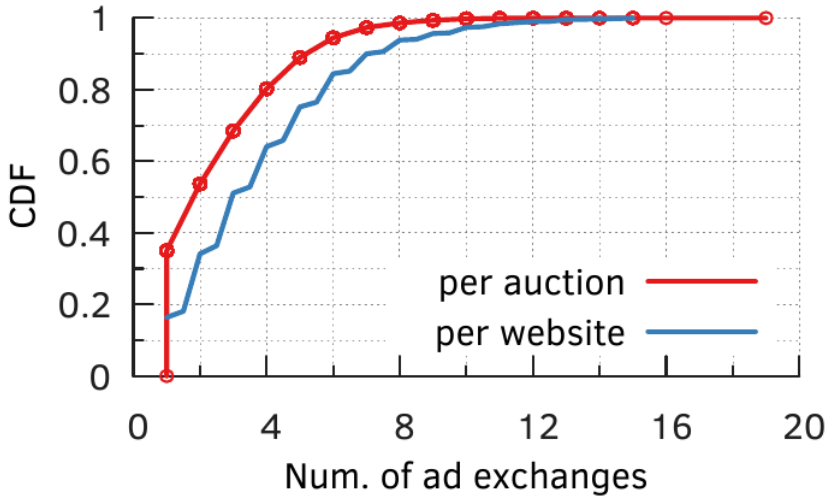
Privacy of users considered – IRB review

Attribute	Value
Users	≈ 400
Duration	8 months
Cities	356
Countries	51
Websites	5,362
Ad exchanges	255
Page visits	103,821
Auctions	393,400
Bids	462,075

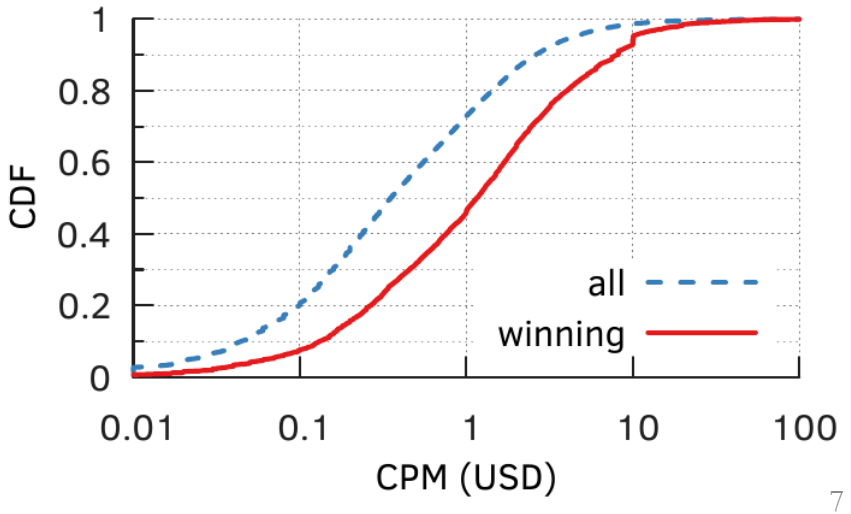
1. Extension source code and dataset available: https://myadprice.github.io

The Revenue-Latency Tradeoff

- Does it make sense to contact as many exchanges as possible?
- Publishers are conservative: ~60% contact at most 4 exchanges



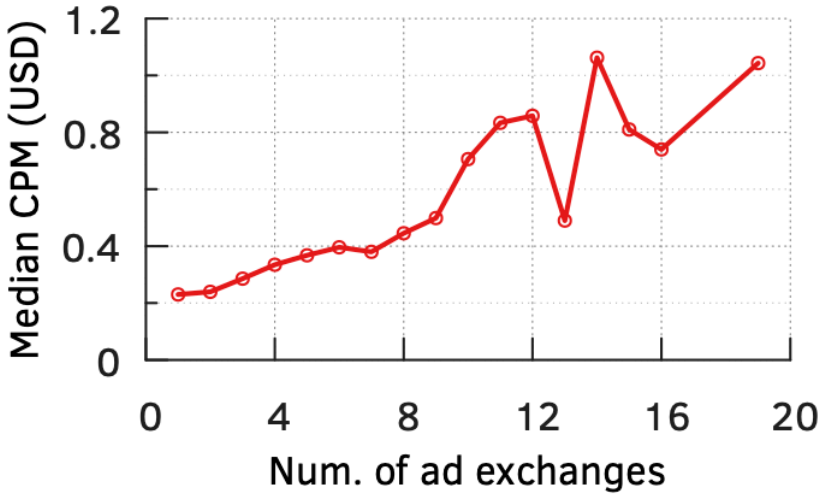
- All bids are not the same
- Median winning CPM is $1.15, while median non-winning is $0.35

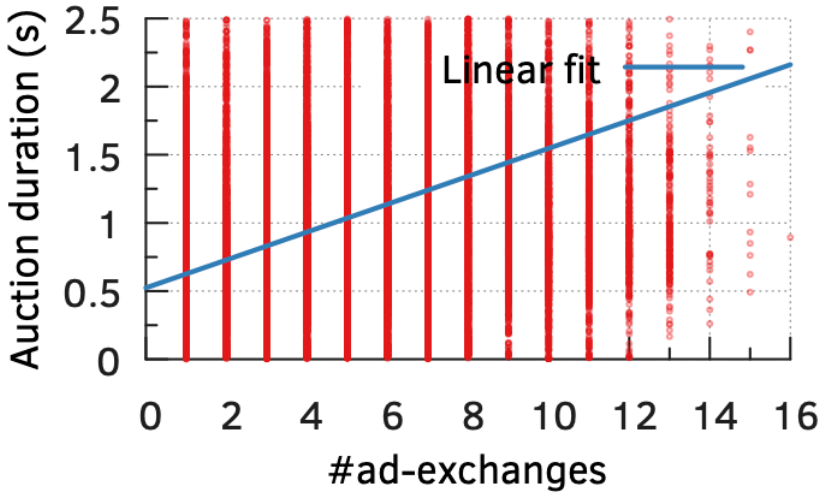


The Revenue-Latency Tradeoff

- Contacting more exchanges increases CPM for an ad slot
- Going from 1 to 8 exchanges doubles median CPM

- But also increases auction duration
- Delay in showing ads = bad user experience, perhaps lower click rate

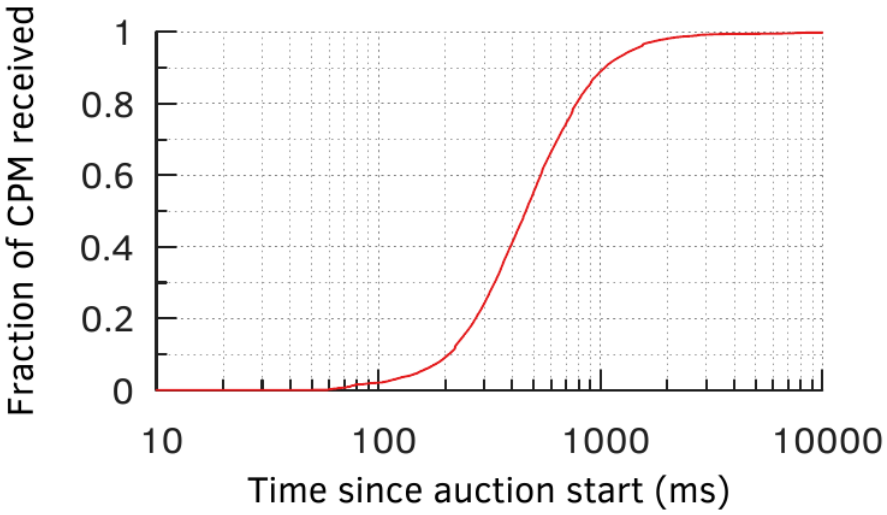


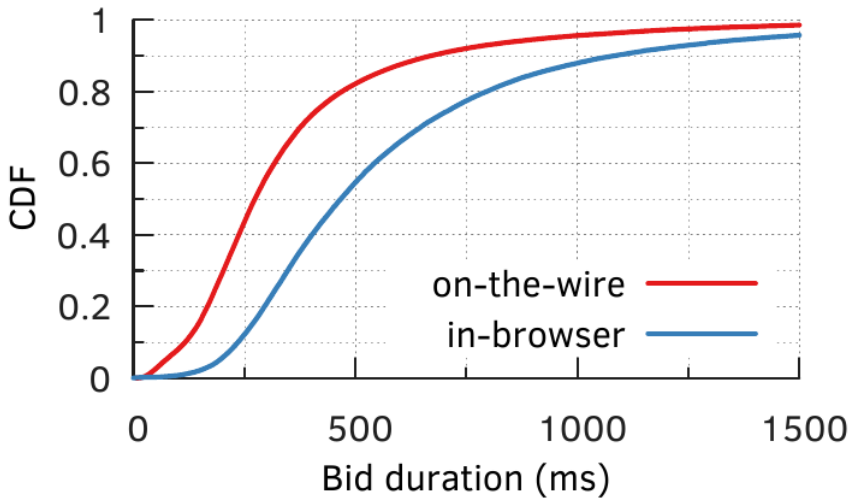


Latency Breakdown

- Time wasted on waiting for bids that will probably not alter the auction result

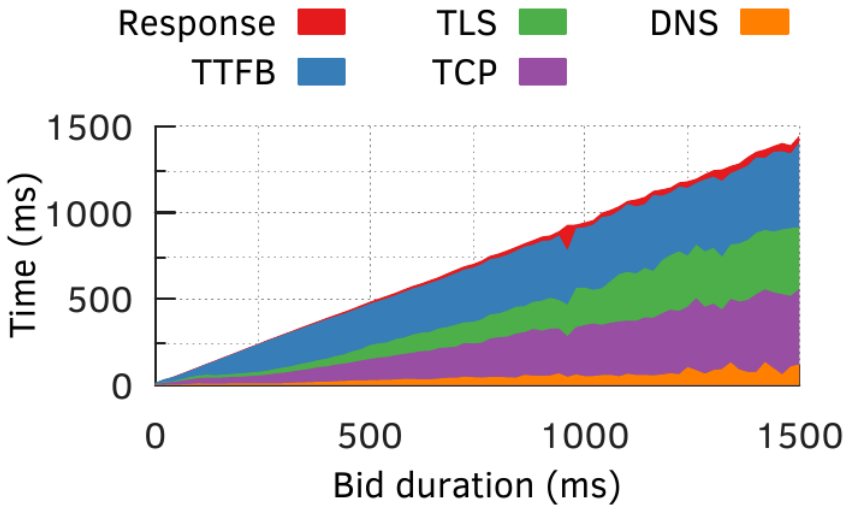
- Prioritizing other content, inefficient JavaScript implementations, even synchronous.
  - Contributes 174ms in the median





Latency Breakdown

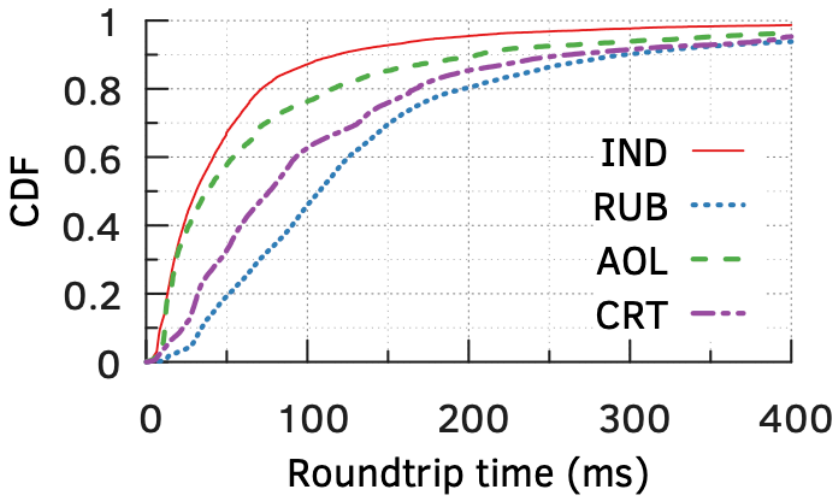
- 60% requests made on pre-existing, persistent connections
  - median duration is 230 ms
  - Time To First Byte (TTFB) dominates
- For the 40% non-persistent
  - median duration is 352 ms
  - TCP and TLS handshakes are 38% in the mean
  - Lack of support for low-RTT protocols. TLS 1.3 (11.4%), QUIC (6.6%), TCP Fast Open (76% but tricky)



Exchange Infrastructure

- Distributed deployments:
  - Index Exchange (IND): 88
  - Rubicon (RUB): 20
  - (AOL): 20
  - Criteo (CRT): 20

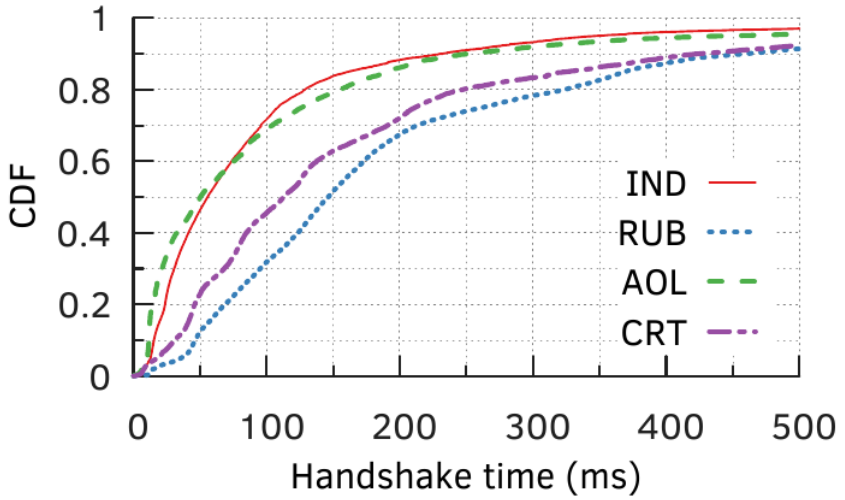
- Sometimes bad routing by ad exchanges
  - Large RTTs
  - Large variation in RTTs for users in the same city against one exchange

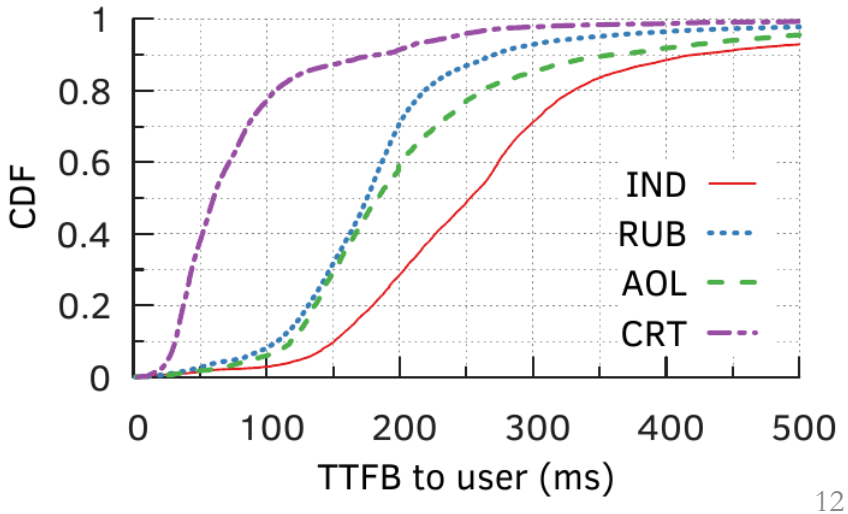


Exchange Infrastructure

- CRT, AOL gain in handshake time by supporting TLS 1.3

- TTFB dominates for most auctions
  - CRT has huge advantage
  - IND suffers
  - Unknown reasons, no visibility





Conclusions

- The revenue-latency tradeoff is valid
- Inefficiencies at the implementation and infrastructure levels
- Exchange-side auctions can be optimized
- Low RTT protocols and enhancements should be adopted

- Header bidding latency is not a fundamental problem

Future Work

- Increase measurement coverage
  - From ad exchange perspective
  - Revenue comparison with traditional real-time bidding

- Privacy-preserving advertising
  - Browser is in control
  - Store targeting information locally, send with ad requests
  - Like Privad, Brave Ads

Thank you!

Questions?