"What's the Score?" : A First Look at Sports Live Data Feed Services

Pengfei Wang*† and Aleksandar Kuzmanovic†

*Software College, Northeastern University, China

†Department of Electrical Engineering and Computer Science, Northwestern University

wangpf@stumail.neu.edu.cn, akuzma@northwestern.edu

***Abstract*—The significant interest that sports fans show for live game events, coupled with the major relevance that such information has for the online betting industry, singles out the *live sports data* as "the most important secondary information in the world." As a result, a set of specialized — sports data feed service — providers, has emerged. In this paper, we devise a methodology to evaluate such services in terms of speed and accuracy, at scale. We provide, to the best of our knowledge, the first measurement study of such services. By obtaining a direct access to the data feed of a leading sports data provider, we manage to assess the impact of different entities present in the data delivery chain. By measuring National Basketball Association (NBA) and English Premier League (EPL) live games from 40 sports websites, associated with 3 different data feed providers, we find that: (i) the direct data feed that we evaluated is systematically faster than a live cable TV broadcast provider, (ii) the variance of delays significantly increases once the feed gets redistributed by sports websites, and (iii) there exists an order-of-magnitude discrepancy in terms of delay, accuracy, and data diversity among different providers.**

I. Introduction

Hundreds of millions of people around the world follow various sports events, including football (soccer), basketball, baseball, American football, hockey, rugby, tennis, cricket, and more. However, not always is it possible for everyone to watch live sports events. For example, due to the time difference, soccer games in elite European competitions are held during peak working hours in the US. Similarly, games in many sports leagues are held at the same time, making it nearly impossible to watch them concurrently. Yet, knowing the actual scores in real time is often considered the most precious information by passionate sports fans. Likewise, accurate and timely information about games' scores and other events could help the sports betting industry, which is a growing stakeholder in the $423B global gambling market [10], provide a better Quality of Experience to users.

Given the significant interest in sports data, many websites provide real-time game score and other game-related information. However, independently obtaining and comprehensively maintaining sports data for each individual website is impractical, costly, and inefficient. Indeed, it is virtually impossible for each website to assign reporters to each of the games, reach agreements with sports associations, *etc*. As a result, specialized sports data feed services have emerged. Such services collect, maintain, and provide real-time data feeds to numerous websites. Despite the major role that such

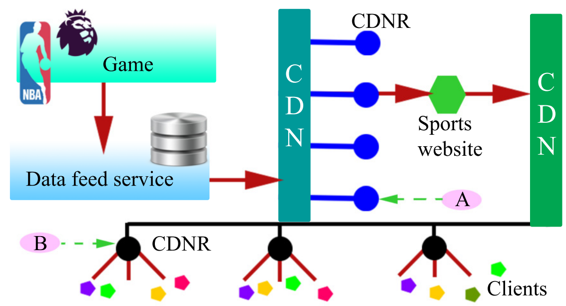

Fig. 1. The end-to-end sports live data feed distribution process: CDNR denotes a CDN replica; vantage point A measures a data feed "directly," from a feed's CDN replica, while B measures it "indirectly," via a sports website's CDN replica.

services have for the millions of sports fans and the online betting industry, to the best of our knowledge, the absolute and relative performance of such services is unknown, both in terms of speed and accuracy.

In this paper, we provide a first measurement study of live data feed services. To that end, we devised methods to capture and analyze sports data at scale. While our methodology can be generalized and applied to other kinds of data feed services which update data in real time (*e.g.*, weather, traffic, stocks, exchange rate), we exclusively focus on sports data in this paper. In particular, we devise a methodology to assess the speed and accuracy of live data feed services. While this might appear as a trivial problem at a first look, we argue that is not the case by outlining the key challenges.

The first challenge is what, how, and where to measure. Figure 1 shows the entire process of delivering real time information to the users via data feed services. The service gathers live sports data from a match, stores it in its database and delivers the data to its customers (*e.g.*, a website) via a content distribution network (CDN). The website gathers the real time data from the feed and updates its own database in order to further distribute the data to its users. Our study shows that almost all sports websites are also hosted on CDNs, hence the endpoint users will get the data via CDNs. Necessarily, the CDNs used by sports feed providers and websites may or may not be the same. *Our first non-technical contribution* lies in securing (through collaboration) a measurement point directly from a leading sports data feed provider, *i.e.*, in the same way a client website gets it. This helps us decouple and quantify the latency induced by the sports data feed provider and its CDN from the latency induced by any individual website and its CDN.

The second challenge lies in estimating the *absolute delays* induced by the system shown in Figure 1. Such estimates are essential metrics used in the sports betting industry. However, apparently straightforward methods to solve the problem failed. In particular, contrary to our expectations, we found that the sports data feed that we evaluated was providing data *faster* than the *live cable TV broadcast* provider was. To address this challenge, we devise a method to estimate the end-to-end absolute delays by utilizing *relative delays* extracted from archived videos, an approach that is both accurate and scalable. In particular, the relative time intervals are captured from the archived game video by utilizing Optical Character Recognition (OCR). Then, by utilizing simple event timestamp shifting manipulations, we manage to estimate the ground-truth events' timestamps. In addition, we define a set of metrics to estimate the data feed services' accuracy.

Our key findings are the following: (i) The sports live data feed that we directly evaluated can be quite fast, *e.g.*, systematically faster than the cable TV providers for NBA games. (ii) The variance of delays of the evaluated data feed is rather small and independent from a particular location in the world where the delay is measured, which means that the underlying system distributes the content well to the client websites. (iii) However, the necessary middle point (sports websites) often brings a rather significant variance in terms of end-to-end delays. (iv) Nonetheless, by combining information from multiple websites associated with a single data feed provider, it is possible to fairly accurately approximate a feed service performance even without a direct access to it. (v) The results among different sports data feed service providers show significant differences in terms of delay (one order of magnitude), accuracy, as well as non-uniform performance for different sports.

II. DATA

Here, we explain the different types of data and how we collect it.

Sports. The first challenge was to select the sports to measure. Given that different types of sports and associated games have diverse event features, we choose two popular, yet totally different sports, basketball and soccer. In particular, we focus on the National Basketball Association (NBA) and English Premiere League (EPL) games, which are popular and whose scores are reported by numerous websites. The rhythm of basketball is very fast, and scores are changed very frequently. While the score changes far less often in a soccer game, we found that sports data feed services are reporting a rich set of events that we evaluate. This diversity helps us comprehensively evaluate our delay and accuracy metrics.

Measurement points. We focus on three different data feed services and on 40 associated websites that use these three services. Each of the 40 websites is associated with a single sports data feed. We have one data feeds service to access, actually we weren't able to get all data feeds services' permission. But we have 40 sports websites knowing which data feeds service they are using. As mentioned above, we have a direct access to one of the data feed services. That measurement point is denoted as A in Figure 1. We obtain the rest of the data by measuring the 40 websites from the measurement point B, as shown in Figure 1.For each event, and for each measurement endpoint, we record the event identity and when it is shown (timestamp).

TABLE I
PLANETLAB ENDPOINTS LOCATIONS

Location	ID	Location	ID	Location	ID
USA-Cntrl.	L1	USA-West	L2	Spain	L3
N. Zealand	L4	USA-East	L5	Germany	L6

After the observation of data feeds service and websites, we find they are usually deployed in CDNs. It can solve the bottle neck of the Internet partially. It also costs time to update real time data into all CDN replicas. Meanwhile, these websites are mainly for users of different locations since they are suffixed with country level domain (e.g. .es, .ca, .fr, .au, .hk, etc.).

Measurement infrastructure. To overcome the delay induced by CDN replicas or the bottleneck of Internet, we measure the data feed service (measurement point A) and sports websites (measurement point B) from the six distinct locations shown in Table I, using Planetlab endpoints. Planetlab [3], in its own words, "is an open platform for developing, deploying, and accessing planetary-scale services." Most sports websites to be measured are located in America or Europe, so we use at least an America endpoint and a European one, and the third one is mainly based on the suffix of website. To record events' timestamp accurately, the endpoints' system time should be synchronized. To that end, we use the Network Time Protocol (NTP) [8]. The timestamp of an event is generated with Unix time which is defined as the number of millionseconds that have elapsed since 00:00:00 Coordinated Universal Time (UTC), Thursday, 1st January 1970 [1].

Data fetching methods. Crawling websites of getting events information is another step that can bring time error due to the complexity of modern websites. Here, we explain how we fetch data from the sports websites. The first challenge is that the information about the events for a game is usually not in the initial HTML document. Hence, crawling only the initial HTML document like a search engine won't work for some sports websites. An easier method to obtain the events information is to load the entire webpage while downloading its elements and render it as a browser. But, in order to render a webpage, we need to download additional corresponding files (*e.g.*, pictures, CSS, JavaScript, *etc.*) and load them. After observing all sports websites during a game, we find a third — light-weight — way to crawl using Ajax requests. The three methods are:

Initial HTML document crawling: This method is suitable for the sports websites where events information is stored in the initial HTML document (*e.g.*, .../index.html). We thus only need to download the initial HTML document. Discovering the corresponding URL is straightforward. The events information can be extracted with a DOM tree analysis.

Ajax crawling: Asynchronous JavaScript And XML (Ajax) is a technique that can fetch data from the server without refreshing the entire webpage. We observe that a number of sports websites utilize the Ajax technique to update sports events information periodically to clients even when users are not refreshing the webpages. The Ajax response is typically generated in a JSON format, which is lightweight, hence easy to crawl. However, singling out the particular requests of interest isn't always easy. This is because often a number of different requests gets exchanged while loading a website. We apply the following heuristics to detect the targeted requests: The request typically has the following features: it repeats periodically; the URL sometimes ends with ".json"; the "Content-Type" of the response is "application/json." For crawling, we need to be able to replay such requests. While most of the Ajax requests can be replayed, not all can be. In particular, some website developers add security mechanisms such that one cannot send the request even when the URL is known. For example, one sports website uploads the request with a key parameter which is encrypted with a public key. The request is initialized from a javascript which is unreadable due to compression. Hence, we are unable to replay such requests.

Page render crawling: In the scenarios where the information is unaccessible in the initial HTML document, and cannot be replayed as the Ajax request, we utilize a third method — page render crawling. This method loads a webpage entirely. To scrape sports events automatically, we use PhantomJS, which is a headless WebKit scriptable with a JavaScript API. In addition, we record the time when we send the first request as the timestamp of an event.

Obviously, the first two crawling methods are more accurate and much easier for fetching events information since only one resource is fetched from the website server, and the page render crawling is suitable for all websites but may not be as good as the other two since it needs more time to load webpages fetching various kinds of resources. Eventually, we end up with 18 websites using initial HTML crawling, 16 using Ajax crawling, and 6 render crawling. With the above steps, we can obtain the events information measured from a data feed service and sports websites by endpoints, *i.e.*, at vantage points A and B shown in Figure 1. Below, we explain how we obtain the ground-truth data.

III. Methodology

A. Estimating Absolute Delays

Comparing relative performance among different data sources, *e.g.*, data feed providers or websites, is a straightforward task. In particular, once the measurement vantage points are synchronized, concurrently fetching data from multiple sources provides insight into the relative performance, *i.e.*, which one is faster and by how much. However, estimating the *absolute delay*, *i.e.*, the time elapsed from the time an event happens until it is reported by a source, is far more challenging. The key challenge lies in estimating when did an event actually happen. In an attempt to gain knowledge about the ground truth, we first utilized a non-scalable, yet (we hoped) accurate method to estimate ground truth – live TV.

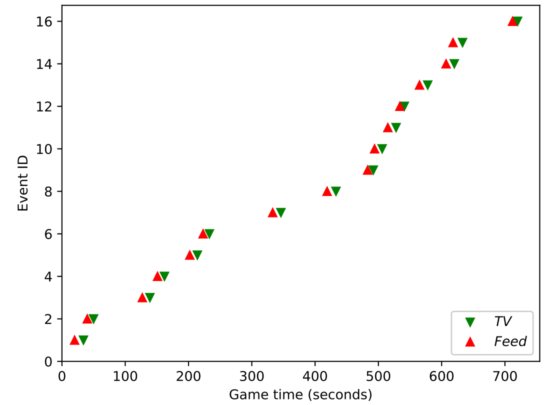

Fig. 2. Events timestamp comparison between a Live TV and a Data Feed service. Data Feed service systematically faster than Live TV.

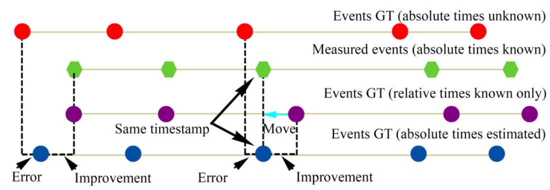

Fig. 3. The process of estimating absolute end-to-end delays using groundtruth (GT) relative time intervals.

We conduct an experiment where we concurrently record scores of an NBA game both from a live cable TV broadcast and a sports data feed that we have access to. Figure 2 shows the results. To our surprise, we find that the data feed service systematically outperforms the live TV results by 7 to 14 seconds. The reason for this unexpected behavior is that TV programs intentionally deploy such delays in an attempt to prevent profanity, bloopers, violence, or other undesirable material from making it to the air. The effects for us, however, are negative — we are unable to use live TV data as a ground truth to estimate absolute delays.

To address the problem, we continue as follows. Our key observation is that we can still accurately estimate the relative delays between events. We will demonstrate below that we are able to utilize this information to estimate absolute events' timestamps, and hence absolute end-to-end delays. Thus, instead of utilizing live data streams or TV, we utilize *archived videos*, widely and freely available online, to evaluate live data feed services. To that end, we capture the relative time intervals between successive events in a game, using Optical Character Recognition (OCR) (as we elaborate in detail in Section III-B).

Figure 3 depicts the procedure of estimating absolute end-

Fig. 4. The progress of capturing score-related events from an archived video.

to-end delays using accurate relative delays. The first line shows the ground-truth events (red dots), whose absolute timestamps are *unknown*, hence we aim to estimate them. The second line shows the measured events (green dots), with *known* absolute timestamps. The third line shows ground-truth events with accurate relative delays (extracted from a video), such that the timestamp of the first event is aligned with the corresponding event in the measured events data set. The figure clearly shows that such a setup isn't realistic, given that many events on the third line happen *later* than the corresponding events in the measured data set. This implies that the absolute timestamps for the third line are clearly misplaced. In an attempt to remove this un-natural setup, we shift the events shown in the third line to the left, thus creating events shown in the fourth line, which corresponds to the estimated absolute ground-truth line. In this case, we enforce the following two conditions: (i) no timestamp of any event in the estimated absolute ground truth data set can be greater than the corresponding event in the measured data set, and (ii) at least one pair of events from the two sets is aligned with each other.

As shown in Figure 3, in this way, we provide the best possible estimate (line 4) of the absolute ground truth of events, making it quite close to the actual ground truth (line 1). However, there necessarily exists an error in this process, shown in the figure. As we explained above, this error does *not* affect our estimates of relative performance of different data sources. Moreover, we argue that the error in this case is minimal. Indeed, the measured events (line 2 in the figure) are already faster by approximately 10 seconds on average than the Live TV. We will further show (Section IV-A), that the measured real-world results lag by at least 6 seconds behind our estimated ground truth. This indeed leaves little to no room for the necessary, but in this case apparently minor, error.

B. Fetching Data from the Video

Here, we explain how we extract scores from an archived video automatically. Automatic data extraction is essential for scaling purposes. Figure 4 shows this process. We utilize the technique known as Optical Character Recognition (OCR) [9] to extract score events from an archived sports video. Each 200 ms, we crop the score board part of the video making the video much smaller, and extract images from the cropped video. Then, we apply image processing (*i.e.*, binarization, *etc.*) to make the image more readable by a machine. Finally, we utilize Tesseract [11], an advanced open source OCR engine, for data recognition.

While the above method is suitable for capturing the change of score events, it is incapable of extracting, in a simple way, other game-related events. For example, fouls, free-kicks, yellow cards, *etc*, all of which are regularly reported in a sports data feed. Moving beyond manual data mining in this case requires more advanced image processing techniques. Alternatively, extracting the information from *audio* might be a simpler approach. Both of these potential techniques are, however, beyond the scope of this paper.

C. Delay and Accuracy

The criteria we design to quantify the performance of sports data feed services should be universal for all sports, so we define the delay and accuracy which are pervasive. Necessarily, we define delay as the time latency between the corresponding (same) events in a measured data set and the estimated ground truth. Since not all events may be present in the measured data set, we compute delay only when such events are present.

We define two accuracy measures as follows. Denote by S the ground-truth events set, and by S' a measured events set. Then, we denote by D the accuracy ratio, defined as the ratio of captured events relative to all the events. Formally,

$$D = \frac{|S \cap S'|}{|S|} \tag{1}$$

For example, D =1.0 implies complete accuracy, while D <1.0 means that the set misses some events; D =0 implies that all events are missing. Next, we define E as the fraction of the number of events which are in the measured set S' but not in the ground-truth set S, normalized to the number of elements in S. Formally,

$$E = \frac{|S' - S|}{|S|} \tag{2}$$

Hence, E measures the fraction of erroneous events, where false events show up in the measured set, yet not in the ground-truth set. Our results below show that in our measurements of NBA and EPL, we didn't find any error events, but we certainly did find a significant number of missed events.

IV. Results

Here, we show the results. As outlined above, we conduct measurements from the vantage points shown in Table I. We directly measure one data feed, and further measure 40 web-sites that provide sports data information, originally fetched from three independent data feed providers. We conduct our measurements in April and May 2017, measuring the end of the regular season and playoffs in the NBA, as well as the end of the EPL season.

A. Direct Data Feed Results

Initially, we evaluate the direct data feed measurement point, shown as point A in Figure 1. By utilizing our methodology, we estimate ground truth and the corresponding absolute-delay estimates, and Figure 5 shows the results. The lag of the feed

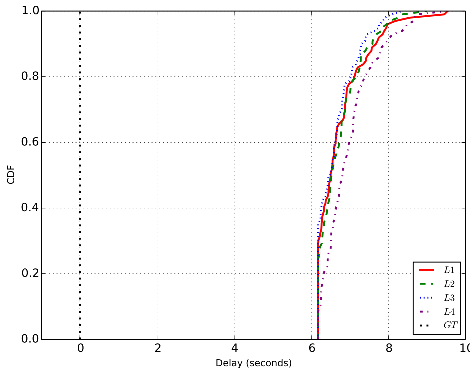

Fig. 5. The CDF of estimated absolute delays measured directly from a data feed from different locations for an NBA game.

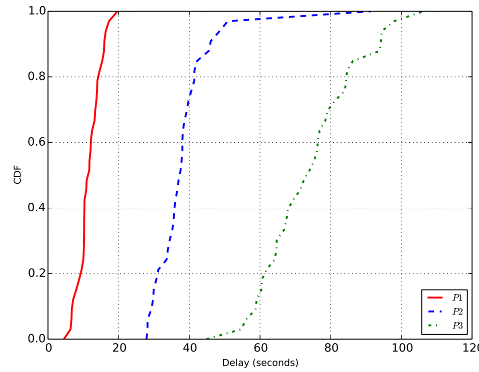

Fig. 7. The CDF of delays for three data feed providers, indirectly estimated by using the best measurement results from 40 associated websites for a number of NBA games.

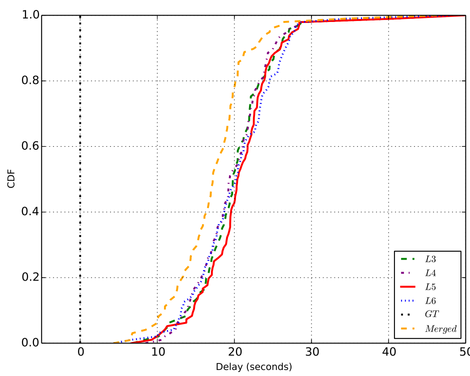

Fig. 6. The CDF of estimated absolute delays measured for a single sports website from different locations for an NBA game.

behind the estimated ground truth is between 6 and 10 seconds. Note that we record the event timestamp with a millisecond precision in all experiments. The figure shows that measuring the data feed from multiple vantage points does *not* change the result. Except for the New Zealand vantage point, L4, which does experience a slightly longer delays, the performance is otherwise very unison. This shows that the particular data feed provider successfully distributes the sports data via a CDN. Finally, we observe that the *variance* of the measured results is rather small, since the bulk of the distribution lies between 6 and 7 seconds, and the tail isn't long. This implies that the direct feed indeed has a reliable performance.

B. Indirect Data Feed Results

Here, we first analyze the performance experienced by the clients, who access the websites which provide sports events. In particular, we measure the 40 sports websites from the vantage point B, as shown in Figure 1. Then, we explore if it is possible to estimate the performance of the sports data feed providers *indirectly*, by aggregating information obtained from the websites. Indeed, not always is one able to get access to a sports data feed. Still, being able to assess the performance of data feed providers indirectly is certainly valuable, since it enables accountability yet without relying on the direct cooperation with the feed provider.

To start with, we first measure a single website which uses the data feed provider earlier evaluated in Figure 5. Figure 6 shows the results. The first insight is that the average delay necessarily increases relative to the delay shown in Figure 5. In particular, the median of approximately 7 seconds when measured directly from the feed, jumps to approximately 20 seconds, when measured via the website. Moreover, the *variance* of the result significantly grows relative to the direct feed measurement. Figure 6 shows one additional line, termed "merged". It shows the potential to utilize the spatial diversity of measurement vantage points, and select the *best* result for each individual event. The result shows that even for a single website, the median delay can be improved by as much as 5 seconds in the median case. Hence, we attempt to use all the 40 websites to see if we can approximate the date feeds' original performance. We anonymize all three data feeds providers because we have made a promise of confidentiality with one of them.

Figure 7 shows the results. We made statistics (not shown in the paper because of the page limit) between the direct and indirect measurements, and the most striking result is that while the original performance of the data feed provider P1 wasn't fully accurately re-established, we can get that the direct measurement result of the P1 in Figure 7 pretty fairly reassembles the one shown in Figure 5. This means that indirectly measuring data feed providers is feasible. Figure 7 further shows that the performance of the second provider lags substantially behind the first provider, and has a fairly long tail. Likewise, the third provider has the median delay approximately an order of magnitude longer than the first provider.

C. EPL and Accuracy Results

Here, we first focus on the delay performance for EPL. For space constrains, we provide only the median delay

TABLE II
THE ACCURACY RESULTS FOR NBA GAMES

	Acc. Ratio	Error	Number of events
P1	0.999	0	1,150
P2	0.933	0	1,074
P3	0.606	0	698

TABLE III
THE ACCURACY RESULTS FOR EPL GAMES

	Acc. Ratio	Error	Number of events
P1	0.996	0	487
P2	0.665	0	325
P3	0.851	0	416

performance for the three providers: 66.04s for P1, 30.16s for P2, and 119.24s for P3. We provide three insights here. First, contrary to our expectations, we find a plenty of events reported from a game, including free kicks, yellow or red cards, substitutions, goals, fouls, and shots. Second, the number of reported events (further analyzed below) apparently impacts the delay performance. In particular, while P1 and P3 provide a diverse set of events, P2 provides a limited set of events, which can help with latency. Third, we hypothesize that P2 might utilize a mechanism to automatically report events quickly, which further might explain its performance.

Tables II and III show the accuracy metrics, defined in Section III-C, for NBA and EPL, respectively. Tables also show the total number of measured events. We find no errors. However, there exists a number of missing events, reflected in a moderate accuracy ratio, *i.e.*, 0.606 for P3 for NBA, and 0.665 for P2 for EPL.

V. Related Work

Here, we briefly survey related work in web crawling and CDN measurements, both of which relate to our work. Various methods have been designed in the past to crawl websites. Websites can be crawled with various kinds of methods. A browser-based crawling method is used in [14], where the entire web page rendering progress is recorded automatically with a general browser (*i.e.*, Chrome). PhantomJs is widely used to render and extract information from websites since it can do bulk background crawling automatically by controlling a webpage's DOM via scrips [4], [5]. While these methods are integrated in our system, our key contributions, summarized above and below, lie beyond web crawling.

Measuring CDN performance has been an active research area since the CDN inception, nearly two decades ago. The content delivery networks utilize a number of globally distributed servers to provide users with a low-delay access to websites with a reliable access. Early work demonstrated that selecting an optimal CDN replica is a challenging task [7]. Over the years, numerous other measurement studies, *e.g.*, [2], [6], [12], [13] provided deep insights into CDN performance. Our sports data feed measurements indirectly evaluate the underlying CDNs performance. We show that chained CDN architecture, shown in Figure 1, inflates both the mean and the variance of the end-to-end delay.

VI. Conclusions

In this paper, we conducted the first measurement and analysis of live sports data feed services. By conducting measurements directly from a feed and indirectly via websites, we made several contributions: (i) a methodology for estimating absolute delays using only relative delay measurements obtained from an archived video, (ii) the first insights about the role of different entities in the sports data distribution, (iii) showing that a direct data feed can be well-approximated via indirect sources, and (iv) demonstrating that there exists an order-of-magnitude discrepancy in delay and accuracy among current providers. The future research avenues, for ours and other communities, include: (i) design of a fully automated sports data feed auditing system, (ii) development of advanced automatic event extraction methods from video and audio, (iii) explore the applicability of the proposed methods in other data feed services, and (iv) design more efficient "multi-tenant" live data feed distribution architectures.

References

[1] Unix time. 2017. http://unixtime.info/.

[2] M. Calder, A. Flavel, E. Katz-Bassett, R. Mahajan, and J. Padhye. Analyzing the Performance of an Anycast CDN. In *Proceedings of the 2015 ACM Conference on Internet Measurement Conference*, pages 531–537. ACM, 2015.

[3] B. Chun, D. Culler, T. Roscoe, A. Bavier, L. Peterson, M. Wawrzoniak, and M. Bowman. Planetlab: an overlay testbed for broad-coverage services. *ACM SIGCOMM Computer Communication Review*, 33(3):3–12, 2003.

[4] A. Hannak, P. Sapiezynski, A. Molavi Kakhki, B. Krishnamurthy, D. Lazer, A. Mislove, and C. Wilson. Measuring personalization of web search. In *Proceedings of the 22nd international conference on World Wide Web*, pages 527–538. ACM, 2013.

[5] A. Hannak, G. Soeller, D. Lazer, A. Mislove, and C. Wilson. Measuring price discrimination and steering on e-commerce web sites. In *Proceedings of the 2014 conference on internet measurement conference*, pages 305–318. ACM, 2014.

[6] C. Huang, A. Wang, J. Li, and K. W. Ross. Measuring and evaluating large-scale CDNs. In *ACM IMC*, volume 8, 2008.

[7] K. L. Johnson, J. F. Carr, M. S. Day, and M. F. Kaashoek. The measured performance of content distribution networks. *Computer Communications*, 24(2):202–206, 2001.

[8] D. Mills, J. Martin, J. Burbank, and W. Kasch. Network time protocol version 4: Protocol and algorithms specification. Technical report, 2010.

[9] S. Mori, H. Nishida, and H. Yamada. *Optical character recognition*. John Wiley & Sons, Inc., 1999.

[10] B. Pempus. Report: Global Gambling Market Was $423B In 2014. http://www.cardplayer.com/poker-news/18644-morgan-stanley-global-gambling-market-was-423b-in-2014.

[11] R. Smith. An overview of the Tesseract OCR engine. In *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, volume 2, pages 629–633. IEEE, 2007.

[12] A.-J. Su, D. R. Choffnes, A. Kuzmanovic, and F. E. Bustamante. Drafting behind Akamai: Inferring network conditions based on CDN redirections. *IEEE/ACM Transactions on Networking (TON)*, 17(6):1752–1765, 2009.

[13] S. Triukose, Z. Wen, and M. Rabinovich. Measuring a commercial content delivery network. In *Proceedings of the 20th international conference on World wide web*, pages 467–476. ACM, 2011.

[14] M. Varvello, J. Blackburn, D. Naylor, and K. Papagiannaki. EYEORG: A Platform For Crowdsourcing Web Quality Of Experience Measurements. In *Proceedings of the 12th International on Conference on emerging Networking EXperiments and Technologies*, pages 399–412. ACM, 2016.