Application-Specific Secure Gathering of Consumer Preferences and Feedback in ICNs[*]

Reza Tourani
New Mexico State University
rtourani@cs.nmsu.edu

Satyajayant Misra
New Mexico State University
misra@cs.nmsu.edu

Travis Mick
New Mexico State University
tmick@cs.nmsu.edu

ABSTRACT

The shift from the host-centric to the information-centric paradigm results in many benefits including native security, enhanced mobility, and scalability. The corresponding information-centric networking (ICN), also presents several important challenges, such as closest replica routing, client privacy, and client preference collection. The majority of these challenges have received the research community's attention. However, no mechanisms have been proposed for the challenge of effective client preferences collection.

In the era of big data analytics and recommender systems customer preferences are essential for providers such as Amazon and Netflix. However, with content served from in-network caches, the ICN paradigm indirectly undermines the gathering of these essential individualized preferences. In this paper, we discuss the requirements for client preference collections and present potential mechanisms that may be used for achieving it successfully.

CCS Concepts

•**Security and privacy** → **Security protocols;** *Web protocol security;*

Keywords

Information-centric networks; preference collection; recommendation systems.

1. INTRODUCTION AND MOTIVATION

Several service providers (e.g. Amazon, TripAdvisor, Netflix) rely on the availability of client preferences in order to serve each individual client's needs, and to assess the needs of the customer base as a whole. In Information-centric Networking (ICN) such preferences cannot always be ascertained, as ubiquitous caching eliminates the need for interaction between the client and provider.

Although advantageous in terms of network latency and load, this is detrimental to a content provider's ability to assess user preferences and provide effective service. More specifically, the content provider will be ignorant of the preferences of the clients whose requests are satisfied by in-network caches. This is an important facet that needs to be addressed in ICNs.

To elaborate more, we present two examples to review the client-producer interactions of two major content providers, Netflix and Amazon. In Netflix [2, 9], when a client's player is started it contacts the Netflix server (which resides in the Amazon cloud) to authenticate itself and request a video stream. Upon receiving the client's request, the Netflix server generates and delivers a manifest (meta-data) to the client, through an SSL channel. The manifest file is client-specific (generated specifically for the client according to its capabilities) and includes information such as different video/audio qualities, list of CDNs, chunks' URLs, and metadata that enables fast forward/rewind (trick play). The client's player requests the content from the best ranked CDN to start streaming. It periodically sends logs and heartbeat messages to a Netflix control server in the cloud. In this architecture, the client's content requests are recorded by the control server, helping Netflix to track the client's preferences and statistics. Hulu, another major multimedia content provider, exploits the same approach as Netflix [9].

Amazon is a bit different. Amazon's static content is stored in a CDN; once a client logs in to her account, the process that logs her activities and preferences is triggered and every subsequent search or purchase is recorded. By employing techniques such as collaborative filtering and content-based filtering, Amazon provides targeted recommendations according to the client's preferences and search history. It has been shown in [1] that over 50% of HTTP web pages are non-cacheable (cache lifetime = 0). Furthermore, many web pages utilize analytical tools such as *Google Analytics*, which employ JavaScript snippets and cookies to enable the extraction of traffic information. Due to ICN's pervasive caching, clients' requests are either satisfied by the content providers (or their CDNs) or in-network caches. If data is retrieved from the provider directly, the preference feedback is easily obtained, however data provision from in-network caches makes feedback acquisition more challenging. If authentication or authorization are not needed for data access, the client is under no obligation to send its preferences to the providers. The routers that satisfy the content request are also under no obligation for sending any preference feedback.

[*] This work was supported in part by the US National Science Foundation grants 1241809 and 1345232 and US DoD grant W911NF-15-1-0393. The information reported here does not reflect the position or the policy of the federal government.

ICN'16, September 26 - 28, 2016, Kyoto, Japan

© 2016 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ISBN 978-1-4503-4467-8/16/09…$15.00

DOI: http://dx.doi.org/10.1145/2984356.2984367

This will adversely affect today's online business model of feedback-driven customer service.

This problem has received scant attention in our research community. Katsaros *et al.* [7] explored information exposure of named content and proposed using ephemeral content names to necessitate a client-provider interaction. Employing this method, content providers are able to record the statistical information they require from clients, even if the content is cached under a stale name.

The main drawback of this scheme is that it undermines caching: temporary ephemeral names result in the content being useless after a name's expiration, even though the content itself is still useful. It also requires a trusted third party to validate the ephemeral names and the providers' identities. To the best of our knowledge, this is the only relevant work in the literature.

An in-depth exploration of this challenge is important to find a viable solution for client preference collection without undermining the principles of the ICN paradigm. Absence of such a solution would prevent content providers from adopting the ICN paradigm and undermine ICN's application in the Internet. This serves as the motivation for studying the problem. In this paper we raise the question: *What does it take to effectively collect clients' feedback and preferences in a network where content can potentially be delivered from any node?*

We explore the suitability of delegating the task of preference feedback collection to different network entities such as caching routers, clients, and ISPs, and evaluate the drawbacks of each assignment. We discuss potential approaches to feedback collection, including methods that extract statistics from existing communications (interest-content interaction), as well as other methods which require extra communication. We also present approaches that use manifests to enable providers to track their clients' preferences. Note that manifests are a special type of content object, representing metadata; these objects often obtain access control information, publisher identification, and content chunks hash digests for unstructured content items, and have been proposed to improve the flexibility of the ICN paradigm.

In Section 2, we discuss the impact of different content types on feedback collection, as well as the scope of the information which must be collected. Section 3 reviews possible information collection approaches which incur no additional communication overhead. In Section 4, we address the shortcomings of the previous section's approaches and elaborate on two manifest-based preference tracking and feedback collection models.

2. CONTENT CATEGORIZATION AND FEEDBACK SCOPE

In this section, we categorize all content items into four categories and discuss the properties of each category. Furthermore, we briefly discuss the information that is required for precise client tracking and effective recommendation. Figure 1 illustrates the content categorization and the properties of the content that belongs to these categories and their intersections. As it is shown, the static and dynamic content categories are disjoint, as are the public and private categories. However, the intersections of static content with the public and private categories form the static-public and static-private content categories, respectively. Similarly, dynamic content are either dynamic-public or dynamic-private. We first discuss the four basic categories and later elaborate on the properties of their intersections.

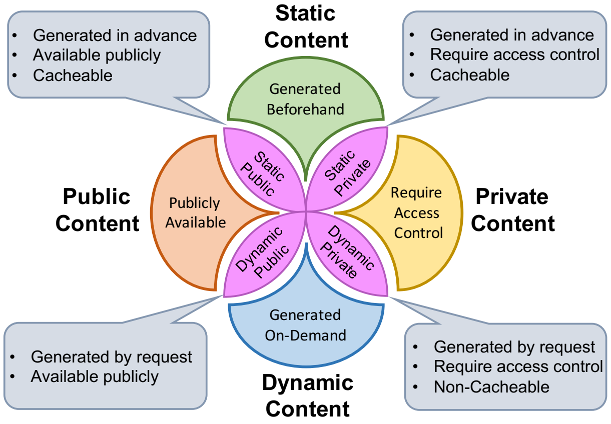

Figure 1: Content categories and their properties.

2.1 Categorization of Content

Content that belong to the *static content* category are pregenerated by the content provider and eligible to be cached at the intermediate routers for an indeterminate length of time. Subsequent requests for these content can be satisfied by caching routers without contacting the provider. In addition to content caching, request aggregation in ICN undermines information extraction and preference tracking for this category, especially for individualized preferences. We will discuss this in more detail in the next section.

In contrast to static content, *dynamic content* is generated upon a client's request. Content generated by VoIP services, targeted searches, and personalized online banking services belong to this category. Although these content can be cached, the positive impact of such caching is minimal — the content may be client specific, of low popularity, or subject to expiration. Nonetheless, the advantage of dynamic content is its inherent need for client-provider interaction. This allows the provider to profile the client's preferences and collect the required statistics.

The third category, *public content*, consists of those unencrypted content that are publicly available for all consumers. As access control is not required for this category, these content objects can be widely cached in the network then used to satisfy subsequent requests. Similar to the static content category, a request for a publicly available content can be satisfied by a caching router, making the task of preference tracking more challenging.

Private content, in contrast to public content, is only available to an individual or some subset of consumers, and is usually encrypted. In order to access a private content, a legitimate client has to successfully authenticate herself to either the content provider or an authentication server. This compulsory authentication (either at the provider or a third party) can be used to track the client's request. Personal banking data and subscriber-restricted multimedia content such as are representative of this content category.

Now we discuss the properties of the intersections of these categories in detail. Static-public content objects are widely available in network's caches, with open access permission for all clients. These properties make this category the most challenging for preference tracking purposes. Currently, this content category forms about two-thirds of the North American fixed network Internet traffic. Over 90% of Netflix traffic, as a major contributor of North American Internet traffic, belongs to this category [8]. In contrast, dynamic-private content are generated on-demand, upon a client's successful authentication. Both of these properties require the client-provider interaction, easily facilitating preference tracking.

Static-private content is cacheable, but has an authentication/authorization requirement that helps the providers to extract their clients' preferences even when the content is cached. However, the selection of the access control enforcement entity (content provider versus third party) affects the efficiency of the preference tracking mechanism. The efficiency is undermined if the access control enforcement entity is not obliged to provide access information to the provider. This group of content – usually encrypted for secure communication, authorization, privacy, and integrity – forms a smaller portion of the Internet's daily traffic [8], although it is increasing.

The encrypted Internet traffic in North America (fixed access) has increased from 29% in 2015 to 37% in 2016, which leaves about 63% of the fixed network traffic unencrypted. This is while 64% of Internet traffic is encrypted in the mobile access network (in the same region, in 2016). The dynamic-public content group, similar to the previous category, requires client-provider interaction, which makes preference tracking less challenging.

2.2 Collected Preference Data

A feedback message should contain adequate information about the requester and its requested content. The requested content name is available in the request packet. However, ICN eliminates any notion of the client's identity in its request, enhancing client privacy and enabling request aggregation. Lack of the client's identity in the request packet undermines the preference tracking mechanism, especially when preferences are needed for providing customer-specific service. We will compare different collection models in Section 3 from this standpoint.

A feedback message that contains the requester's identity or other sensitive information, however, needs to be protected. This protection is crucial in order to preserve the requester's privacy.

3. FEEDBACK COLLECTION AND DELIVERY

An efficient preference tracking mechanism in ICN requires the cooperation of clients. Ideally, a cooperative client following the protocol, forwards the essential information to the content provider. In this case, there is no need for many changes or additional functionalities on network entities, as the client forwards feedback messages to the corresponding content provider. However, this assumption is not acceptable in all scenarios, and without clients cooperation, there is no fine-grained information to be delivered to the content providers.

Thus, in this paper we explore different preference tracking mechanisms, where non-cooperative clients are assumed. This means that either a client should be asked explicitly for feedback in return for the delivered content or the network has to extract required information from the ongoing communication without further obligating the client to send feedback. In this regard, there are two main questions that needed to be answered for a practical design of such a mechanism. First, which network entity is in charge of collecting and delivering the statistical information and feedback to providers. Second, depending on the collector entity, what level of granularity the collected information can be.

Figure 2 illustrates different potential preference tracking approaches. We divide these approaches into two main classes: *manifest-free* and *manifest-based*. The manifest-free class includes approaches that do not require additional communications and can itself be divided into three subclasses, with different collector entities. The manifest-based class utilizes manifest files for preference and feedback collection and delivery. This class can be further divided into two subclasses depending on the entity that serves the manifest file. We will discuss this class of preference tracking in Section 4.

As depicted in Figure 2, this section reviews the *Manifest-Free* collection mechanisms. More specifically, we discuss the potential collector entities along with the approaches that can be adopted by those entities to gather preferences and potential pitfalls. Clients' feedback can be collected and forwarded to providers via three network entities: intermediate routers, clients, and ISP's collection servers (a third party entity). As mentioned before, we assume the non-cooperative client model and discuss the approaches that do not incur communication overhead. In other words, the collector entity has to extract information from the existing interest and data packets that flows in the network, between client and provider or intermediate caches.

3.1 Collection by Intermediate Routers

An intermediate router can act as the collector entity to extract and collect statistical information and preferences from the received requests. This router, upon receiving a request, collects the content name and other metadata, available in it. Following the conventional CCN/NDN forwarding model, the router creates a PIT entry and forwards the interest to the upstream router or updates the existing PIT entry, if applicable. It then periodically transmits the collected information to the corresponding content providers. However, this raises important challenges.

First, the computational overhead of information collection and processing at the intermediate routers undermines this model's scalability. Second, due to lack of client's identity in request packets, the collected information loses its per-client level precision. Furthermore, request aggregation in ICN undermines the precision of the collected information; one request in an intermediate router might represent a set of aggregated requests in its downstream router. Eventually, all the routers on the path to the content provider record the same event for multiple times.

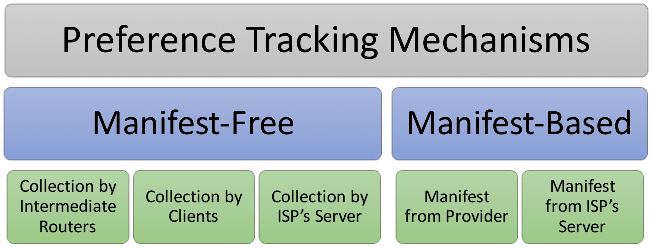
Preference Tracking Mechanisms
Manifest-Free
Manifest-Based
Collection by Intermediate Routers
Collection by Clients
Collection by ISP's Server
Manifest from Provider
Manifest from ISP's Server

Figure 2: Preference Tracking Model Hierarchy.

Therefore, there is a need for a feedback collection model, which helps the intermediate routers to avoid redundant information collection. In this context, the collection model defines the event by which an intermediate router is triggered to record the required information from the request packet. We envision two collection models by the intermediate routers, namely per-interest collection and per-hit collection.

As the names suggest, in the per-interest collection, every router will be triggered to collect information upon receiving a request. The per-hit collection model remedies the drawback of the per-interest model (i.e., redundant information collection) by delegating the collection task to the router that serves the content from its cache. The per-hit model generates more accurate statistics since only one router is recording each single event. Although the per-hit model solves this problem, the aforementioned challenges of intermediate routers collection (computation overhead, lack of client's credential, and request aggregation) remain unsolved.

In case that a content provider requires real-time statistical information, the router responsible for information collection has to deliver the information to the content provider per request. However, forwarding this request creates a PIT entry in the upstream routers and causes the content provider to return the requested content. To avoid such an effect, the serving router flags the request so the upstream routers only forward it toward the provider, who also considers it as a real-time preferences tracking message. To conclude, this scheme neither provides any notion of client nor produces precise statistics such as number of clients who requested a content. Hence, we move forward to discuss the possibility of having other network entities as the collector.

3.2 Forwarding by Clients

Forwarding of the statistical information and preferences by clients can be employed to address the drawbacks of the previous model. The advantage of this model is two-fold: collection of a more fine-grained, per-client statistics and eliminating the routers computational overhead. However, the main drawback of this model is non-cooperative clients can refuse to provide feedback and there is no mechanism to guarantee clients participation. Also, for each access the clients should be obliged to either contact the provider or transmit their feedback for each request packet.

The first solution to address this problem is for the provider to publish the entire content into the network except a small portion, which is required for content reassembly. For successful content reassembly, the client has to request this portion of the content, directly from the provider. Thus, the provider, upon recieving the request for the small content portion, logs the client's information for further processing. The question that needs to be answered for this model is the size of the content that remains at the provider, which directly affects the communication overhead. The main drawback of this model is that it is difficult to deter a client that has access to the complete content (including the small provider portion) from publishing this portion, which can undermine the approach. Although we note that the malicious client gains little from this.

Content encryption and access control enforcement is another method of forcing clients to contact providers. On this front, a provider encrypts its content with a unique key and publishes the content into the network. A client who retrieves the encrypted content is required to interact with the provider to obtain the decryption key; this provides an opportunity for the provider to track the client's preferences. This model is more efficient with private content that need authentication and authorization by default.

This mechanism may also be undermined if a client shares its decrypted content in which case other clients will be able to retrieve the decrypted content and avoid interaction with the provider. Again, the incentive for a client to do this is limited, most of these content are public anyways (most Internet traffic is public) and most private content is client specific, which the client does not wish to share. The main pitfall of this mechanism is its dependency on an always online authentication server (or a provider), which provides the key. Also, this mechanism, in spirit is not suitable for public content category (having private metadata for public content), which forms a considerable portion of the Internet traffic. For these reasons, we investigate the applicability of ISPs as the collector entities.

3.3 Collection by the ISP's Designated Server

In order to address the shortcomings of the previous model, in this subsection, we discuss the practicality of ISP as the collector entity. In this model, an ISP designates a server for preference tracking and statistical information processing purposes. In conjunction with previous interactive approaches (storing a portion of content or providing the decryption key), the server can be instructed to cache a small portion of each content or the content decryption keys. Thus, clients in an ISP interact with the server (operating on behalf of the providers) to reduce the fetch latency and network-core communication overhead, and improve cache utilization. Another benefit of this model is that there is no need for an always-online provider (or access control entity), which consequently promotes content accessibility and improves quality of service. In both cases (serving a small content portion or the decryption key) the server plays the role of an access control enforcement entity. Delegating the access control enforcement in ICN has been extensively discussed in the literature [3, 5, 6].

This approach is also in line with the model in which the intermediate routers are the collector entities. In this regard, the intermediate routers forward the collected information or the modified requests (adding a flag) to the server and offload the information processing to the server. The server processes the obtained information and either directly (per request) or periodically forwards them to the provider. However, this is not beneficial due to inaccuracy and insufficiency of the collected information by the intermediate routers.

4. MANIFEST-BASED APPROACHES

This section discusses the manifest-based preference tracking mechanisms, in which a client needs to obtain the manifest file to be able to request the content. A manifest file contains information, which helps the client to request the content. Figure 3 illustrates the CCNx manifest structure specification [4]. The CCNx manifest file is composed of a manifest name and a payload. The manifest payload itself contains a sequence of *Sections* (possibly empty). Each *Section* includes the content's chunk names and their first indices, their hash digests, and access control specifications.

Despite the communication overhead, which is incurred for obtaining manifest files in these mechanisms, requesting a manifest file is advantageous for both clients, for content retrieval and content providers, for client mining purposes. As per Figure 2, in the manifest-based preference tracking mechanisms, either the content provider directly transmits the manifest file to clients or this task is delegated to ISPs' designated servers. In the following, we review these two approaches in detail.

4.1 Requesting Manifest from the Provider

In this model, we assume that manifest files are not cacheable, and hence, every manifest request should be satisfied by its corresponding provider. In order to request a content, a client should initially send a manifest request for that content to the provider (this can be encrypted by the client with a key shared with the provider). Upon receiving the manifest request, the provider records the required information about the client and the requested content. We envision the manifest request carries client's credential, which will be extracted by the content provider for preference tracking purposes (similar to the Netflix client-specific manifest). Embedding client's credential is not only necessary for preference and feedback collection applications, but can be used for other applications such as manifest-based access control enforcement. After extracting client's information and preferences, the provider forwards the requested manifest toward the client. The client with the manifest then requests the actual content utilizing the metadata in the manifest's sections; this content can be served by a caching node or the content provider.

Despite the applicability and scalability of this model, there are challenges that needs to be addressed. The main drawback of this model is the requirement for the content provider's availability and extra latency for obtaining the manifest prior to content request. However, manifest has been used by the community in a variety of applications such as access control, content redundancy elimination, and adaptive video streaming. Thus, the additional latency can be omitted for a better cause. Another minor inefficiency of this model is in scenarios, where the content has not been cached in the network. In this case, the client's content request, after obtaining the manifest from the provider, will be forwarded to the content provider.

This is a wasteful practice as the provider could have forwarded the content along with the manifest upon receiving the manifest request (in case that the provider hosts the content and manifest in the same location). Although this scenario might rarely happen, it is still a challenge that needs to be addressed. In this regard, one naive solution would be forwarding the content along with the manifest for the first manifest request on every interface. This helps the downstream routers under every interface to cache the content, so they can serve the subsequent content requests from their caches. Eventually, the strong assumption of uncachable manifest needs to be revised for a scalable solution.

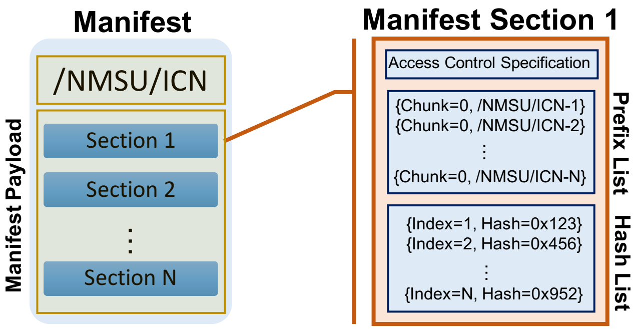

Figure 3: CCNx Manifest Structure Specification.

4.2 Requesting Manifest from the ISP's Designated Server

This model tries to address the strong assumption of manifest uncachability in the previous model. However, in this model, we propose a restricted manifest caching in the sense that only specific nodes are allowed to cache manifest files. Each ISP designates a server (group of servers) responsible for collecting statistics, feedback, and preferences in addition to caching and serving manifest files. For content retrieval, a client initially sends a manifest request towards the ISP's designated server. The server returns the manifest if it has been cached before, otherwise, the request will be forwarded towards the provider. In either case, the server stores the corresponding statistic for the client. It periodically delivers the collected statistics to the corresponding providers.

The manifest request may contain client-specific information, such as a temporary ID or a pseudonym, which is only known to the client and provider and serves to identify the client. Alternatively, the client's identity can be encrypted with a secret key, shared between the client and provider. Encrypting the client's credential preserves the client's privacy from the intermediate nodes while allows the provider to perform fine-grained preference tracking.

The main advantage of this model over the previous model is lower communication overhead and latency. Furthermore, this model does not need an always online content provider; this improves content availability and clients' quality of experience. However, this model is not suitable if the content provider requires real-time client's feedback. For real-time feedback, the designated server that satisfies the client's request with the cached manifest, has to instantly forward the manifest request towards the provider. This forwarded manifest-request should be flagged to prevent on-path servers from satisfying the request. The message can be tagged as a feedback message, so that the provider does not have to perform a manifest delivery. In both these models, the ISP's designated server will be the single point of failure and network bottleneck, which can be addressed with the addition of redundancy. Furthermore, the latency, although reduced because of caching, will still be more than the case where clients directly request their content without obtaining manifests.

5. CONCLUSIONS

In this paper, we discuss the challenges in collecting statistical information and clients' preferences in ICN paradigm, in which pervasive caching and requests aggregation hinder clients mining. We categorize content into four categories (i.e., Static, Dynamic, Public, and private) and elaborate on their properties. Furthermore, the potential collector entities have been investigated with their advantages and drawbacks.

Despite practicality of the mechanisms that involve content providers (by content encryption or partial content delivery), employing an ISP server seems more promising due to lower communication overhead and improved content availability. The manifest-based models facilitate content retrieval and help content providers to track their clients' behaviors and preferences more precisely. We believe that in conjunction with ISPs cooperation, manifest-based approaches can be utilized for an efficient and scalable client preference tracking mechanism.

6. REFERENCES

[1] Statistical HTTP Facts. http://httparchive.org/interesting.php.

[2] V. Adhikari, Y. Guo, F. Hao, M. Varvello, V. Hilt, M. Steiner, and Z. Zhang. Unreeling Netflix: Understanding and improving multi-CDN movie delivery. In *IEEE INFOCOM*, pages 1620–1628, 2012.

[3] M. Aiash and J. Loo. A formally verified access control mechanism for information centric networks. In *Proceedings of the 12th International Conference on Security and Cryptography*, pages 377–383, 2015.

[4] CCNx manifest structure, 2015. https://www.ietf.org/mail-archive/web/icnrg/current/msg01182.html.

[5] R. S. da Silva and S. Zorzo. An access control mechanism to ensure privacy in named data networking using attribute-based encryption with immediate revocation of privileges. In *12th Annual IEEE Consumer Communications and Networking Conference (CCNC)*, pages 128–133, 2015.

[6] N. Fotiou, G. Marias, and G. Polyzos. Access control enforcement delegation for information-centric networking architectures. In *Proceedings of the second edition of the ICN workshop on Information-centric networking*, pages 85–90. ACM, 2012.

[7] K. Katsaros, L. Saino, I. Psaras, and G. Pavlou. On information exposure through named content. In *10th IEEE International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness (QShine)*, pages 152–157, 2014.

[8] Spotlight: Encrypted internet traffic, 2016. https://www.sandvine.com/trends/global-internet-phenomena/.

[9] F. Hao V. Hilt Z. Zhang M. Varvello V. K. Adhikari, Y. Guo and M. Steiner. Measurement study of Netflix, hulu, and a tale of three CDNs. *IEEE/ACM Transactions on Networking*, 23(6):1984–1997, 2015.