Flexible Multimedia Content Retrieval Using InfoNames

Arun Kumar, Ashok Anand, Aditya Akella
University of Wisconsin-Madison
{arun, ashok, akella}@cs.wisc.edu

Athula Balachandran, Vyas Sekar, Srinivasan Seshan
Carnegie Mellon University
{abalacha, vyass, srini}@cs.cmu.edu

ABSTRACT

Multimedia content is a dominant fraction of Internet usage today. At the same time, there is significant heterogeneity in video presentation modes and operating conditions of Internet-enabled devices that access such content. Users are often interested in the content, rather than the specific sources or the formats. The host-centric format of the current Internet does not support these requirements naturally. Neither do the recent data-centric naming proposals, since they rely on naming content based on raw byte-level hashing schemes. We argue that to meet these requirements, enabling content retrieval mechanisms to *name* and *query* directly for the underlying *information* is a good way forward. In addition to decoupling content from available sources and transfer protocols, these "information-aware names" or *InfoNames* explicitly decouple the information from content presentation factors as well. We envision an InfoName Resolution System (IRS) to resolve location based on InfoNames, while taking into account the operating conditions of devices. In this demo, we present an application to show how InfoNames can serve as presentation-invariant and portable names to fetch video content independent of device capabilities and resource constraints.

Categories and Subject Descriptors

C.2 [**Computer Communication Networks**]: Miscellaneous; H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Design, Algorithms

1. INTRODUCTION

Multimedia content constitutes a dominant and increasing fraction of Internet usage. Their use today is characterized by two key trends. First, any particular media item is available from multiple hosting locations, and in many formats and resolutions. Second, users' devices are capable of playing multiple media formats and resolutions, and have different operating conditions (bandwidth, power, etc.). From a user perspective, we are often interested in the content itself, rather than the source or formats, and furthermore, in dynamically adapting these aspects to match our operating constraints. The host-centric model of the Internet does not support these requirements naturally. Data-centric architectures [8, 7, 9] decouple content from location, but do not provide the capability to flexibly select from alternate presentations, since the differences in format, resolution, etc. lead to different names based on byte-level hashing. Using keywords from content metadata (e.g., name of movie) to bind content (e.g., via Google) may not work that well either, since metadata is usually non-unique and incomplete for many media files.

Our position is that such flexible multimedia retrieval requires mechanisms that allow users to directly *name*, and *query* for, the underlying *information*. Here, we use the term information to represent the perceptual entity that captures the content's defining or most significant features. Using information-aware naming and querying schemes, users can also retrieve different portions of media from multiple sources, in different presentation formats if needed, thereby increasing both *availability* and *flexibility*.

While these ideas are appealing, they pose key design and implementation challenges. First, generating such information aware names or *InfoNames* is more complex than traditional hash-based naming (e.g., [5]). The key challenge lies in ensuring that InfoNames are *presentation invariant* and *bound* to the information. InfoNames must be *unique* across dissimilar content and *compact* relative to content size. We leverage ideas from the multimedia literature (e.g., [4, 6, 1, 2]) to design InfoNames with these properties. We envision that the InfoNames can be tied to human readable identifiers (e.g., keywords) through out-of-band mechanisms such as search engines, social networks etc., similar to mechanisms for binding "flat" names to human understandable identifiers [7].

Second, we need an appropriate *InfoName Resolution System* or IRS (analogous to today's DNS). Content providers can register their InfoNames, and consumers can lookup and retrieve content based on InfoNames. The IRS must also support suitable APIs for content providers and consumers to specify their capabilities in the query. E.g., flash is unsupported on an iPhone, and lower resolution is desired in battery or bandwidth-constrained settings.

In this demo, we present a proof-of-concept design and implementation of mechanisms that enable such flexible retrieval of multimedia content. We present a suitable API for the IRS, extending the conventional *find()/register()* API [7] to use *InfoDescriptors*, in addition to InfoNames. These InfoDescriptors carry meta-data to qualify key attributes of the content (e.g., format, resolution) and the InfoName (e.g., the algorithm used). Information providers and consumers specify their preferences or capabilities using InfoDescriptors. Table 1 describes the API for Information Resolution System.

We demonstrate one application using this framework, which shows how InfoNames serve as presentation-invariant and portable names to fetch video content independent of device capabilities (e.g., ability to play particular format of video) and resource constraints (e.g., bandwidth). Users can "bookmark" the InfoNames to refer to some human understandable context (e.g., a movie name)

SIGCOMM'10, August 30–September 3, 2010, New Delhi, India.
ACM 978-1-4503-0201-2/10/08.

Command	Description
Register(URI, InfoName, InfoDescriptor)	Registers the video at the given URI to contain the specified InfoName and InfoDescriptors
DeRegister(URI)	Unregisters the video at the specified URI
GetInfoName(URI)	Given the URI to the video, it returns the globally accessible InfoName corresponding to the video
Find(InfoName, Constraints)	Given the InfoName for a particular video and the constraints, it selects the best fit video and returns its corresponding URI

Table 1: The API of the InfoName Resolution System.

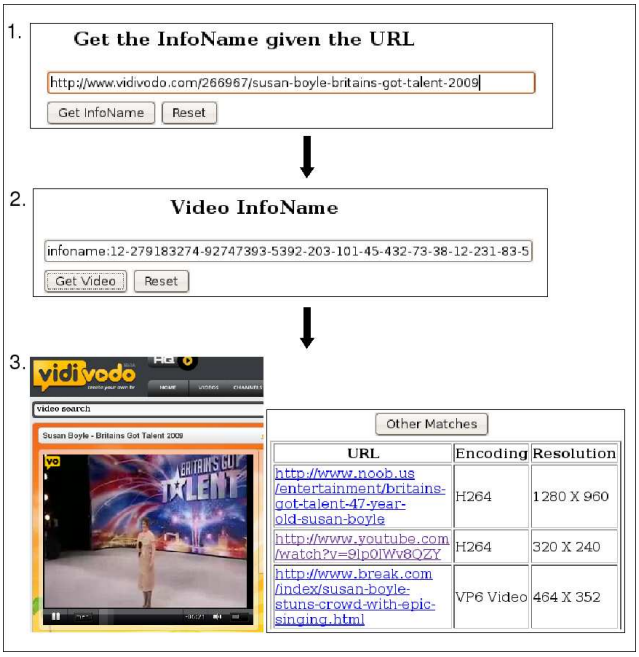

Figure 1: Snapshots of the different steps in the InfoName-aware video streaming application: 1. The URL is used to get the InfoName of that video. 2. The InfoName is obtained. Note that it is not meant to be human-readable, but rather serves as a portable and persistent binding to the information. It is used along with implicit metadata constraints to retrieve best matching video URLs. 3. The best matched result is played. Optionally, other matched results can be viewed.

and share these persistent identifiers with other users. This enables *late binding* to particular content sources and presentation formats, thereby allowing users and applications to best adapt to their capabilities and operating constraints.

2. DEMONSTRATION

We demonstrate the application of portable InfoName identifiers as follows. For the purpose of this demo, we have implemented the underlying resolution system as a centralized store-and-lookup server that maintains a database of InfoNames and their mappings with the InfoDescriptors. We use a relational schema for InfoNames and use MySQL to insert/retrieve them.

Obtaining InfoNames:

The user obtains the InfoName given a URI. This involves invoking the *GetInfoName()* call which returns the InfoName for the video URI. For the demo, the IRS is pre-populated with InfoNames and InfoDescriptors. We do so only for the purposes of the demonstration. In practice, the IRS can obtain InfoNames and their mappings on-the-fly from different content sources (e.g., transcoding proxies or InfoName-aware caches) or alternatively obtain this data by crawling several potential sources. This is the first step in Figure 1.

Identifying potential sources:

When a user on a laptop clicks on *Get Video* for the InfoName, the laptop constraints (e.g., capability to play some set of formats, preferred resolutions based on available network bandwidth etc.) are passed along with the InfoName in *Find()*. The IRS looks up the set of video URIs that have the same (or similar) InfoName and returns a ranked list of URIs matching the users' criteria. The user then gets the corresponding video and it is played on laptop. We show how this results in getting a different video format for another laptop with different capabilities. This is the second step in Figure 1 which shows the best fit video, after the user clicks on *Get Video*. The 3rd step shows that by clicking *Other Matches*, we find URIs of the other results that also match the same InfoName.

Dynamic adaptation:

We vary the bandwidth between video server laptop and client laptop, and show how different resolutions of videos are selected accordingly. Today's streaming solutions (e.g., Microsoft SmoothHD and Adobe Zeri) provide capabilities for bandwidth-aware adaptive video streaming. However, Infonames generalizes this to all media, and not only to those where such systems are deployed.

3. REFERENCES

[1] The LibFooID Audio Fingerprinting Library. `http://code.google.com/p/libfooid`.

[2] The Open Source Perceptual Hash Library. `http://phash.org`.

[3] J. S. Boreczky and L. A. Rowe. Comparison of Video Shot Boundary Detection Techniques. In *Proc. SPIE, Storage and Retrieval for Media Databases*, pages 170–179, 1996.

[4] B. Coskun and B. Sankur. Robust video hash extraction. In *In Proceedings of the European Signal Processing Conference, EUSIPCO*, 2004.

[5] M. Rabin. Fingerprinting by Random Polynomials. Technical report, Harvard University, 1981. Technical Report, TR-15-81.

[6] S.-C. Cheung and A. Zakhor. Estimation of Web Video Multiplicity. In *Proc. SPIE, Internet Imaging*, 2000.

[7] T. Koponen et al. A Data-Oriented (and Beyond) Network Architecture. In *Proc. SIGCOMM*, 2007.

[8] N. Tolia, M. Kaminsky, D. G. Andersen, and S. Patil. An Architecture for Internet Data Transfer. In *Proc. NSDI*, 2006.

[9] V. Jacobson, D. Smetters, J. Thornton, M. Plass, N. Briggs, and R. Braynard. Networking Named Content. In *Proc. CoNEXT*, 2009.