Similarity Content Search in Content Centric Networks

Petros Daras[b], Theodoros Semertzidis[a,b], Lambros Makris[b] and Michael G. Strintzis[a,b]

[a]Information Processing Laboratory
Department of Electrical and Computer Engineering
Aristotle University of Thessaloniki
GR54124, Thessaloniki, Greece
+302310996396

[b]Informatics and Telematics Institute
Centre for Research and Technology Hellas
GR57001, Thessaloniki, Greece
+302310464160

daras@iti.gr, theosem@iti.gr, lmak@iti.gr, strintzi@eng.auth.gr

ABSTRACT

Content searching and downloading are the two dominant actions of the Internet users today, despite the fact that the Internet was not originally architected to serve such actions. Content Centric Networking is the new trend in the research community to build network architectures that route content by name and not by the host's network address so as to efficiently deal with content persistence, availability and authenticity issues. In this work we propose an extension to the Content Centric Network protocol in order to support content search as a native process of the network. By using object descriptors to represent the actual content objects and integrate these descriptors in the network protocol we manage to search and retrieve content from the network not only by name but also by the content itself. In this approach, searching for information is a process distributed to the reachable network. Moreover, content aggregation is handled by the end user and not by content aggregation portals, thematic content search engines or information curators. By doing so, search is not any more an application but part of the network and can pave the way for many novel applications which have never been thought until today.

Categories and Subject Descriptors

H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval – *Query Formulation, Search Process, Selection Process.*

General Terms

Algorithms, Design, Experimentation.

Keywords

Content Centric Networking, search, similarity, descriptors.

1. INTRODUCTION

The evolution of the Internet as we know it changed the way users perceive and exploit its content, since both content production and consumption take place in an alternate fashion. Although the number of interconnected users and the amount of information increase exponentially, searching still relies only on commercial search engines, aggregators, information portals using basically keyword-based searching.

MM'10, October 25–29, 2010, Firenze, Italy.

In the quest for solutions that answer the needs of today's and future Internet, the research community proposes several network architectures which focus on the actual content and not on the hosts that store the content in question. In these new approaches routing of content is achieved by its name and not by the host's network address [1][2][3]. With these approaches, networking solves problems such as the persistence, availability and authenticity of the content [2]. Despite the fact that these architectures try and solve problems regarding the ability of the network to withstand excessive use, they do not answer other issues that emerged early in the evolution of the Internet, such as content-based search and retrieval.

Internet search engines, such as Google and Yahoo!, made giant steps towards serving content to users and thus they became equal to the Internet in the users' minds. The queries sent to the Google search engine raised from 2.7 billion in 2006 to 31 billion searches in 2007, a ten-fold increase. For this year alone, it is estimated that hundreds of Exabytes of unique information will be newly generated. However, the searching for content has still several open issues to be a solid effective procedure. Some of these issues are the centralized indexing of the search engines, the inability of the search engines to access all user-generated content, the dependence of the users to the commercial advertisement driven, search engines, etc. Thus, it is clear that a plethora of information is lost due to obsolete search services and methods. One of the most problematic areas of the non-corporate search engines is the fact that they cannot track data nearby the user such as inside a LAN or in an intranet environment.

A typical search engine extracts (in an offline process) feature vectors (descriptors) from the content (which are used as indices of the content) as well as from the query object (online process), by using a specific method such as the CMVD [5] descriptors for 3D models, CEDD [6] for image retrieval, etc. When a new content-based search is performed, the descriptors extracted from the query object are compared to the descriptors of the search engine's index and a similarity measure is computed. The result returned from the search engine is a list of objects (e.g. links, images, 3D models, audio files, etc.) ranked from the most similar to the least similar ones. Figure 1 depicts a typical procedure which a multimedia search engine follows in order to answer to a specific query by object (query by example).

In the current work we propose an extension to the Content Centric Networking (CCN) protocol in order to incorporate a native similarity search property, for content-based object retrieval in the network. By the term content-based object retrieval we refer to the procedure used by the majority of the multimedia search engines [4]

to index and retrieve objects based on a query object with similar characteristics such as color, texture, shape and other similarity features in images, videos, 3D models or audio content.

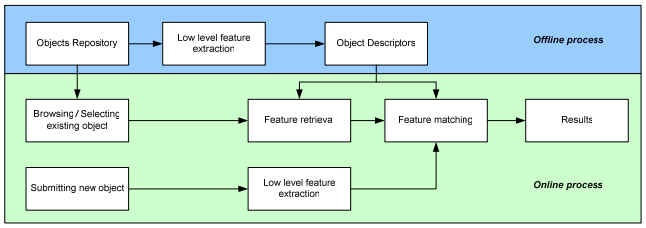


Figure 1: A typical Multimedia Search Engine diagram.

The rest of the paper is organized as follows: Section 2 provides a detailed description of the proposed CCN search protocol while Section 3 addresses the advantages and innovation of the concept. Section 4 describes a CCN Search application to demonstrate the proposed idea. Finally Section 5 summarizes the work done and concludes the paper.

2. CCN SEARCH PROTOCOL

The CCN protocol is based on two packet types, namely *Interest* and *Data* packet. The Interest packet, which is transmitted by a consumer, is propagated through the network to the nodes that have registered available content. Longest prefix matching is used, to match content names to the Interest name in order to request for data. Moreover, the Data packets can traverse the network only as a reply to an Interest packet, along the path that the Interest packet traced [1].

With this architecture, the CCN routers forward Interest packets, using longest prefix matching, to interfaces that have registered the prefix in question. This is an exact match process and in this case the user has to know the name or in some extend the prefix of the content s/he wants to retrieve.

What we propose in this work is a protocol for searching the network by flooding it with a query requesting similar objects. Since the interest packets are the ones that drive the communication in the CCN protocol, the search must be initiated by an interest packet. However, the name of this interest packet is not a prefix of some content but a search query prefixed with a conventional name that passes through CCN routers. We, thus, introduce the "*search*" namespace as a top level namespace that CCN parties register to share *searchable* content.

The other components of the search Interest name is the globally routable domain of the search initiator that the other parties use to reply to it, a search id to discriminate this query from another one committed by the same CCN party and finally the actual feature vector of the query object. Figure 2 depicts the flooding of the search interest to the CCN network as well as the interest name with the necessary name components. The initial search interest will reach every party, with registered *searchable* content to its local CCN router, and trigger the similarity search process.

Every party that shares searchable content performs a feature extraction task on it. As a result, a feature vector is extracted for every object (image, video, 3D model, audio, text, etc) and an index is formed for every object type.

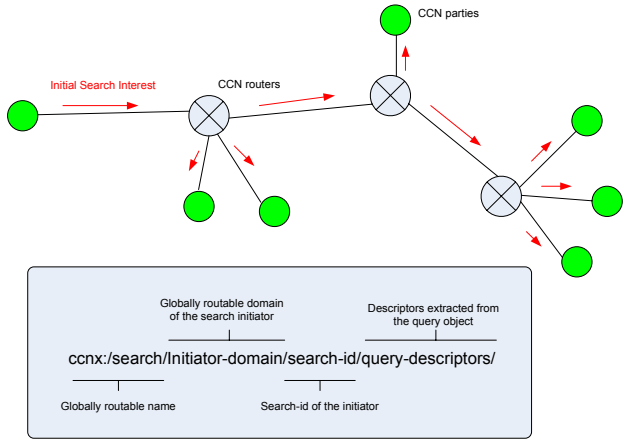


Figure 2: The initial Interest name that triggers the similarity search in CCN.

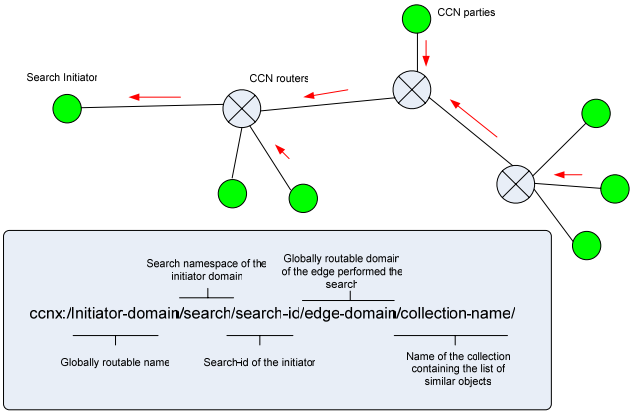


Figure 3: The CCN parties that have similar objects reply with an interest to let the initiator know their availability.

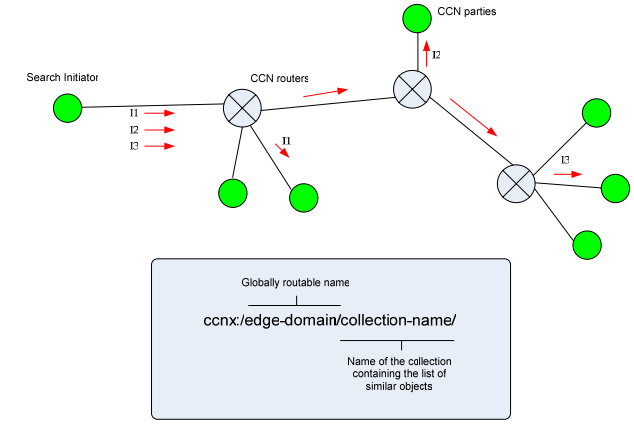


Figure 4: The initiator transmits an interest for each collection name containing the lists of similar objects.

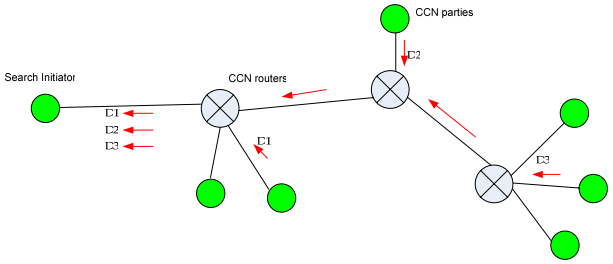


Figure 5: Data packets carrying the collections of names of similar objects return to the search initiator.

When a search query is received, the CCN party performs similarity search by comparing the query descriptors against the index (using a typical L-distance) to find the set of most similar objects passing a threshold of percentage of similarity or another way of reducing the results set.

In the case that a CCN party has similar objects, a series of actions have to take place. Firstly, the CCN party compiles a collection containing "CCN links" of names of the similar objects. The CCN links work like the HTTP hyperlinks by providing a label name and a target name. For the link's label and target name the similar object's descriptor and CCN name are used, respectively. Secondly, a unique name is generated for this new content (the collection of links). Thirdly, a dummy data object is sent as a reply to the initial interest to consume it at the local CCN router if it still exists (it is possible that another dummy data object from another interface consumed the interest in the CCN router). Finally, an Interest is sent targeting the search initiator to inform him/her about its availability. As shown in Figure 3, the name of this replying interest starts with the name of the initiators domain. Then, it uses: the search namespace to make clear that this is not an Interest about some actual content; the search id to identify the specific search and finally two new components. The first one is the globally routable name for this CCN party and the last one is the name of the new collection object that contains the results set. These two components may be treated as one, globally routable CCN name of the newly created similarity collection.

The search initiator receives these interests and forms a new interest as a reply for each one, in order to request the actual collection objects. Figure 4 depicts the form of these new interests that now target individual parties to receive the results of every independent search. Figure 5 presents the data packets send by every party to the search initiator. At the end of this process the search initiator ends up with a set of collection objects that contain links to names of similar objects. At this stage the search initiator may continue with two different approaches:

The first one is to request each content object contained in the collections that s/he received. Due to the distributed and independent nature of the search process, it is possible that the collections of object names have some overlapping data. In this case the search initiator may request only unique data.

A second way to proceed is to collect the descriptors from all the link labels, as we mentioned above, and compute a new index based on the results of the whole reachable network. With this approach every node in the network may index the content in a custom way and provide fast search services. An alternative approach is to retrieve the results set and then apply a different method for feature vector extraction to rank the set in a different way based on the exact needs of the user.

In any case the final sorting of the resulting content objects takes place at the search initiator since this is the most appropriate one to perform this task, based on its individual needs. This is a completely different approach from the centralized indexing and ranking provided by the known search engines.

The search process may also be fine tuned by: (a) Building more sophisticated queries to help the performance and quality of similarity search at each CCN party; (b) Specifying trusted parties or preferred name prefixes; (c) Using the Scope property of the Interest packet that works similar to the TTL (Time to live) of the IP protocol. A fast search may be performed by constraining the Interest to a certain scope, whereas an extensive search, which would allow serious time delays, could have larger TTL values.

3. ADVANTAGES AND INNOVATION

The proposed work approaches the searching for content from a completely different perspective. The user of the Search enabled CCN is considered attached both in a peer-to-peer network and in a high throughput backbone network that caches a deluge of information. By using the efficient caching and forwarding mechanisms of CCN, nearby content is always available without even the need of Internet connection.

The search process is transparent to the end user since s/he does not need to know where the retrieved content is stored. This makes extremely easy for the user to search at once the local file system, the nearby computers on his/her LAN or intranet infrastructure as well as the whole Internet. This possibility combined with the security and authentication features inherited by the CCN protocol will be a breakthrough for corporate /campus searching.

By doing so, searching and indexing may be as close or as open as the users of a specific network decide to be. However, the major problem of indexing content behind NATs and Firewalls is eliminated. Moreover, users may build custom indices of the nearby content in the network or compile collections of content on the fly. Content Centric Networking provides a solid architecture for content caching that makes content aggregation extremely easy and clean process. Finally, by using a CCN to perform search, the need for dedicated servers is eliminated. This also implies less traffic in the network for the popular content and a step towards green ICT.

4. CCN SEARCH IMPLEMENTATION

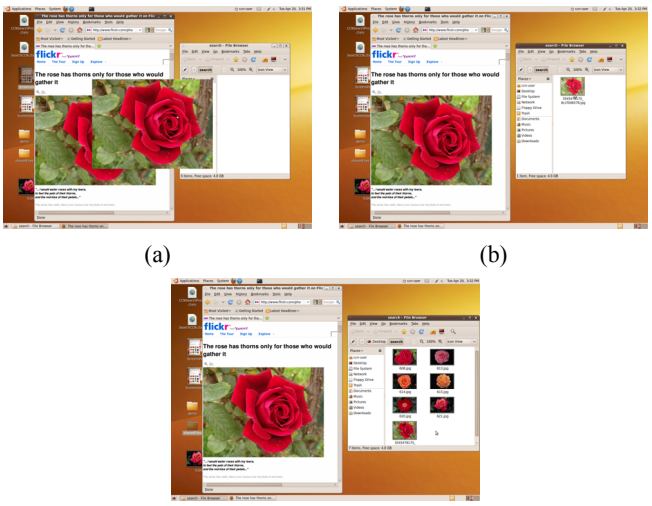
A search application was implemented to demonstrate the applicability of the proposed logic to the CCN protocol.

One of the main concerns during the design of the application was the seamless integration of the search process in the user's desktop. As we stated later, the similarity searching has to be as transparent to the end user as possible. The users need to search in the network in the same way they search in their local file system. Applications like Google desktop provide such functionality but in a very primitive way, based on keywords.

In our case the search is triggered by dragging and dropping a content object in a folder marked as "search folder". The CCN search process is starting behind the scenes and after some seconds the folder is populated with similar objects found in the Content Centric Network both in the local and the remote computers. Figure 6 (a) and (b) show the dragging and dropping actions of an image

from a web browser to the "search folder" whereas in (c) the "search folder" is filled with similar objects returned from the searching of the network.

For the implementation of the CCN search application we used Java running on an Ubuntu 9.10 machine. For the CCN network we used the CCNx [7] open source library that provides a primary implementation of the CCN protocol both in java and C as well as some helper and test applications. The core application of the CCNx project is the CCND, which is a user space daemon for linux based machines. CCND performs the entire packet forwarding procedures of the CCN protocol and interconnects hosts using the UDP protocol. For the use case scenario tested in this work, 3 VirtualBox [9] installations with Ubuntu 9.10 guest OS where used to form a CCN network.



(a) (b)

(c)

Figure 6: The user drags the image from the web browser (a) and drops it to the search folder (b) to trigger the search in CCN. Finally the search folder is populated with the results of the search process (c). The search procedure is transparent to the end user.

The monitoring of the "search folder" is achieved by using the inotify-tools project [10], which are both a C library and a set of command line tools. More specifically the "inotify-wait" command provides an interface to monitor and act upon filesystem events. For the feature extraction on Images we used the Lire [8] Java library that contains an implementation of the CEDD [6] descriptors. A vector of 144 features was extracted for every *searchable* object in the network in an offline procedure. For the query image the same descriptors extracted in real-time triggered by the filesystem event.

Each CCN party in the network that gets the search query performs a matching procedure to find similar objects in its repository. In the case that similar objects exist a collection is compiled, as we mentioned in Section 2. Finally the "search folder" of the search initiator is populated with the similar content.

5. CONCLUSIONS AND FUTURE WORK

In this paper we presented a novel approach for integrating content-based similarity search in Content Centric Networks. With this approach content-based searching becomes a transparent operation to the end user. Having the ability to search and retrieve content not only using names but also similar content will improve the user experience and promote new ways of interacting with the network and the content. In this way content is easily traceable and the end user gains the ability to select, sort, aggregate content in a custom way that addresses his/her specific needs. For the future, we plan to test this concept in a larger network and evaluate in detail its scalability. The sources of the CCN Search implementation will be made available as open source software in the near future.

6. ACKNOWLEDGMENTS

This work was supported by the EU FP7 projects I-SEARCH, ICT-248296 and 3DLife, ICT-247688.

7. REFERENCES

[1] V. Jacobson, D. K. Smetters, J. D. Thornton, M. Plass, N. Briggs, and R. Braynard. Networking Named Content. In CoNext, 2009.

[2] Teemu Koponen, Mohit Chawla, Byung-Gon Chun, Andrey Ermolinskiy, Kye Hyun Kim, Scott Shenker, and Ion Stoica. A Data-Oriented (and Beyond) Network Architecture. Proceedings of ACM SIGCOMM 2007, August 2007.

[3] Matthew Caesar, Tyson Condie, Jayanthkumar Kannan, Karthik Lakshminarayanan, Ion Stoica, and Scott Shenker. ROFL: Routingon Flat Labels. In SIGCOMM, 2006.

[4] Nozha Boujemaa, Ramon Compano, Christoph Doch, Joost Geurts, Yiannis Kampatsiaris, Jussi Karlgren, Paul King, Joachim Koehler, Jean-Yves Le Moine, Robert Ortgies, Jean-Charles Point, Boris Rotenberg, Asa Rudstrom, Nicu Sebe. "State of the Art on Multimedia Search Engines".IST CHORUS coordination action. Report D2.1, 2007.

[5] Petros Daras, Apostolos Axenopoulos, "A Compact Multi-view Descriptor for 3D Object Retrieval," cbmi, pp.115-119, 2009 Seventh International Workshop on Content-Based Multimedia Indexing, 2009

[6] S. A. Chatzichristofis and Y. S. Boutalis, "CEDD: Color and Edge Directivity Descriptor - A Compact Descriptor for Image Indexing and Retrieval.", « 6th International Conference in advanced research on Computer Vision Systems ICVS 2008» Proceedings: Lecture Notes in Computer Science (LNCS) pp.312-322, May 12 to May 15, 2008, Santorini, Greece.

[7] Project CCNx. http://www.ccnx.org/ (April 2010)

[8] Lux, M. and Chatzichristofis, S. A. 2008. Lire: lucene image retrieval: an extensible java CBIR library. In Proceeding of the 16th ACM international Conference on Multimedia (Vancouver, British Columbia, Canada, 2008)

[9] Oracle VirtualBox. http://www.virtualbox.org/ (April 2010)

[10] Inotify-tools Project. http://wiki.github.com/rvoicilas/inotify-tools/ (April 2010)