# A Keyword-based Similarity Content Search in Named Data Networking

Minbin Luo

State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, 100876, China

## Email: luominbin@bupt.edu.cn

Keywords: NDN; Search; Content Retrieval; Content-centric Networking; Future Network;

**Abstract.** Named Data Networking is a future Internet architecture which enables content retrieve and access by its name. Different from the current network, Named Data Networking is more like a distributed network. Nevertheless, users do not know name of the content. In this paper, we present a Keyword-based NDN protocol which is an extension to the current Named Data Networking instead of centralized search mechanism. By using inverted tables as an index for content both in cache and repository, we manage to act the same like modern search engine, but completely different inside and faster.

# Introduction

Internet plays an increasingly important role in our society and it has changed our way we retrieve information. Meanwhile, along with a rapid speed of consuming multimedia contents increase, there is an exponential increase in amounts of data in network which brings a big challenge to today's host-to-host network. Given the circumstances, Named Data Networking project and new approaches was proposed to solve the issue of effective delivery data content [1][2][3].

NDN is a consumer-driven network. There are two types of packets: Interest and Data. Consumers broadcast interest packets and every other party who have received interest packets use the longest prefix matching to check data in cache and repository and respond. If there is no data in both, the party will match PIT (Pending Interest Table) and add a new face to it. If there is no match in PIT, it would check FIB(Forwarding Information Base) and propagate it. Otherwise this interest would be discarded.

Due to rapid growing of data, Internet search engines like Bing and Google still have some issues to be solved. Current search engines use web crawlers to pick up pages and it takes a great deal of time. It is lack of synchronicity of information retrieval because resource collection is separated from user queries. New information can't be updated immediately which lead to die chain and errors. Massive data in network also can result in inefficiency.

In this paper we proposed a scheme that is an extension to Named Data Networking(NDN) protocol to support a keyword-based search property for named data. Inverted table [4] is used as an index for keyword search. NDNBrowser [5] is a browser implemented by webkit and support NDN network. It gives users a traditional way for users to retrieve specific data. The whole network would return a collection of URLs for users to select.

### Search mechanism design

The basic idea is that every party in the network which has relevant and available resources can return a collection of URLs for search initiator to manager. In Named Data Networking, packets always associated with a name. Our search start at a single interest packet named as figure 1. Global ID indicates an organization or a company. App ID is the name of the application which starts the query. Propagation component can be broadcast and unicast and Action can be request and response. The first interest is a special interest that is different from other interests and it has a timeout field to end propagation. Every party that run NDN protocol would trigger a search process.



There is an inverted table as index for both cache and repository in NDN protocol. Figure 2 describes how inverted table interacts with permanent storage named Repository and temporary storage named Content Store(CS). Inverted table always synchronizes with Repository and CS. Once the query interest comes, it goes for inverted table to find matched name. While a collection of relevant URLs is formed, data packet loaded with URLs will be inserted into Repository waiting for take. The NDN party would send a notification interest to the initiator. Using interest as a notification is mentioned [6]. The name in response interest is like<sup>.</sup> /bupt.edu.cn/C2Searcher/unicast/response/keyword/C1 name/Collection Name. The interest uses unicast to propagate to the initiator. The initiate node receives interest and sends a normal interest to request for the collection of URLs. However the node without relevant URLs does not reply to the initiator and every party broadcasts the interest to other entrances.



Fig.2. Search processing Architecture.

In Named Data Networking, data transfer only by broadcast. Route is establishing after the search request interest is spreading and entries have been inserted into FIB. The path of Broadcast is like a tree meanwhile every node in the tree has only one path to the root and that is the unicast. A Freshness time is set in FIB entry which determines the maximum time it can be considered valid because it doesn't need persistence for unicast. Figure 3 describe the whole search process between search initiator and the responser.

At the final stage, the initiate node ends up with a collection of content packets that contains a set of URLs. The initiator can use different strategies to the result. For example, obviously different responses may contain duplicated URLs and we can take advantages of it. Because the search interest broadcasts, the nearest party may return fastest response. When the request ends, the initiator will get the most nearest nodes' responses, thus we can know the hottest relevant URLs by counting the number of different URLs. The application in the initiator can remove duplicate URL and sort them by frequency of occurrence to raise the hot content to the top.



## **NDN Search Implementation**

A search extension and an application are realized to prove the feasibility of the design to the NDN protocol. A search client was seamless integrated with a NDNBrowser. The search can be easily used by users in their desktop and be transparent to the users as possible. Users only need to type a keyword to trigger a search as simple as they use Google.

In our case the search is started by inputting a word into the text box and clicking search like figure 4 (a). The search process is starting in the background and a list of relevant URLs is filled under the search box after several seconds. Figure 4 (b) shows the result of the search and figure 4 (c) describe that a player pops up to play the movie resource that the user clicked.



(b) Fig.4. The user input keyword and result is returned.

The NDNBrowser is achieved based on Webkit which is implemented by C++. This page is designed by JSP and Ajax to communicate asynchronously. CCNx [7] 0.8.0 which is an open source project is used as NDN framework and it has both C and Java library. For this scenario, VirtualBox is used and 3 Ubuntu 8.04 guest OS are used to establish a NDN network.

## **Conclusions And Future Work**

We proposed and implement a distributed and novel search mechanism. Take advantages of NDN network, the user can do resource searching even if they are offline and they also can offer others content if they want. Search is embodied seamlessly into the network and transparent to users. Information now is closer to people and the traffic in network will reduce. This new search mechanism will not let users to adapt to it and gives users other ways to select valuable content from all the results. In the future, we would like to test the search performance in a large national wide scale and use machine learning and nature language process to optimize the whole process to give users a better experience.

### Acknowledgement

In this paper, the research was sponsored by Service Identification and Migration mechanism of Future Internet (No. 2012CB315802), National Key Basic Research Program of China (973 Program) and Research & Development and Experimental Verification of a Novel Web Application Platform on CCN (No. BY2013095-4-01), Prospective Research on Future Networks of Jiangsu Future Networks Innovation Institute. Thanks to State Key Laboratory of Networking and Switching Technology for the chance to take part in the project.

### References

[1] Zhang L, Estrin D, Burke J, et al. Named data networking (ndn) project[J]. Relatório Técnico NDN-0001, Xerox Palo Alto Research Center-PARC, 2010.

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

[3] 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.

[4] Zobel J, Moffat A. Inverted files for text search engines[J]. Acm Computing Surveys, 2006, 38(4):2006.

[5] Qiao X, Nan G, Peng Y, et al. NDNBrowser: An extended web browser for named data networking[J]. Journal of Network and Computer Applications, 2015, 50: 134-147.

[6] Zhu Z, Wang S, Yang X, et al. ACT: audio conference tool over named data networking[C]//Proceedings of the ACM SIGCOMM workshop on Information-centric networking. ACM, 2011: 68-73.

[7] Saadallah B, Lahmadi A, Festor O. CCNx for Contiki: implementation details[J]. 2012.

[8] Oracle VirtualBox. http://www.virtualbox.org/ (October 2015)