Caching Simulators for Content Centric Networking

Deepali D. Ahir[1], Sagar B. Shinde[2]

[1]Pune University, M .E. S College of Engineering, Wadia Campus, 19, Bund Garden,
V.K. Joag Road, Pune, India

[2]Pune University, M .E. S College of Engineering
Wadia Campus, 19, Bund Garden, V.K. Joag Road, Pune, India

Abstract: *Content sharing is becoming popular application of internet, yet internet design has little consideration for content sharing, so to keep up with latest trend and to efficiently use internet for content sharing, a new network concept is proposed called Content centric networking (CCN). It is one of the capable frameworks which support caching and improve fast content delivery. CCN is designed in such a way that each router in a network has storage space for caching. This paper discusses different tools capable for evaluating performance of caching techniques, different routing strategies used by the simulator for monitoring different distinctive metrics, like communication overhead, file download time, hit ratio, hit distance.*

Keywords: Content Centric Networking (CCN), Routing strategies, Zipf's law, Caching, Information centric networking (ICN), CCNx.

1. Introduction

Rapid developments in internet technology have totally changed the internet use from host-to-host communication system to a content distribution podium. Users are now capable of creating their own content, easily and fast, and likely to share their content with other people, as an extension of the human need for communication and experience sharing.

Today, various contents are usually hosted by media servers and web portals, and the only way to retrieve content is to establish an end-to-end connection with them. While the communication pattern in the Internet has been evolving since its early days, from client-server model to peer-to-peer networking, and to cloud networking, one significant point has been shown, i.e., the usage pattern of the Internet has become largely content-oriented. That is, content consumers do not care where and how to obtain a piece of content.

Yet the internet was designed as a *communications* network, not a *media distribution* network. These limitations impact every part of the ecosystem – from carriers to publishers, across wired and wireless communications – and operators need economical ways to solve these problems beyond marginal improvements on existing solutions and tools.

CCN is an alternative approach to the networking architecture based on the principle that a communication network should allow a user to focus on the data he or she needs, rather than having to reference a specific, physical location, from where that data is to be retrieved. CCN enables content caching to reduce congestion and improve delivery speed, a simpler configuration of network devices, and security built into the network at the data level. The initiative has continued to gain momentum with an open source code release, Android implementation release, and commercial engagements with prominent industrial partners by PARC. When compared to the current TCP/IP communication model, CCN has the following different characteristics:

i) Receiver-centric communication model: Receivers pull information by sending an interest message. At most one data message is delivered in response to an interest.
ii) Hierarchical content naming scheme: CCN does not address specific hosts, but content object itself. Content is given hierarchical names, which is similar to URLs. Interest packets are forwarded by doing longest-prefix matching at forwarding decision phase.
iii) Cache and forward architecture: Every CCN devices can cache data and use them to serve future requests.

The information centric network literature ([12]) mostly adopts the Zipf's probability distribution as a reference model to represent the pattern of client's requests in several scenarios, such as Web, File Sharing, Video on Demand (VoD) and User Generated Content (UGC). The choice of the Zipf's parameter (see Section II) heavily influences the effectiveness of caching contents in intermediate routers, in terms of cache dimensioning and hit ratio. As a consequence, the evaluation of possible forwarding strategies to be used in information centric network is also influenced by the supposed popularity distribution of contents. Rest of the paper is organized as follows; Section 2 includes basic aspect of CCN and Zipf's law, Section 3 describes simulators and routing strategies used for evaluating caching performance, section 4 comes up with the conclusion.

2. Background of CCN and ZIPF'S Law

This section describes detailed architecture and the data transmission procedure in CCN. It also explains the basic principle of the Zipf's law.

2.1 CCN Architecture

The CCN uses CCNx protocol which is based on two packet types, namely Interest and Data packet (Fig. 1) [1][4]. Interest packet is analogous to "HTTP GET", which carries a name that identifies the preferred data. That is, interest packet acts as a query for content. Meanwhile, data packet is similar to "HTTP response", which is utilized to carry the actual content. Interest packet contains a unique identifier

Paper ID: 020132220

1632

(content name), a set of parameters such as the order preference (selector), and a random nonce value to prevent the packet from looping. The most significant field from these is the content name. Fig. 2 shows an example of content name. Each name prefix has a hierarchical structure, and '/' character represents delimiter between different components. The first part of the identifier provides the global routing information; the second part contains the organizational routing information; finally, the last part shows the versioning and segmentation functionality [1].
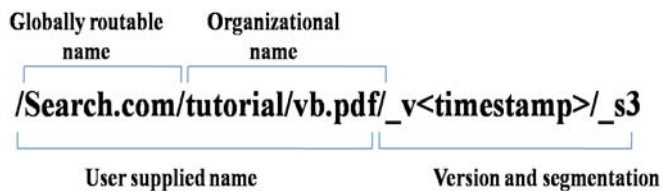

Figure 1: Format of CCN message

Figure 2: Hierarchical Content Name

CCN architecture contains the following three data structures to provide buffering/caching and loop-free forwarding.

a) Content Store (CS)
A buffer memory organized for retrieval by prefix match lookup on names. Since CCNx Content Object messages are self-identifying and self-authenticating, each one is potentially useful to many consumers. The CS needs to implement a replacement policy that maximizes the possibility of reuse such as Least Recently Used (LRU) or Least Frequently Used (LFU). The CS may retain Content Object messages indefinitely but is not required to take any special measures to preserve them; the CS is a cache, not a persistent store.

b) Face
A *face* is a generalization of the concept of *interface*: a face may be a connection to a network or directly to an application party. A face may be configured to send and receive broadcast or multicast packets on a particular network interface, or to send and receive packets using point-to-point addressing in the underlying transport, or using a tunnel (for example a TCP tunnel). A face may also be the connection to a single application process running on the same machine, via an encapsulation like UDP or an OS-specific interprocess communication path. All messages arrive through a face and are sent out through a face.

c) Forwarding Information Base (FIB)
A table of outbound faces for Interests, organized for retrieval by longest prefix match lookup on names. Each prefix entry in the FIB may point to a list of faces rather than only one.

d) Pending Interest Table (PIT)
A table of sources for unsatisfied Interests, organized for retrieval by longest prefix match lookup on names. Each entry in the PIT may point to a list of sources. Entries in the PIT must timeout rather than being held indefinitely.

2.2 ZIPF's Distribution

Several studies from past year states that Zipf's distribution is the discrete distribution that best represents the request frequency of contents [3][12] in several scenarios, such as web, file sharing, video on demand (VoD) and User Generated Content (UGC) in intermediate routers, in terms of cache dimensioning and hit ration. In order to quickly satisfy users' request for web content, ISP's utilize caching, whereby frequently used files are copied and stored "near" to users on the network. It is important to note, however, that the effectiveness of caching relies heavily on the existence of Zipf's law.

Zipf's law states that given some corpus of natural language utterances, the frequency of any word is inversely proportional to its rank in the frequency table (i.e., the smaller the rank the higher the request frequency). Thus the most frequent word will occur approximately twice as often as the second most frequent word, three times as often as the third most frequent word, etc [3]. Zipf's parameter (α) has the strong influence on the effectiveness of caching content in the network. If we denote with M the content catalog cardinality and with $1 \leq i \leq M$ the rank of i-th most popular content, the probability of requesting the content with rank i can be expressed as:

$$P(X = i) = \frac{1/i^{\alpha}}{C}$$

With $C = \sum_{j=1}^{M} 1/j^{\alpha}$, where α >0. The percentage of requests directed to the k most popular contents, i.e., $P(X = i) = k$, is representative of the impact that α parameter has in shaping content requests. If we consider a content catalog with cardinality 105, which is quite bigger than the catalog estimated for a VoD service [17], we obtain the results shown in Tab. I, which reports the percentage of the content catalog representative of the 95% of the content requests for different values of α. It can be noticed that with a small variation in α parameter, passing from 1.2 to 1.4, there is a considerable reduction in the percentage of the content catalog to which client's requests are directed. In particular, with α= 1.4, the 95% of requests are directed towards only the 700 most popular contents [2].

Table 1: Percentage of the Content Catalog Representing the 95% Of Requests

α	[%]
0.8	79.1
1	54.5
1.2	12.6
1.4	0.7

1633

3. Simulators and Routing Strategies

Different projects in the area of Named Data Networking (NDA) have been using different simulators already. CCNx [4], ndnSIM [5] and cccnSim [6] are the examples of these simulators.

3.1 CCNx

CCNx [4] is one of the emulator used in NDN projects. It is a emulation testbed, which consist of most of the operations included in the CCN proposal [19]. These operations focus heavily on security issues, as well as end node and router design.

As CCN proposes changes in the traditional OSI layer architecture of the Internet, CCNx is focusing on the interoperability of these new layers in a real simulation testbed, while it is paying less attention to transport and caching operations. Due to this reason, alternative simulation software has been developed to focus on transport and caching operations of the architecture. In particular, ndnSIM [5] builds on top of ns-3 and provides all the required hooks to evaluate mainly the transport behavior of the new networking paradigm under CCN.

- **Routing Strategy used by CCNx**

A content consumer (CC) asks for contents by sending an interest packet. That interest packet is routed towards the content publisher through the *longest prefix matching* of the name. Data are then routed the reverse path back to the CC. The CC has to ask for each segment of the content in the same way. That is, one content is fragmented to multiple segments and each piece of content is addressable. Moreover, the intermediate CCN routers can store any parts of the content data that goes through them [4]. Subsequent requests for the same content data, or requests for retransmission of some part of the content data, can then be served by the copies of the various parts of the content data that have been opportunistically stored in the intermediate nodes. Thus, every content data is delivered not only from an original content holder, but also from each CCN router to other content requesters

- **Longest Prefix Match**

Content routers use Longest Prefix Match (LPM) for Interest packet forwarding. LPM can be implemented either in hardware using a TCAM or in software using a multi-bit trie [13] or Bloom filters [14]. Solutions based on software are preferred because of TCAM's excessive cost and power consumption. LPM using Bloom filters is the most promising approach [15]. In the following, we describe how this approach works assuming LPM over content names. Assuming content names formed by B components, B Bloom filters are stored on on-chip memory, each corresponding to a unique prefix length.

Each Bloom filter is populated with prefixes having the same length. An off-chip hash-table is used to store for each prefix the next hop information, i.e., the interface where to forward an Interest packet. Upon reception of an Interest packet whose content name has B components, B Bloom filters

associated to each possible prefix length are then queried. Given Bloom filters are stored on chip all Bloom filters can be looked up in a single clock cycle. As a result, a vector of possible prefix matches then obtained, then query the hash-table starting from the longest prefix match is queried in order to verify eventual false positives, and the interface associated to the longest prefix match is retrieved.

3.2 ndnSIM

ndnSIM [2] is based on ns-3 network simulator framework. It provides all the required things to evaluate the transport behavior of networking in CCN. The simulator is implemented in a modular fashion, using separate C++ classes to model behavior of each network-layer entity in NDN: pending Interest table (PIT), forwarding information base (FIB), content store, network and application interfaces, Interest forwarding strategies, etc. This modular structure allows any component to be easily modified or replaced with no or minimal impact on other components. In addition, the simulator provides an extensive collection of interfaces and helpers to perform detailed tracing behavior of every component, as well as NDN traffic flow. In this simulator different values of Zipf's parameter (α) are use for evaluating the performance of the network.

The design of ndnSIM has the following goals:
- Being an open source package to enable the research community to run experimentations on a common simulation platform.
- Being able to faithfully simulate all the basic NDN protocol operations.
- Maintaining packet-level interoperability with CCNx implementation [2], to allow sharing of traffic measurement and packet analysis tools between CCNx and ndnSIM, as well as direct use of real CCNx traffic traces to drive ndnSIM simulation experiments.
- Being able to support large-scale simulation experiments.
- Facilitating network-layer experimentations with routing, data caching, packet forwarding, and congestion management.

Routing strategies used by ndnSIM:

- **Flooding:** In this strategy every node forwards the received Interests towards all of its interfaces except for the incoming one (i.e., the interface it received the Interest from.).
- **Best-Route with Caching:** In this strategy Dijkstra's algorithm [16] is applied to calculate the best paths to reach every permanent content copy and then the FIB of each node get sets accordingly.
- **Best-Route without Caching:** In this technique the in-network caching capacity is eliminated, so the Interests expressed by clients are satisfied only by repositories which store the seed copies. This will let us evaluate the effective benefits introduced by a distributed caching capacity, as fostered byNDN, under different values of the _ parameter.

Paper ID: 020132220

1634

3.3 ccnSim

ccnSim is a scalable chunk-level simulator of Content Centric Networks (CCN). ccnSim is written in C++ under the Omnet++ framework, and allows to assess CCN performance in scenarios with large orders of magnitude for CCN content stores (up to 10^6 chunks) and Internet catalog sizes (up to 10^8 files) on off-the-shelf hardware (i.e., a PC with a fair amount of RAM)[9]. ccnSim support simulations of large catalogue and cache sizes and it is shown to scale to simulations of millions of requests, its implementation is bound to NDN/CCN architecture.

3.3.1 Forwarding strategies use by ccnSim
The forwarding strategy receives an interest for which no PIT entry exists yet. Then, it decides on which output face the interest should be sent. We make the assumption that each node knows the location of the permanent copy of each content. There are different strategies actually implemented within ccnSim. One particular strategy can be chosen by setting the FS parameter of a node module.

- **Shortest Path Routing (SPR)**
The strategy layer chooses the shortest path repository and sends packets on the corresponding interface.

- **Random Repository**
The strategy layer chooses one repository at random out of the given set of repository. Note that this strategy requires that the core nodes follow the path chosen by the edge node (the node to which the client is attached to).

- **Nearest Replica Routing (NRR**) - **Two phases**
With this setting, the strategy layer rst explores the neighboring nodes by flooding meta-interest (i.e., interests which do not change the content of the cache) with a given time to live (TTL). Then, the strategy sends the interest packet toward the nearest nodes having the content available. Setting TTL to ∞ (i.e., greater than the network diameter) makes NRR degenerating in the ideal NRR (iNRR) strategy, which explore the entire network, looking for a copy of the given content.

- **Nearest Replica Routing (NRR 1) - One phase**
In this last case, a node which receives an interest sets up an exploration phase, in which the node floods the neighborhood with the request for the given object. When the copy is found the data comes back (it can be a permanent copy or a cached copy as well). The scope of the flooding can be set by the means of TTL2 parameter. By using all above simulators performance of content centric network is measured using the parameters like hit ratio, hit distance, file download time and overhead.

4. Conclusion

This paper describes the Content Centric Networking and the different simulators/emulators used by researchers for performance testing of CCN. Different simulators are implemented on different platforms and they are having their own pros and cons also these simulators use their own caching and routing strategies. Each of these routing and caching strategies are described in brief in this paper, which will help researchers to understand these simulators better and can also help in making a choice of simulator/emulator to use.

References

[1] J. Lee and D. Kim "Proxy-assisted Content Sharing Using Content Centric Networking (CCN) for Resource-limited Mobile Consumer Devices" IEEE Transactions on Consumer Electronics, Vol. 57, No. 2, May 2011

[2] Michele Tortelli, Luigi Alfredo Grieco and Gennaro Boggia DEI, Politecnico di Bari (Italy) "Performance Assesment of Routing Strategies in Named Data Networking" GTTI 2013 Session on Telecommunication Networks.

[3] http://en.wikipedia.org/wiki/Zipf's_law

[4] http://www.ccnx.org/

[5] A. Afanasyev, I. Moiseenko, and L. Zhang. ndnSIM: NDN simulator for NS-3. Technical Report NDN-0005.

[6] R. Chiocchetti, D. Rossi, and G. Rossini. ccnsim: an highly scalable ccn simulator. In IEEE International Conference on Communications 2013 (ICC'13)', Budapest, Hungary, jun 2013.

[7] http://icarus-sim.github.io.

[8] Alexander Afanasyev, Ilya Moiseenko, and Lixia Zhang "ndnSIM: NDN simulator for NS-3", NDN, Technical Report NDN-0005, 2012. http://named-data.net/techreports.html Revision 2: October 5, 2012

[9] http://perso.telecom-paristech.fr/~drossi/index.php?n=Software.ccnSim

[10] Diego Perino, Matteo Varvello "A Reality Check for Content Centric Networking" ICN' 11, Aug 19, 2011 Toronto, Ontario, Canada

[11] Lorenzo Saino, Ioannis Psaras and George Pavlou "Icarus: a Caching Simulator for Information Centric Networking (ICN)", SIMUTools 2014 March 17–19, Lisbon, Portugal. Copyright 2014 ICST, ISBN

[12] Lada A. Adamic1 Bernardo A. Huberman "Zipf's law and the Internet", *Glottometrics 3, 2002,143-150*

[13] Diego Perino, Matteo Varvello "A Reality Check for Content Centric Networking" *ICN'11,* August 19, 2011, Toronto, Ontario, Canada.

[14] S. Dharmapurikar, P. Krishnamurthy, and D. E. Taylor. Longest prefix matching using bloom filters. In *SIGCOMM '03*, Karlsruhe, Germany, Aug. 2003.

[15] http:// www. Named-data.net/.

[16] E. W. Dijkstra, "A note on two problems in connexion with graphs," *NUMERISCHE MATHEMATIK*, vol. 1, no. 1, pp. 269–271, 1959.

Author Profile

Deepali Ahir received M.E. degrees in computer engineering from Pune University, and currently working as Asst. Professor in M.E.S College of engineering, Pune.

Sagar Shinde received M.E. degrees in computer science and engineering from Dr. Babasaheb Ambedkar University, and currently working as Asst. Professor in M.E.S College of engineering, Pune.

Paper ID: 020132220

1635