A Model for Named Data Networking Caching Policies Inspired by Nonlinear Dynamical Systems

António Rodrigues (up200400437@fe.up.pt)

Abstract—**At the time of its inception, the Internet mostly served the purposes of communication between connected end-hosts. Now, at the World Wide Web era, the Internet is immersed in a content-centric paradigm, more concerned about content generation, sharing and access. Recently, a new research trend — Information Centric Networking (ICN) — started advocating for deep modifications on the Internet's network layer, making it content-centric by design, including the widespread use of in-network caching.**

In this paper, we focus on the analysis of cache behavior in a specific ICN architecture — Named Data Networking (NDN) — under different caching policies, network topologies and content usage characteristics. To do so, we specify a simple and but modular NDN router model, loosely inspired in nonlinear dynamical systems. We implement the specified model in MATLAB, providing some simulation results with X simple caching policies, specifically (...).

I. Introduction

Departing from its initial model as a network for host-to-host communications, the Internet started shifting towards a content-centric model with the advent of the World Wide Web in the 1980s. This model persisted, and with increasingly demanding usage requirements, leading to the development of technologies such as Content Delivery Networks (CDNs) and Peer-to-Peer (P2P) networks [1]. These were built around the architecture's edge, due to the so-called 'ossification' [2] of the Internet's core, leading to inefficiencies in terms of latency, bandwidth usage, among others. Given the widespread adoption of the content-centric model, researchers to think about new and clean-slate designs for the Internet's core, in order for it to natively cope with these issues. Among such efforts [3], the research field of Information Centric Networking (ICN) [4] emerged, advocating the deliberate abolition of network locators, replacing of IP addresses with content identifiers and calling for the widespread use of in-network caching, so that content can be easily served from multiple anywhere in the network [5]–[10]. Here we focus on the aspect of in-network caching in one of such clean slate designs, the Named Data Networking (NDN) architecture [6].

In this paper, we focus on the analysis of cache behavior in NDN networks under different caching policies, network topologies and content usage characteristics. To do so, we specify a simple and but modular NDN router model, loosely inspired in nonlinear dynamical systems [11]. We implement the specified model in MATLAB, providing some simulation results with X simple caching policies, specifically (...).

The remainder of this paper is organized as follows. In Section II we provide an overview over the NDN architecture, focusing on the basic operation of its forwarding engine and the way it involves in-network caching. In Section III, we present the overall methodology followed during this work, including an explanation of the considered NDN router models, caching policies and network topologies, while in Section IV we show details about the implementation of such models in MATLAB. In Section V we present a set of experiments ran over our model implementation, as well as the respective results. Finally, in Section VI we draw some pertinent conclusions from the presented work.

II. Named Data Networking (NDN)

In the Named Data Networking (NDN) [6] architecture, clients issue subscriptions for content objects by specifying a hierarchical (URL-like) content name, e.g. `/pdeec/mtsp/2014/`, which is directly used in NDN packets. Destination network locators (e.g. IP addresses) are not used in this case, as NDN routers are able to forward such packets towards appropriate content-holding destinations, solely based on such names. NDN contemplates two fundamental types of packets, 'Interest' and 'Data' packets, used for content subscriptions and publications, respectively. Interest packets are originally released into the network by clients willing to access a particular content, addressing it via its content name, while Data packets carry the content itself.

An NDN router is conceptually composed by three main elements: (1) a Forward Information Base (FIB), (2) a Pending Interest Table (PIT) and (3) a Content Store (CS) [6]:

- **Forward Information Base (FIB):** Routing/forwarding table holding entries which relate a name prefix and a list of router interfaces to which Interest packets matching that content name prefix should be forwarded to.
- **Pending Interest Table (PIT):** A table which keeps track of the mapping between arriving Interest packets and the interfaces these have been received from, in order to save a reverse path for Data packets towards one or more subscribers (this may be a 1:N mapping, as an Interest packet matching the same content may be received in multiple interfaces).
- **Content Store (CS):** A cache for content, indexed by content name or item. This novel element allows for content storage at the network level. In-network caching allows an Interest to be satisfied by a matching Data packet in any location other than the original producer of the content, constituting one of the main content-oriented characteristics of NDN.

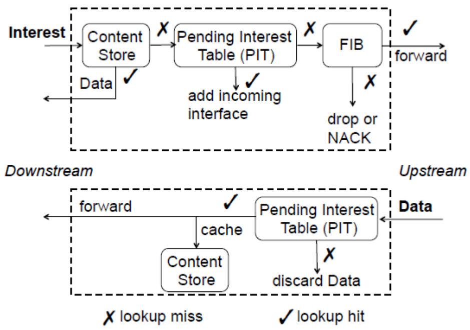

Fig. 1: Interest and Data packet processing according to NDN's forwarding engine [12].

In NDN, communication is receiver-driven, i.e. having the desire to fetch a particular content, a client releases an Interest packet into the network so that it is forwarded towards an appropriate content holder. In Figure 1 [12], we provide a graphical description of the mechanics of the forwarding engine of an NDN router, supported by the textual description provided below:

1) An Interest packet arrives on an interface (e.g. `iface0`) of an NDN router.
2) A longest prefix match on the content name specified in the Interest (e.g. `name`) is performed. The NDN router will now look in its CS, PIT and FIB, in that order, in order to resume the forwarding action:
   a) If there's a match in the router's CS, a copy of the respective CS entry will be sent back via `iface0`, the Interest packet is dropped. Depending on the pre-specified caching policy (e.g. MRU, LRU, LFU [1], etc.), the organization of the CS may change at this point. **End.**
   b) Else if there is an (exact) match in the PIT, `iface0` is added to the mapping list on the respective entry. The Interest packet is dropped (as a previous one has already been sent upstream). **End.**
   c) Else if only a matching FIB entry is found, the Interest packet is forwarded upstream, via all remaining interfaces on the list (except `iface0`), towards an eventual content holder. A PIT entry `<name, iface0>` is added. **End.**
   d) Else if there is no match at all, the Interest packet is simply discarded. **End.**

Note that in NDN only Interest packets are forwarded: intermediate NDN routers (i.e. between client and content holder) forward the Interests and have their respective PIT tables updated with Interest-to-interface mappings, pre-establishing a reverse path for Data packets to follow as soon as a content holder is found. When the reverse path is 'followed' (i.e. in the 'downstream' direction, lower part of Figure 1), each intermediate NDN router receiving a Data packet looks in its PIT for `<name, iface>` entries, and forwards the Data packet through all matching interfaces. In addition, a CS entry is created to cache the content locally at the router (again, depending on the caching policy, the organization of the CS may change at this point). If a Data packet with no matching PIT entries arrives, it is treated as unsolicited and discarded.

[1] http://en.wikipedia.org/wiki/Cache_algorithms.

III. Methodology

A. Heading 1

IV. Implementation

This section provides noteworthy details on our implementation of the methods introduced in Section III.

A. Heading 1

V. Experiments

This section describes the experiments we conducted to evaluate our and third party implementation of the Multinomial Naive Bayes models and respective semi-supervised extensions described in Section III.

A. Experimental Setup

VI. Conclusions

References

[1] Andrea Passarella. A Survey on Content-Centric Technologies for the Current Internet: CDN and P2P solutions. *Computer Communications*, 35(1):1–32, 2012.

[2] M Handley. Why the Internet Only Just Works. *BT Technology Journal*, 24(3):119–129, 2006.

[3] J Pan, S Paul, and R Jain. A Survey of the Research on Future Internet Architectures. *Communications Magazine, IEEE*, 49(7):26–36, July 2011.

[4] George Xylomenos, Christopher N. Ververidis, Vasilios a. Siris, Nikos Fotiou, Christos Tsilopoulos, Xenofon Vasilakos, Konstantinos V. Katsaros, and George C. Polyzos. A Survey of Information-Centric Networking Research. *IEEE Communications Surveys & Tutorials*, PP(99):1–26, 2013.

[5] Teemu Koponen, Mohit Chawla, Byung-Gon Chun, Andrey Ermolinskiy, Kye Hyun Kim, Scott Shenker, and Ion Stoica. A Data-Oriented (and Beyond) Network Architecture. In *ACM SIGCOMM Computer Communication Review*, volume 37, pages 181–193, 2007.

[6] Van Jacobson, Diana K Smetters, James D Thornton, Michael F Plass, Nicholas H Briggs, and Rebecca L Braynard. Networking Named Content. *Proceedings of the 5th international conference on Emerging networking experiments and technologies CoNEXT 09*, 178(1):1–12, 2009.

[7] Dirk Trossen and George Parisis. Designing and Realizing an Information-Centric Internet. *IEEE Communications Magazine*, 50(7):60–67, 2012.

[8] Dipankar Raychaudhuri, Kiran Nagaraja, and Arun Venkataramani. MobilityFirst : A Robust and Trustworthy Mobility- Centric Architecture for the Future Internet. *ACM SIGMOBILE Mobile Computing and Communications Review*, 16(3):2–13, 2012.

[9] Dongsu Han, Peter Steenkiste, Michel Machado, and David G Andersen. XIA : Efficient Support for Evolvable Internetworking. In *The 9th USENIX Symposium on Networked Systems Design and Implementation (NSDI'12)*, pages 1–14, 2012.

[10] Christian Dannewitz, Dirk Kutscher, BöRje Ohlman, Stephen Farrell, Bengt Ahlgren, and Holger Karl. Network of Information (NetInf) - An Information-Centric Networking Architecture. *Comput. Commun.*, 36(7):721–735, April 2013.

[11] J. K. Hedrick and A. Gira. *Control of Nonlinear Dynamic Systems: Theory and Applications*. 2010.

[12] Cheng Yi, Alexander Afanasyev, Lan Wang, Beichuan Zhang, and Lixia Zhang. Adaptive Forwarding in Named Data Networking. *ACM SIGCOMM Computer Communication Review*, 42(3):62–67, June 2012.