PGCOMP - Programa de Pós-Graduação em Ciência da Computação
Universidade Federal da Bahia (UFBA)
Av. Adhemar de Barros, s/n - Ondina
Salvador, BA, Brasil, 40170-110

http://pgcomp.dcc.ufba.br
pgcomp@ufba.br

Nos últimos anos, a rede centrada na informação (Information-centric networking - ICN) ganhou atenção das comunidades de pesquisa e indústria como um paradigma de rede de distribuição de conteúdo eficiente e confiável, especialmente para lidar com aplicativos centrados em conteúdo e que necessitam de alta largura de banda, juntamente com os requisitos heterogêneos de redes emergentes, como a Internet das coisas, Redes veiculares Ad-hoc e Computação na borda da rede. O armazenamento em caches na rede é uma parte essencial do design da arquitetura ICN e o desempenho da rede geral depende da eficiência da política de armazenamento de conteúdos utilizada no cache. Portanto, muitas políticas de substituição de conteúdos foram propostas para atender às necessidades de diferentes redes. A literatura apresenta extensivamente estudos sobre o desempenho das políticas de substituição em diferentes contextos. As avaliações podem apresentar diferentes variações de características de contexto, levando a diferentes impactos no desempenho das políticas ou diferentes resultados das políticas mais adequadas. Por outro lado, existe uma lacuna de pesquisas para compreender como as características do contexto influenciam o desempenho das políticas. Também faltam iniciativas que auxiliem no processo de escolha de uma política adequada a um cenário específico. Nesse sentido, esta tese aborda essas lacunas de pesquisa ao (i) apontar o que é contexto da perspectiva das políticas de substituição de cache e as características de contexto que influenciam o comportamento do cache, e (ii) propor uma estratégia de meta-política de cache para auxiliar na escolha de políticas adequadas ao contexto vigente. Para o estudo de delimitação de contexto, realizamos uma extensa pesquisa da literatura de ICN para mapear as evidências relatadas de diferentes aspectos do contexto em relação aos esquemas de substituição de cache. Além da contribuição de entender o que é contexto para políticas de cache, a pesquisa forneceu uma classificação útil de políticas com base nas dimensões de contexto usadas para determinar a relevância dos conteúdos. Além disso, como uma investigação de aspectos holísticos para representar o contexto, e motivados pela área emergente das redes centradas no humano, realizamos um estudo de caso exploratório sobre a influência do comportamento humano no desempenho das políticas. Neste sentido, realizamos um estudo baseado em simulação que avaliou o desempenho das políticas de substituição de cache por meio de clusters formados pelos usuários de acordo com seus hábitos de escuta musical. Os resultados mostram evidências de que aspectos distintos de contexto afetam o desempenho das políticas de cache. Após os estudos de contexto, apresentamos uma estratégia de meta-política capaz de aprender a política mais adequada para caches online e se adaptar dinamicamente às variações de contexto que levam à mudanças em qual política é a melhor. A meta-política se beneficia da diversidade de políticas e seus aspectos de contexto, e faz uma separação entre a lógica de remoção do conteúdo e o gerenciamento das informações de contexto usadas pela política. A estratégia modela a escolha de políticas adequadas como um problema de aprendizado online com retorno parcial. A meta-política oferece suporte à implantação de um conjunto diversificado de políticas de cache autocontidas em diferentes redes. Ela permite que o dispositivo de cache funcione como um sistema adaptativo agnóstico aos contextos subjacentes, como padrões de solicitação de conteúdo ou variações de popularidade. Os resultados experimentais mostraram a eficácia e adaptabilidade da meta-política a diferentes contextos em simulações com dados sintéticos e reais.

DSC| 26| 2022

On learning suitable caching policies in Information-centric Networks

Stéfani Silva Pires

UFBA

On learning suitable caching policies in Information-centric Networks

Stéfani Silva Pires

Tese de Doutorado

Universidade Federal da Bahia

Programa de Pós-Graduação em
Ciência da Computação

Fevereiro| 2022

Universidade Federal da Bahia
Instituto de Computação

Programa de Pós-Graduação em Ciência da Computação

ON LEARNING SUITABLE CACHING POLICIES IN INFORMATION-CENTRIC NETWORKS

Stéfani Silva Pires

TESE DE DOUTORADO

Salvador
11 de fevereiro de 2022

STÉFANI SILVA PIRES

ON LEARNING SUITABLE CACHING POLICIES IN INFORMATION-CENTRIC NETWORKS

Esta Tese de Doutorado foi apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal da Bahia, como requisito parcial para obtenção do grau de Doutor em Ciência da Computação.

Orientador: Leobino Nascimento Sampaio

Salvador
11 de fevereiro de 2022

Ficha catalográfica elaborada pela Biblioteca Universitária de Ciências e Tecnologias Prof. Omar Catunda, SIBI - UFBA.

P667 Pires, Stéfani Silva

On learning suitable caching policies in Information-centric Networks / Stéfani Silva Pires. – Salvador, 2022.

116 f.

Orientador: Prof. Dr Leobino Nascimento Sampaio

Tese (Doutorado) – Universidade Federal da Bahia. Instituto de Computação, 2022.

1. Redes - informação. 2. Cache na rede. 3. Informática. I. Sampaio, Leobino N.. II. Universidade Federal da Bahia. III. Título.

CDU 004

STÉFANI SILVA PIRES

"ON LEARNING SUITABLE CACHING POLICIES IN INFORMATION-CENTRIC NETWORKS"

Esta tese foi julgada adequada à obtenção do título de Doutor em Ciência da Computação e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação da UFBA.

Salvador, 11 de fevereiro de 2022

Prof. Dr. Leobino Nascimento Sampaio
(Orientador - PGCOMP/UFBA)

Jussara Marques de Almeida

Profa. Dra. Jussara Marques de Almeida
(UFMG)

Prof. Dr. Daniel Sadoc Menasche
(UFRJ)

Christina von Flach G. Chavez

Profa. Dra. Christina Von Flach Garcia Chavez
(UFBA)

Tatiane Nogueira Rios

Profa. Dra. Tatiane Nogueira Rios
(UFBA)

To my parents, my starting points.

ACKNOWLEDGEMENTS

Union makes us stronger and takes us further. If I got this far, it was because I had the fortune of having amazing people by my side. My first words of thanks are to my parents for their understanding and motivation that fed my soul.

To my advisor Professor Leobino, my deep gratitude for accepting this challenge and leading me with competence, maturity, and empathy. Special thanks to Professor Artur (In Memorian) for the collaboration and learning alive in our memory.

I thank Professors Christina and Fabíola and my friend Alberto for their embracement and transforming presence.

I thank my dears Adriana, Mateus, and Renato, for their surreal altruism and for each step we have taken together over these long years.

I thank the entire INSERT family, especially Italo, Allan, Rafael, and Cleber, for their fellowship and availability.

I thank my friends in Room 19, especially Augusto, Babacar, Flávia, Diego, and Matheus, for their support and optimism.

I thank my IFBA colleagues, especially Leandro, Gustavo, and Rogério, for their companionship and trust.

I am grateful to the many friends who took part in this journey, especially Isabel, Jheni, Denise, Rodrigo, Fátima, and Rafael.

There is a piece of each of you in this achievement. Thank you very much for flavoring my journey.

It is not the mountain we conquer, but ourselves.

—EDMUND HILLARY

RESUMO

Nos últimos anos, a rede centrada na informação (*Information-centric networking - ICN*) ganhou atenção das comunidades de pesquisa e indústria como um paradigma de rede de distribuição de conteúdo eficiente e confiável, especialmente para lidar com aplicativos centrados em conteúdo e que necessitam de alta largura de banda, juntamente com os requisitos heterogêneos de redes emergentes, como a Internet das coisas, Redes veiculares Ad-hoc e Computação na borda da rede. O armazenamento em caches na rede é uma parte essencial do design da arquitetura ICN e o desempenho da rede geral depende da eficiência da política de armazenamento de conteúdos utilizada no cache. Portanto, muitas políticas de substituição de conteúdos foram propostas para atender às necessidades de diferentes redes. A literatura apresenta extensivamente estudos sobre o desempenho das políticas de substituição em diferentes contextos. As avaliações podem apresentar diferentes variações de características de contexto, levando a diferentes impactos no desempenho das políticas ou diferentes resultados das políticas mais adequadas.

Por outro lado, existe uma lacuna de pesquisas para compreender como as características do contexto influenciam o desempenho das políticas. Também faltam iniciativas que auxiliem no processo de escolha de uma política adequada a um cenário específico. Nesse sentido, esta tese aborda essas lacunas de pesquisa ao (i) apontar o que é contexto da perspectiva das políticas de substituição de cache e as características de contexto que influenciam o comportamento do cache, e (ii) propor uma estratégia de meta-política de cache para auxiliar na escolha de políticas adequadas ao contexto vigente. Para o estudo de delimitação de contexto, realizamos uma extensa pesquisa da literatura de ICN para mapear as evidências relatadas de diferentes aspectos do contexto em relação aos esquemas de substituição de cache. Além da contribuição de entender o que é contexto para políticas de cache, a pesquisa forneceu uma classificação útil de políticas com base nas dimensões de contexto usadas para determinar a relevância dos conteúdos. Além disso, como uma investigação de aspectos holísticos para representar o contexto, e motivados pela área emergente das redes centradas no humano, realizamos um estudo de caso exploratório sobre a influência do comportamento humano no desempenho das políticas. Neste sentido, realizamos um estudo baseado em simulação que avaliou o desempenho das políticas de substituição de cache por meio de clusters formados pelos usuários de acordo com seus hábitos de escuta musical. Os resultados mostram evidências de que aspectos distintos de contexto afetam o desempenho das políticas de cache. Após os estudos de contexto, apresentamos uma estratégia de meta-política capaz de aprender a política mais adequada para caches online e se adaptar dinamicamente às variações de contexto que levam à mudanças em qual política é a melhor. A meta-política se beneficia da diversidade de políticas e seus aspectos de contexto, e faz uma separação entre a lógica de remoção do conteúdo e o gerenciamento das informações de contexto usadas pela política. A estratégia modela a escolha de políticas adequadas como um problema de aprendizado online com retorno parcial. A meta-política oferece suporte à implantação de um conjunto diversificado de políticas de cache autocontidas em diferentes redes. Ela permite que o dispositivo de cache funcione como um sistema adaptativo agnóstico aos contextos subjacentes, como padrões de solicitação de conteúdo ou variações de popularidade. Os resultados experimentais mostraram a eficácia e adaptabilidade da meta-política a diferentes contextos em simulações com dados sintéticos e reais.

Palavras-chave: Redes centradas na informação, Cache na rede, Política de substituição de cache, Ciência de contexto, Aprendizado online

ABSTRACT

In recent years, Information-centric networking (ICN) has gained attention from the research and industry communities as an efficient and reliable content distribution network paradigm, especially to address content-centric and bandwidth-needed applications together with the heterogeneous requirements of emergent networks, such as the Internet of Things, Vehicular Ad-hoc NETwork, and Mobile Edge Computing. In-network caching is an essential part of ICN architecture design, and the performance of the overall network relies on caching policy efficiency. Therefore, a large number of cache replacement strategies have been proposed to suit the needs of different networks. The literature extensively presents studies on the performance of the replacement schemes in different contexts. The evaluations may present different variations of context characteristics leading to different impacts on the performance of the policies or different results of most suitable policies.

Conversely, there is a lack of research efforts to understand how the context characteristics influence policy performance. There is also a lack of initiatives to assist the process of choosing a suitable policy given a specific scenario. In this direction, this thesis address those research gaps by (i) pointing out what is context from the perspective of cache replacement policies and the context characteristics that influence cache behavior, and (ii) proposing a caching meta-policy strategy to assist the choosing process of suitable policies according to the current context. For the context delimitation study, we have conducted an extensive survey of the ICN literature to map reported evidence of different aspects of context regarding the cache replacement schemes. Beyond the contribution of understanding what is context for caching policies, the survey provided a helpful classification of policies based on the context dimensions used to determine the relevance of contents. Moreover, as an investigation of holistic aspects to represent context, and motivated by the emergent area of human-centric networking, we have performed an exploratory case study on a human behavior influence over the policies performance. To accomplish such goal, we carry out a simulation-based study that evaluated the performance of cache replacement policies through clusters formed by users according to their music listening habits. The results fostered the evidence that distinct context aspects have an effect on caching policy performances. Following the context studies, we present a meta-policy strategy capable of learning the most appropriate policy for cache online and dynamically adapting to context variations that leads to changes in which policy is best. The meta-policy benefits from the diversity of policies and its context aspects, decouples the eviction strategy from managing the context information used by the policy, and models the choice of suitable policies as online learning with bandit feedback problem. The meta-policy can support the deployment of a diverse set of self-contained caching policies in different networks. It enables cache routers to work as adaptive systems agnostic to the underlying contexts, such as content request patterns or popularity variations. Experimental results in single and network of caches have shown the meta-policy effectiveness and adaptability to different contexts in synthetic and trace-driven simulations.

Keywords: Information-centric networking, In-network caching, Cache replacement policies, Context-awareness, Online learning

CONTENTS

LIST OF ACRONYMS

Chapter 1

INTRODUCTION

In recent years, the proliferation of bandwidth-needed applications and the increased capacity of modern communication devices (e.g., smartphones, network-equipped vehicles, wearables) have led to a bloom of multimedia contents consumed at the network. Due to this emerging scenario, the host-centric Internet model has experienced significant challenges in meeting the current and future users' and applications' requirements. The Internet architecture was originally designed in a host-centric paradigm to support end-to-end communication. This model struggles to face key communication requirements of modern network applications such as high content distribution, node's mobility, and network scalability.

One of the strategies to make the Internet feasible in such highly content distribution scenarios relies on networking caching approaches, which use cache-equipped devices to provide the most requested contents locally. Information-centric networking (ICN) (AHLGREN et al., 2012) is one of such initiatives. ICN is a content-centric network communication model that stand out as potential candidate to substitute the current TCP/IP model (RAHMAN et al., 2020). It consists of a receiver-driven networking model that focuses on the distribution and retrieval of contents through a publish-subscribe paradigm.

In ICNs, a content request is based on the content's name, not on its location, such as the content provider's IP address. Contents should have unique names, and any network node with the content can respond to the request. To this end, ICN replicates content in a distributed way in *Cache-enabled Routers (CRs)* over the network that are located closer to the user. Therefore, delivering the closest content copies to the user saves communication resources, thus reducing network congestion, server loads, and access latency while providing better Quality of Service (QoS) and Quality of Experience (QoE) levels. Content-Centric Networking (CCN) and its successor Named-Data Networking (NDN) (ZHANG et al., 2010) are examples of initiatives implementing ICN concepts.

In general, any network device can potentially work as a CR with a Content Store (CS) data structure to implement the cache service. The performance of CS plays a vital role in the overall packet forwarding engine to guarantee high-speed packet processing of ICN architectures. According to Pan, Huang e Li (2017) and Pan et al. (2019), the performance bottleneck of the packet forwarding systems relies on CS operation and should be the focus of ICN optimization strategies. This way, ICN-based initiatives strongly rely on cache replacement policies to manage the CS and keep relevant content available to the users. Cache replacement policies are methods used to choose which content to evict from the cache when there is the need for storing new content, and no more space is available. Examples of replacement policies include Least Recently Used (LRU), Least Frequently Used (LFU), Random, First-In-First-Out (FIFO), and Recently/Frequently Used (LRFU) (LEE et al., 2001). A replacement policy ensures that the content most expected to be accessed in a short time will remain in the cache, and the policy will, therefore, elect to evict the content that is less expected to be accessed. Different policies lead to different caching performance, and thereby the performance gain of a network of caches like ICN depends on the reliability of the cache management.

The current literature presents a massive number of performance evaluations for cache replacement policies comparing different policies concerning different network contexts. A network context refers to a network type—e.g., Edge networks, Internet of Things (IoT) networks, or Vehicular Ad-hoc NETwork (VANET)—instantiated with particular characteristics for a given purpose. A network context thus brings up a broader view that encompasses characteristics regarding the network type and other entities related to network performance (e.g., user habits while using the network). Each performance evaluation may present distinct variations in the context characteristics, as well as different impacts on policy performances, including changes in performance rank. The variance of results indicates that the policies' performance tends to vary according to the context's characteristics, and the process of choosing the suitable policies should consider the context in which the caches operate.

The attention to the caching policies is of paramount importance, especially in modern networks with the recent advances in 5th generation (5G) technology, Mobile Edge Computing (MEC), and Network Virtualization. Such technologies are revolutionizing the edge, allowing the emergence of new content-demanding applications IoT, VANETs, and new network types. The ICN model natively copes with the mobility, scalability, and security requirements of those new environments. Beyond the benefits of in-network caching, decoupling the content delivery process from the content location brings native support to mobility and multicast packet forwarding. In this way, there is an actual and rising tendency to deploy ICN-enable edge networks in 5G-ICN virtual network slices (SANTOS et al., 2021; ULLAH et al., 2020; GÜR; PORAMBAGE; LIYANAGE, 2020; NOUR et al., 2019; JEDARI et al., 2020). That scenario allows the dynamic creation/relocation of virtual cache nodes according to the demand for content consumption and customized for the context in which the cache node will be deployed.

1.1 PROBLEM AND CHALLENGES

The deployment of caches on the network leverages the content delivery process and improves network performance. However, the caching benefits will be efficiently achieved only with the deployment of caching policies suitable to the network context in which the cache operates. One relevant challenge to be addressed in managing the cache is how to choose which caching policy should be instantiated to obtain optimal caching performances. The choice of suitable policies poses a particular challenge when considering the dynamic nature of networks. The network dynamics leads to on-demand changes in the context characteristics, for instance, changes in traffic patterns or user preferences, and the cache must adapt to these changes in an attempt to ensure the best network performance.

In face of the current network diversity and dynamism, with different types of applications, heterogeneous characteristics involving mobility, and emerging technologies, it is not possible to have a single optimal caching policy capable of meeting the operation requirements off all network contexts. Therefore, the main research challenge we intend to address is **how to assist the choosing process of suitable cache replacement policies according to the current context, and further, to cope with the natural dynamism of context variations in networks.** In this direction, this thesis aims to substantiates the reasoning of the caching policy decision process during the design of caching systems in ICNs. The problem of finding best-fitting cache replacement policies exponentially grows in complexity when there is a diversity of context aspects. In the following paragraphs we elaborate on some of the challenges.

Several works incorporated the adaptation of policies according to some context. For instance, Moon et al. (2016) presented a cache management scheme for wireless NDNs, in which common Access Points (APs) and user devices attached to the APs have available cache capacity. The authors advocated that each device can choose to work with a different cache replacement policy to improve network performance. In addition to that, Charpinel et al. (2016) proposed a Software-Defined Networking (SDN) approach to provide programmable cache replacement algorithms. The replacement algorithms are defined in a control plane, allowing a CCN controller to modify the replacement schemes dynamically and allocate different strategies for each node. Finally, Pacifici e Dán (2013) proposed autonomous caching in peering ISPs for collaborative deciding their replacement policies.

Although studies recognize the need to adopt policies according to the network context, the choice itself of suitable schemes is not trivial. There is no explicit and general understanding of the relationship between the context characteristics and the policies. Such understanding is essential to assist the choosing process and, consequently, adapt policies according to the context. More specifically, there are no overall directions or categorization in which context may influence policy behavior. Yet, regardless of the isolated evidence of individual works reporting their contexts and impacts on the policies' performance, there is no comprehensive work discussing a unified view of the different contextual characteristics and their effects on the policies. The delimitation of context characteristics and their common effect can enhance and substantiate the caching management and the design of caching solutions.

Despite the contributions of previous literature reviews related to caching policies and ICN aspects (AHLGREN et al., 2012; BARI et al., 2012; ZHANG; LI; LIN, 2013; TYSON et al., 2012; XYLOMENOS et al., 2013; FANG et al., 2014; AMADEO et al., 2014;

ZHANG; LUO; ZHANG, 2015; ABDULLAHI; ARIF; HASSAN, 2015; FANG et al., 2015; IOANNOU; WEBER, 2016; AMADEO; CAMPOLO; MOLINARO, 2016; SAXENA et al., 2016; DIN et al., 2017), there is a lack of guidelines to understand context characteristics and their effect on the cache replacement policies in ICNs. Furthermore, surveys on web cache replacement policies (WANG, 1999; PODLIPNIG; BÖSZÖRMENYI, 2003; BALAMASH; KRUNZ, 2004; PANDA; PATIL; RAVEENDRAN, 2016) do not address that subject. To the best of our knowledge, there is no broad investigation on cache replacement schemes for the ICN domain or an integrated vision of the impacts of different context characteristics in the policy choice process.

Similarly, there is a lack of research efforts to assist the process of gathering which caching policy should be deployed in a given network. As a result, the lack of suitable schemes hinders the more efficient use of available cache resources, and therefore the effective extraction of the caching service expected benefits.

1.2 RESEARCH QUESTIONS AND OBJECTIVES

Based on the premise that

> *ICNs can achieve improved performance by creating customized caches concerning the context, specifically by choosing the caching policy according to the current context,*
>
> and, *there is no single optimal policy to meet the requirements of all network contexts, since the performance of the caching policies varies according to the context variations,*

this work embraces the research gaps mentioned earlier on the choosing caching policy problem by tackling the following research questions:

> **RQ1:** What are the different context characteristics that influence the performance of caching replacement strategies in ICN?
>
> **RQ2:** How to explore the cache replacement policies and instantiate best-fitting strategies dynamically adapting to on-demand changes, considering the available context characteristics of the overall scenario?

Based on such questions, the main objectives of this thesis are:

> **Research Goal 1:** Contribute to the understanding and delimitation of context-awareness from caching replacement policy perspective.
>
> **Research Goal 2:** Design and empirically evaluate a method for context-aware instantiation of suitable caching replacement policies.

1.3 METHODOLOGY

To answer our research questions and achieve the intended goals, we have carried out specific research tasks regarding each goal (Figure 1.1). To tackle the first one, we needed to understand and delimitate context from the caching policies perspective. Many definitions of context have been given in the literature as well as different methods to model and design context-aware applications (ABOWD et al., 1999; BETTINI et al., 2010; DEY, 2001; LIU; LI; HUANG, 2011; VIEIRA; TEDESCO; SALGADO, 2011; ALEGRE; AUGUSTO; CLARK, 2016; ENGELENBURG; JANSSEN; KLIEVINK, 2019). Although there is no single consensual definition, they all converge on the importance and benefits of integrating the awareness of any relevant information from relevant entities with the computational environment.

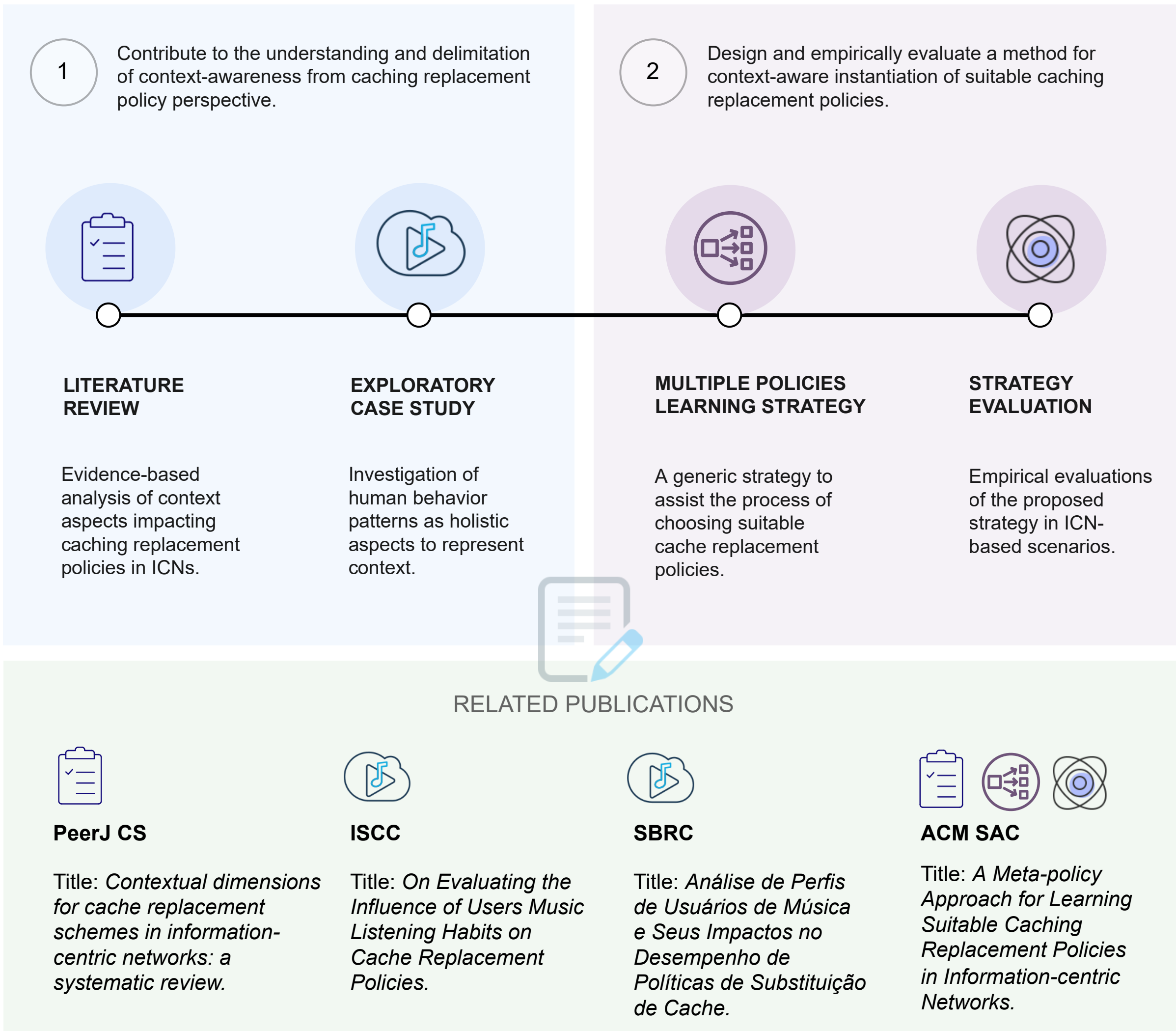


Figure 1.1 Research tasks.

This way, to consolidate our definition of context, we have performed two parallel research tasks: a literature review and an experimental practical case study. The literature review aimed to understand which context dimensions were involved in the experiments reported on the ICN literature. Also, we aimed to map the context dimensions explored by the current existing replacement policies. Furthermore, the review aimed to investigate if any explicit pattern could be explored to enhance the choosing process of policies. The review enables us to map reported evidence of different aspects of context regarding the caching replacement policies. Among the lessons learned, the study emphasized no single optimal caching policy to fit all network settings. Moreover, there is a wide diversity of context aspects related to the police's performance and the challenge to identify explicit patterns linking context variations and policies. Therefore, the challenge to identify patterns led us to conclude that the policy choosing method would not be feasible with rule-based or related systems. The literature review process and results were published as a scientific paper at PeerJ Computer Science journal (see section 1.5).

Regarding the experimental case study, we aimed to consolidate the human characteristics as an emergent context dimension related to the caching policies. Recent research fields like *people-centric networking* (CONTI et al., 2015) and *human-centric multimedia networking* (ROSÁRIO et al., 2016) are gathering attention to the basic fact that users play an essential role in demanding contents or network services, and different human characteristics can lead to different impacts on the network. In this direction, we have investigated the impact of a human habit on the performance of caching policies through simulated NDN scenarios with real data from several users of an online music stream. The study shows the benefits of using caching strategies according to users' behavior patterns on downloading songs. Among the lessons learned, there is the relevance of including human aspects in building modern context-aware caching systems, especially at network edge caches closer to the users. The case study led us to one publication at the 2018 IEEE Symposium on Computers and Communications. Later, with the evolving of the research, another publication at XXXVII Brazilian Symposium on Computer Networks and Distributed Systems.

To tackle the second research question, we started empirical studies about using machine learning techniques in network and communication systems, which led us to employ the online learning reasoning to build our proposed solution on the choosing caching policy problem. We formalize the process of context-aware instantiation of suitable policies as a continuous decision-making process. In this sense, we have proposed a new strategy that benefits from the diversity of policies and their different context aspects to build self-driven cache systems. The strategy explores multiple caching policies to learn which policy is more suitable for a given cache context. Each caching policy in the process can be associated with different context characteristics. This way, as an online continuous learning process, the cache works as a self-driven system by adapting its policy choice to context variations leading to changes on which policy becomes more suitable. We call our strategy *caching meta-policy* since it enhances the cache to learn and adapt its policy without dealing directly with the policies' eviction logic.

In the sequence, we carried out a proof-of-concept evaluation of our proposed new strategy. The evaluation relied on a simulation-based study through which the proposed approach supports caching in networking nodes of an NDN architecture. The scenarios encompassed single-cache and multi-cache networks with synthetic and real web traces. The study shows our strategy feasibility and adaptability to learn in distinct contexts. Among the lessons learned, there is the potential to embrace the overall cache context through multiple policies with different perspectives. Also, the online learning paradigm proved to be a forceful direction to meet the requirements of modern dynamic networks. Our model and its experimental evaluation were published in the technical track on Selected Areas of Wireless Communications and Networking (WCN) of the 37th ACM Symposium on Applied Computing.

1.4 CONTRIBUTIONS

We unfold the first research goal of this thesis into three main research contribution groups:

- An evidence-based mapping of context dimensions correlated with the caching replacement schemes proposed for ICNs. The mapping contributes to (i) provide a classification of contexts to assist those engaged in the design of adaptive caching solutions for ICN that target the more efficient use of available cache resources; (ii) substantiates the reasoning of the caching policy decision process by presenting and analyzing information from previous works; and (iii) enhances the set of knowledge on caching systems regarding emergent networks while underpins context-aware caching solutions.

- An exploratory investigation of human behavior patterns as holistic aspects to represent context regarding caching replacement policies. We present an analysis of the behavioral profiles of music users and how different profiles influence the performance of cache replacement policies. The study contributes to: (i) identify user profile relations with cache replacement policies based on evaluations with real datasets; (ii) ratify the inclusion of human context dimension as a context factor that influences the choice of cache replacement policies; and (ii) substantiate the building of user profiles predictor systems by presenting a correlation model between user profiles and content popularity patterns.

- Research directions useful for researchers who plan to work in this domain. We have pinpointed future research directions regarding context information management, scalability of context suitability, exploration of context information through machine learning techniques, human aspects, and privacy.

Those contributions underpinned the investigation of methods to address our second research goal. The main contributions related are:

- A generic caching meta-policy strategy to assist the process of choosing suitable cache replacement policies. The strategy is designed to learn the most appropriate policy for caching online and dynamically adapting to context variations that leads to changes in which policy is best. The strategy is generic regarding the network type in which the cache operates. Furthermore, the strategy is suitable for caches operating in different settings such as ICNs, Content Delivery Network (CDN), and Web proxy caches.

- Empirical evaluations of the proposed caching meta-policy strategy in ICN-based scenarios.

- Directions for future investigations of online learning models for collaborative caching systems. Moreover, research directions on emergent network technologies to foster the dynamic and adaptive instantiation of caching policies.

1.5 PUBLICATIONS

This thesis resulted in the following publications:

- **On Evaluating the Influence of Users Music Listening Habits on Cache Replacement Policies.**
  Pires, Stéfani S. ; Ribeiro, Adriana V. ; De Souza, Antonio M. ; Freitas, Allan E. S. ; Sampaio, Leobino N.
  In: 2018 IEEE Symposium on Computers and Communications (ISCC), 2018, Natal. 2018. p. 00930.

- **Análise de Perfis de Usuários de Música e Seus Impactos no Desempenho de Políticas de Substituição de Cache.**
  Pires, Stéfani S. ; Araújo, Francisco R. C. ; Freitas, Allan E. S. ; Sampaio, Leobino N.
  In: XXXVII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, 2019. (SBRC 2019). v. 37. p. 848-880.

- **Contextual dimensions for cache replacement schemes in information-centric networks: a systematic review.**
  Pires, Stéfani S. ; Ziviani, Artur ; Sampaio, Leobino N.
  In: PEERJ Computer Science, v. 7, p. e418, 2021.

- **A Meta-policy Approach for Learning Suitable Caching Replacement Policies in Information-centric Networks**
  Pires, Stéfani S. ; Ribeiro, Adriana V. ; Sampaio, Leobino N.
  In: 37th ACM Symposium on Applied Computing (ACM SAC'22).

1.6 DOCUMENT ORGANIZATION

The remaining chapters are structured as follows:

Chapter 2 presents introductory concepts of ICN and caching policies. In addition, the chapter explores the relation of the human-centric paradigm with caching systems.

Chapter 3 details the review of the ICN literature regarding caching replacement policies.

Chapter 4 presents the exploratory case study investigating the impact of the human context on caching policies.

Chapter 5 presents our proposed generic method for choosing caching policies.

Chapter 6 presents the experimental evaluation of the proposed choosing strategy.

Chapter 7 presents our final considerations. We pinpointed subjects out of scope of this thesis and future research directions.

Chapter 2

BACKGROUND

This chapter presents fundamental concepts related to ICN and caching policies. We pinpoint the core features of the ICN model, the discussion for real-world deployment, and the main structure of the in-network caching in NDN architecture (2.1). Next, we explore a set of application areas suitable for the execution of ICN approaches (2.2). Then, we present the types of caching policies used in ICNs, along with an example set of policies proposed in the literature (2.3). We conclude the chapter by discussing the relevance of considering human characteristics in caching solutions and presenting related user-centric research (2.4).

2.1 INFORMATION-CENTRIC NETWORKS

ICN is a new Internet architecture proposal widely discussed in the literature designed to meet the current *de facto* usage pattern of the Internet: the dissemination of content, such as videos and web pages. ICN comprises interconnected core functionalities for content naming, caching, and routing/forwarding to natively provide a content dissemination network. In its fundamental concept, the content name becomes an essential element for network routing, enabling the decoupling of content location from the content delivery process. Allied to that, ICN replicates contents in caches distributed across the network at the routers, and the closest copy will be returned when a user requests a content. Beyond the advantages of caching that provide reductions of network congestion, server loads, and access latency, the premise of independence of content location paves the way for efficient content distribution. Therefore, it adds advantages to ICN architectures, such as native support for mobility and multicast communication.

The informational RFC 8763 (RAHMAN et al., 2020) presented by the IRTF[1] Information-Centric Networking Research Group (ICNRG) discusses some approaches for the real-world deployment of ICNs and trial experiments. Besides the clean-slate approach, there are directions for its coexistence with the TCP/IP—for example, the ICN adoption as

[1] Internet Research Task Force (IRTF) - https://datatracker.ietf.org/rg/icnrg/about

an overlay network. The overlay approach proposes ICN islands deployed over existing IP infrastructure and connected using tunneling solutions. In this way, ICN packets are encapsulated inside IP packets through ICN/IP tunnels. Madureira et al. (2020) propose a resembling overlay approach with an SDN-based core network connecting edge networks operating NDN. In that case, the SDN core network encapsulates the NDN packet. Another approach is ICN as an underlay network, with the ICN islands connected to the Internet through proxies or protocol conversion gateways.

The literature presents several ICN architectures, such as Data-Oriented Network Architecture (KOPONEN et al., 2007), Content Mediator architecture for content-aware Networks (GARCÍA et al., 2011), MobilityFirst (RAYCHAUDHURI; NAGARAJA; VENKATARAMANI, 2012), and the previously mentioned NDN. They explore different architectural decisions about the naming scheme, caching, and routing processes (XYLOMENOS et al., 2013). Overall, the support for in-network caching is an essential feature of ICN design. In general, every router works with a CS structure to temporally store the contents. This way, when a router receives a content request, the router verifies whether the content is present in its own CS and immediately returns the content if stored locally. Otherwise, the router will forward the request to another destination.

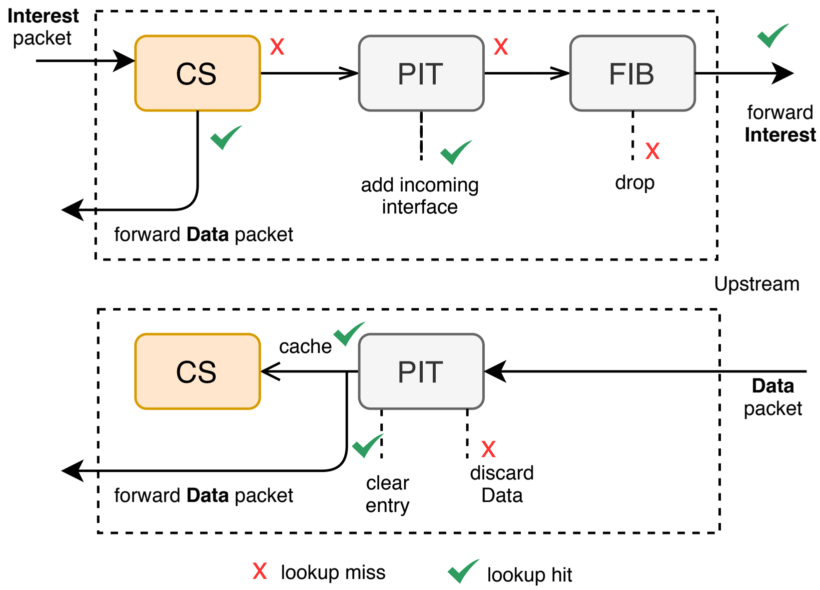


Figure 2.1 Packet forwarding engine at an NDN router (ZHANG et al., 2014).

Among the different architectures, NDN outstands as a recent and promising trend to substitute (or coexist with) the current TCP/IP model. In NDN, each CR has three main structures to support in-network caching: CS, Pending Interest Table (PIT), and Forwarding Information Base (FIB). Figure 2.1 illustrates an overview of the interaction among these structures. A content request comes in the form of an *Interest* packet to the CR, which returns a copy of the content in a *Data* packet format if the content is already present in its CS for the same incoming interface of the Interest packet. Otherwise, a new PIT entry records a pending Interest with the respective incoming interface, and the CR forwards the Interest packet according to some named-based protocol. Multiple interests for the same data are aggregated in the same PIT entry. Once the Data-packet arrives at the CR, the corresponding PIT entry is satisfied by forwarding the data to the saved interfaces. The CS will, therefore, store the passing data according to some cache management protocols.

2.2 ICN APPLICATION AREAS

The informational RFC 7476 (PENTIKOUSIS et al., 2015) presented by the IRTF-ICNRG describes a set of application areas in which ICN architectures can potentially perform better than the current host-centric Internet approach. This technical document discusses diverse network contexts in emergent areas such as social networking, real-time communication, mobile networking, vehicular networking, delay- and Disruption-Tolerant Networking (DTN), IoT, and Smart Cities. In the following subsections, we present the discussion for generic networks on information-centric IoT (ARSHAD et al., 2018; DONG; WANG, 2016), vehicular named-data networking (KHELIFI et al., 2020), and ICN-enable edge and core networks (ZHOU et al., 2017; ZHANG et al., 2018).

2.2.1 Information-centric Internet of Things

The adoption of IoT networks in many segments of society is gradually changing the way people interact with the physical world by connecting new *things* to the Internet. The imminent revolution of IoT applications must be followed by a revolution in how the network structure deals with the *content*. The current Internet architecture is fundamentally not prepared to deal with the massive amount of data from an expected number of billions of heterogeneous devices. The majority of IoT applications will be content-oriented, and TCP/IP will be struggling to meet their bandwidth requirements.

Cache-enabled solutions like information-centric architectures are strong candidates to assist in the deployment of IoT applications (ARSHAD et al., 2018; QUEVEDO; CORUJO; AGUIAR, 2014; DONG; WANG, 2016; ARAúJO; SOUSA; SAMPAIO, 2019). The ubiquitous content caching of ICN contributes to reducing the delay to retrieve content and enhances the contents' availability, especially when dealing with power restricted devices that periodically switch on and off in duty cycling to save resources. In cache-enabled network solutions, IoT traffic usually is offloaded at the Internet content routers through a connected gateway (RAO; SCHELEN; LINDGREN, 2016; MEDDEB et al., 2017) to aggregate the services of specialized IoT cloud platforms, such as Cisco IoT Cloud Connect, Microsoft Azure IoT Suite, and Google Could IoT. Also, the IoT devices can cache the traffic in a dynamically distributed IoT network (HAHM et al., 2016). Whether one case or another, two significant characteristics are a large number of heterogeneous devices and the ephemerality of the content produced by them.

2.2.2 Vehicular Named-Data Networking

The integration of vehicles in the network communication infrastructure is a trend to be incorporated into intelligent transport systems. In vehicular networking, the vehicles can exchange information with any other communication device available next to the vehicle, in a concept of Vehicle-to-everything (V2X) communication. This includes communication between vehicle and other vehicles (Vehicle-to-Vehicle (V2V)), or road infrastructure (Vehicle-to-Infrastructure (V2I)), communication network structure (Vehicle-to-Network (V2N)), pedestrians (Vehicle-to-Pedestrian (V2P)), or any other communication device.

Vehicular networking exhibit singular characteristics in traffic generation patterns, delivery requirements, and spatial and temporal scope (PENTIKOUSIS et al., 2015), mostly due to high node mobility, very intermittent connections, and the support for typical road-traffic-related applications (LI et al., 2020), infotainment applications, and code dissemination (LI; ZHAO; WONG, 2020). Thus, the content requests usually present highly temporal/spatial dependencies, and the in-network caching capabilities of ICNs can potentially improve the content delivery process. The decentralized content distribution among the vehicles allows maintaining communication in the face of intermittent connections. Also, it suits the needs of applications used in such networks, for example, to inform traffic conditions and accidents to a group of geographically close vehicles on the road.

2.2.3 ICN integration with Mobile Edge Computing

Caching at the edge in Mobile Edge Computing (MEC) (SAFAVAT; NAVEEN; RAWAT, 2019) will play an essential role in the next-generation wireless network. The Radio Access Network (RAN) is enhanced with cache capacity on Base Station (BS) structures to better attend the content demand due to its proximity. This way, Small-cell Base Station (SBS), Macro-cell Base Station (MBS), Wi-fi APs, mobile devices, and even recent cache-enabled Unmanned Aerial Vehicles (UAVs) (ZHANG et al., 2020; JI et al., 2020; HUANG et al., 2020) can store contents and respond to the content requests faster. UAVs can act as flying base stations to support the ground cellular network. They can also work as relay nodes to assist content delivery and data collection in areas without available transmission links.

The integration with ICN concepts leverages the mobile-edge caching by supporting in-network caching (ZHOU et al., 2017; PSARAS et al., 2018; SHARIAT; TIZGHADAM; LEON-GARCIA, 2016). The imminent fifth-Generation (5G) mobile networks also reinforces that merge as several initiatives discuss the benefit of the integration with ICN (ZHANG et al., 2018; GÜR; PORAMBAGE; LIYANAGE, 2020; RAVINDRAN et al., 2021).

2.2.4 ICN-enabled Core Network

ICN's benefits encompass large-scale networks with backbone core nodes and high-speed links with different capacities, interconnecting heterogeneous Autonomous Systems (AS) with multiple access networks. In this way, core networks aggregate content requests from different access networks, and unlike the edge, the temporal/spatial correlation of requests is gradually reduced and becomes weaker as the content requests approach the core nodes. Many solutions enhance ICN's applicability at core network structures for inter-domain network services such as routing (LIU et al., 2019), traffic engineering (LI et al., 2019), and globally accessible name schemes (ADRICHEM; KUIPERS, 2013).

2.3 ICN CACHING POLICIES

Cache capacity tends to be a small segment of the amount of distinct content distributed over the network. Thus, it is essential to have an efficient eviction scheme among the cache management protocols. There are different policies to tackle the management of the CS structure. They can be classified as placement and replacement policies. Placement policies, also called insertion policies, target the decision of whether a passing content should be stored locally. Examples of placement policies include Leave Copy Everywhere (LCE), Probabilistic caching (Prob), Leave Copy Down (LCD) (LAOUTARIS; SYNTILA; STAVRAKAKIS, 2004), Betweenness Centrality (Betw) (CHAI et al., 2012), ProbCache (PSARAS; CHAI; PAVLOU, 2012), and CRCache (WANG et al., 2014). On the other hand, as described in the introduction section, the replacement policies are responsible for selecting which content to evict from the cache to store new content. This work focuses on replacement policies, as we detail in the following sections.

Traditional replacement policies, such as LRU, LFU, or FIFO, are eviction strategies inherited from computer memory systems and are commonly used in ICN and web-proxy caching domains. These policies have been extensively explored to analyze cache characteristics and the performance of complex network contexts through approximation models. Orthogonally, they were not designed to fit the needs of a network of caches and do not explore its potential.

Thus, the literature presents a variety of newly proposed schemes. Jin et al. (2017) surveyed solutions for mobile caching in ICN, and among the contributions, they briefly described sets of cache insertion and replacement policies. Besides the usual LRU, LFU, FIFO, and simple Random, the list of replacement policies includes LRFU, LRU-k (O'NEIL; O'NEIL; WEIKUM, 1993), Time Aware Least Recent Used (TLRU) (BILAL; KANG, 2014), Aging Popularity-based Caching scheme (APC) (LI; LIU; WU, 2013), Frequency-Based-FIFO (FB-FIFO) (GOMAA et al., 2013), and Adaptive Replacement Cache (ARC) (MEGIDDO; MODHA, 2004).

However, there is no broader study on replacement schemes for ICN domains. Our thesis tackled that gap by cataloging the schemes proposed for ICN to investigate contextual influences on the policies. Therefore, our work does not seek to discuss individual policies, and the reader can refer to the original literature to further information. To support the reading of the following chapters, Table 2.1 presents a description of content placement and replacement policies reported along with the chapters.

Abb.	Policy Name	Type	Description	Ref.
LRU	Least Recently Used	Replacement	Removes the last accessed content in the cache.	-
LFU	Least Frequently Used	Replacement	Removes the last frequently used content in the cache.	-
FIFO	First-In-First-Out	Replacement	Removes the oldest content placed in the cache.	-

Continued on next page

Table 2.1 – continued from previous page

Abb.	**Policy Name**	**Type**	**Description**	**Ref.**
-	Random	Replacement	Removes one content randomly.	-
-	Size	Replacement	Removes the content with largest size in the cache.	(ABRAMS et al., 1996)
TTL	Time-to-Live	Replacement	Removes expired contents.	(BERGER et al., 2014)
LRFU	Least Recently/Frequently Used	Replacement	Considers the recency and frequency of contents to compute a Combined Recency and Frequency (CRF) metric. CRF values are higher for more recent and frequent contents. The policy evicts contents with lower CRFs.	(LEE et al., 2001)
FCDC	Fast convergence caching replacement algorithm based on dynamic classification method	Replacement	Considers categories of contents by content's popularity and a popularity rank by categories. Contents in lower ranked categories can be evicted for ones in higher ranked categories.	(CHAO et al., 2013)
RUF	Recent Usage Frequency	Replacement	Considers categories of contents by similarity and a popularity rank by categories. Contents in lower ranked categories can be evicted for ones in higher ranked categories.	(KANG; LEE; KO, 2012)
EV	Energy efficiency cache scheme based on virtual round trip time	Placement / Replacement	Considers the energy consumption to store and to transport the content. Places the contents with storage energy smaller than their transport energy, and compares the energy saving of the cached contents with the energy saving of the passing content to evict the contents.	(WANG et al., 2014)
PBRS	Content-Popularity and Betweenness based Replacement Scheme	Replacement	Removes the content with the lower popularity. Computes the content popularity based on the content's requests and node's betweenness centrality.	(LIU et al., 2019)
ABC	Age-based Cooperation	Replacement	Removes the content based on content's Time-to-Live (TTL). Computes TTL based on the node's location in the topology and the content popularity. The closer to the edge and/or the more popular a content, the longer its TTL value. Also called TTL.	(MING; XU; WANG, 2012)
2Q	Two Queues	Replacement	Designed for buffer management, it considers two lists of pages. The first list applies FIFO in the incoming page requests. The second list receives the pages in the first list requested again and their subsequent requests and applies LRU.	(JOHNSON; SHASHA, 1994)
ARC	Adaptive Replacement Cache	Replacement	Designed for buffer management, it considers two LRU lists. The first list contains pages requested once in a recent time, and the second list pages requested at least twice. The policy adaptively decides the number of pages to maintain in each list according to the workload characteristic.	(MEGIDDO; MODHA, 2003)
LIRS	Low Inter-reference Recency Set	Replacement	Designed for buffer management, it considers the number of other pages accessed between the last and penultimate access for a page as Inter-Reference Recency (IRR) metric. The policy removes the page with the largest IRR.	(JIANG; ZHANG, 2002)
MQ	Multi-Queue	Replacement	Designed for buffer management, it considers multiple lists with different access frequencies for different periods.	(ZHOU; PHILBIN; LI, 2001)
PPC	Popularity Prediction Caching	Replacement	Designed for video content. Predicts and caches the future most popular videos' chunks based on the number of requests for neighboring chunks in the same video content. Evicts chunks with the least future popularity.	(ZHANG; TAN; LI, 2018)

Continued on next page

Table 2.1 – continued from previous page

Abb.	Policy Name	Type	Description	Ref.
CCP	Cache Policy based on Content Popularity	Replacement	Considers previous content popularity and the number of hits in a current interval of time to compute the current content popularity. The policy evicts less popular content.	(RAN et al., 2013)
Betw	Betweenness centrality	Placement	Considers the node's position at the topology in terms of node's centrality measures to place the content. Only selected nodes with higher measures cache the content. Also called Leave-Copy-Betw (LCB), or Centrality.	(CHAI et al., 2012)
LCD	Leave Copy Down	Placement	Places the content only in the immediate downstream node of a cache-hit point.	(LAOUTARIS; SYNTILA; STAVRAKAKIS, 2004)
LCE	Leave Copy Everywhere	Placement	Places the contents in all caches along the reverse path of the content request.	(LAOUTARIS; SYNTILA; STAVRAKAKIS, 2004)
Prob	Probabilistic caching	Placement	Each cache in the reverse path of the content request stores the content with a constant probability p. Also called Leave-Copy-Probabilistically (LCP).	(LAOUTARIS; SYNTILA; STAVRAKAKIS, 2004)
-	ProbCache	Placement	Considers the shared storage capacity of the request path and the node's distance to the content producer to calculate the node's probability of caching the content; Also called PProb.	(PSARAS; CHAI; PAVLOU, 2012)
-	CRCache	Placement	Considers the content popularity and the node's centrality measures to calculate the probability of caching the content. The most popular contents are cached in the nodes with the highest centrality. Also called Cross.	(WANG et al., 2014)
PCP	Progressive Caching Policy	Placement	Considers the immediate downstream node of a cache-hit point to store the content, the number of interfaces saved in PIT entry for the intermediate nodes, and the number of requests for edge nodes.	(WANG; BENSAOU, 2012a)
Rand	Single node random caching	Placement	Places the contents in one random intermediate node along the delivery path.	(EUM et al., 2012)

Table 2.1: Set of content placement and replacement policies.

2.4 CACHING POLICIES AND HUMAN CONTEXT

The researches concerning caching policies comparisons in different network scenarios usually investigate variations of parameters that might indicate a direct relation with cache behavior, such as network topology, cache size, average size of transmitted files, and content popularity distribution. However, recent studies have been investigating the influence of human behavior on the network operation, enhancing an emergent research field of *people-centric networking* (CONTI et al., 2015; OLIVEIRA et al., 2016; CHEN et al., 2016; COSTA et al., 2018). The *human aware* paradigm is an emerging theme in the development of computer networking solutions. In this paradigm, the characteristics of human behavior can be incorporated into processes and applications, making the network efficiently adapted to the specific needs of users. Thus, the user is no longer seen as a generic element of the network, and is introduced as a new context aspect capable of influencing the network performance and the decisions of protocols and solutions used.

The new paradigm draws attention to the fact that network performance may vary according to user profile, and characteristics of applications. Both of them can be associated to a group of users. User behavioral profiles offer new inputs in the development of network protocols and strategies, and contribute to the consolidation of the concept of human-centered networks. Hence, network architectures can be benefited from user profile characterization, which can vary from different perspectives, from people's mobility regular patterns to community and social relationships habits.

In ICN architectures, user behavioral data can be explored in the development of strategies aimed at improving the performance of such networks, since they are based on a consumption/request model centered on the content holder. In the following paragraphs we summarize some related works that attempted to incorporate features related to the user in the caching process.

Some caching policies proposed for ICNs intended to incorporate features related to the user in the caching process (AL-TURJMAN; AL-FAGIH; HASSANEIN, 2013; XING et al., 2017; ZHANG; TAN; LI, 2018). However, the human characteristics are not directly used by the policies. For instance, Wei et al. (2014) proposed a mobility-aware caching strategy for mobile networks in which they model the transition of users among WiFi access points as a stationary Markov model. In a broad sense, the user's mobility has the same connotation as the node's mobility. In the surveyed works that deal with mobility, the concept of a node's mobility suits the objectives since the human dimension is not directly associated with mobility patterns. Different user's profiles can be associated with different mobility patterns, for example, different ages or professions (LIANG et al., 2012) or even different personalities (CHORLEY et al., 2013).

There are other initiates recognizing the relevance of the user and attempting to incorporate aspects related to the user in the caching process. For instance, Al-Turjman, Al-Fagih e Hassanein (2013) incorporate the past requesting preferences of groups of users as tuning parameters to assign a weight value for the types of contents accessed, and accordingly adjust the replacement policy to maintain the most suitable types to be accessed by the same group of users. The three main types of content are delay-based, demand-based, and age-based contents. Although they involved the idea of users requesting preferences in the process, the user is still represented just as a node, like a smartphone, or a laptop.

Another work related to the user request pattern was proposed by Xing et al. (2017). The authors used Neural Networks to model and predict the request pattern of groups of users in cellular partitions. They introduced a metric called Accumulate Network Traffic (ANT) to quantify users interest in contents, which represents the overall network flow generated to get a content, and it is based on a set of additional content's features. In this way, it is possible to associate different measures of ANT with different request patterns. The contents are cached according to their ANT value, and the proposed replacement scheme uses the ANT value in the eviction process.

Zhang, Tan e Li (2018) attempt to model the behavior of users when watching videos, aiming to model the relationship between chunks of the same content video type. This relation is the base of a prediction process that utilizes the request information of neighboring chunks to make the popularity prediction for future chunk requests. The proposed replacement scheme works with the predicted future popularity rank to decide which chunk to evict. As the previously mentioned works, this one deals only with content's properties, and do not direct models human attributes.

The work presented by Neves et al. (2013) reproduces a variety of scenarios with the intention of finding the best policy for streaming media in CDNs. The scenarios presented different combinations of video sizes, popularity models, number of requests, cache size and user session duration. Although the work includes a parameter related to user behavior, through the average time a user watches videos – classified as a discrete variable, i.e., short or long session – the analysis does not explore this perspective.

Another little-explored approximation of user behavior can be found in (ROSENSWEIG; MENASCHÉ; KUROSE, 2013). The authors investigate the ergodicity of content networks and its relationship to cache replacement policies. In one of its examples analyzing the effects of different initial states, the work succinctly discusses the influence of user request patterns, and concludes that small changes in request patterns can have a significant impact on policy behavior.

In contrast, the work of Bernardini, Silverston e Festor (2014) directly involves the user in defining cache policies. The authors propose a new cache location policy that looks at the number of connections a user has on their social networks. Users with many connections are considered "influential" and their contents receive a different treatment on the network, being proactively replicated in the caches towards the user's social connections. Simulations with synthetic data show a better performance of this new policy compared to standard policies used in ICN networks.

Furthermore, it is possible to find works that propose the identification of behavior patterns of content producers, seeking to minimize the handoff effects (LEHMANN; BARCELLOS; MAUTHE, 2016; ARAUJO; SOUSA; SAMPAIO, 2018); the use of cache policies according to user context (Ribeiro; Sampaio; Ziviani, 2018); the choosing cache location from users daily routines (SILVA; CAMPISTA; COSTA, 2016); and the forwarding of interests based on forecasting the relocation of producers (ARAUJO; SOUSA; SAMPAIO, 2018).

Although such initiatives present contributions in ICNs architectures, they still timidly explore the behavioral profiles of users in their solutions. Moreover, there is a gap in the literature regarding evaluations of how behavioral profiles of users can influence the cache replacement policies adopted in ICN architectures.

2.5 CHAPTER SUMMARY

ICN proposes a new Internet architecture model centered on the content distribution on the network. The core fundamental of ICN is decoupling the content delivery from the content location address. The model values the content's name instead of its location. Therefore, ICN counts with the in-network caching functionality to deliver contents and caching policies to manage the caches efficiently. There are two main categories of caching policies: placement and replacement policies. The placement policy decides if the cache should store a passing content, while the replacement policy decides which content to evict to free up memory space for new contents. Several policies are proposed in the literature, and our work focuses on the choice of replacement policies. Moreover, there are recent researches on the advantages of integrating human characteristics in the design of caching systems. The chapter explores some initiatives on that subject.

Chapter 3

CONTEXTUAL DIMENSIONS FOR CACHE REPLACEMENT SCHEMES IN ICNS

Based on the principle that cache replacement policies perform differently according to the context, we first attempted to define *what is context* from the perspective of the caching policies. A fundamental question to direct this process is to understand what can influence the performance of the policies. In this direction, we have conducted a Systematic Literature Review (SLR) to investigate evidence in the ICN literature about the effects of context aspects on cache replacement schemes' performance. SLR is a straightforward and consistent process to compile evidence to answer a set of research questions and help further understand the evidence reported.

In this chapter, we first present an outline of the systematic review (Sec. 3.1). Next, we describe the methodology adopted to perform the review, including the definition of specific research questions that drove the review process (Sec. 3.2). Then, we discuss the extracted evidence to answer the research questions, and present analyses of the main findings (Sec. 3.3). Moreover, we argue the applicability of our main findings in emergent ICN-enable scenarios (Sec. 3.4). Last, we discuss remaining aspects on the lessons learned (Sec. 3.5) and summarize the chapter (Sec. 3.6).

3.1 SYSTEMATIC LITERATURE REVIEW OUTLINE

Following a predefined SLR protocol, we first cataloged the cache replacement schemes used in ICNs. The current literature presents various proposed strategies exploring different context aspects to enhance the eviction logic, aiming to achieve more potentially precise and customized techniques. We mapped context dimensions related to the content, network, node, and human aspects. We then categorized the respective context properties used by the replacement schemes proposed for ICNs. With the context properties, we provide a taxonomy of context dimensions and a policy categorization accordingly. Taxonomies may support the choosing process in the absence of the overall understanding of specific network contexts and what influences policy behavior.

In addition to the taxonomy, we compiled the context variations with reported relevant impacts on the policies, especially those leading to changes in the policies' performance rank. This SLR was able to identify common context factors that differentiated the choice of best policy performance. Even so, as expected, there is no single optimal strategy to meet the requirements of all surveyed network contexts, since the performance of the caching policies varied according to the characteristics of each network.

Last, we extended the SLR results with the analysis of proper context dimensions to be explored by the eviction process in different emergent networks, such as information-centric IoTs (ARSHAD et al., 2018; DONG; WANG, 2016), vehicular named-data networking (KHELIFI et al., 2020), and in-network cache-based edge computing solutions (ZHOU et al., 2017; ZHANG et al., 2018). These emergent networks have gained attention from the research and industry communities, fostering the evolution of heterogeneous ICN solutions. The taxonomy and policy classification presented in this paper can help to infer the choice among current or new policies adapted to these networks to ensure better network performance.

Hence, as mentioned before, the SLR contribution is threefold. It (i) provides a classification of contexts to assist those engaged in the design of adaptive caching solutions for ICN that target the more efficient use of available cache resources; (ii) substantiates the reasoning of the caching policy decision process by presenting and analyzing information from previous works; and (iii) contributes to the set of knowledge on caching systems regarding emergent networks and underpins context-aware caching solutions.

3.2 SLR METHODOLOGY

The SLR methodology specifies a well-defined searching protocol, with the definition of research questions, research strings, explicit inclusion criteria of papers, among other steps. The methodology used in this paper follows an adaptation of previously adopted SLRs in the Software Engineering discipline (KITCHENHAM; CHARTERS, 2007; PETERSEN et al., 2008). Figure 3.1 summarizes the adopted SLR process.

The planning process ensures delimitation of the search scope with the definition of leading research questions, inclusion criteria, and the necessary inputs to operate the search. The search process is the paper triage phase to collect relevant works and extract meaningful data that match the research questions. The data analysis evaluates the extracted data to summarize the primary evidence and point contributions. We detail the processes activities in the following.

This study aimed to map context information associated with the performance of cache replacement strategies to help the choosing and design process while applying ICN. Since the scope and definition of context information can be relative to the research domain, we intended to characterize relevant dimensions surrounding the cache replacement schemes. Additionally, we also intended to identify the cache replacement strategies applied and their context characteristics, and investigate reported evidence about how the identified context information influences the behavior of cache replacement policies in ICNs. To this end, we defined the following research questions (RQs):

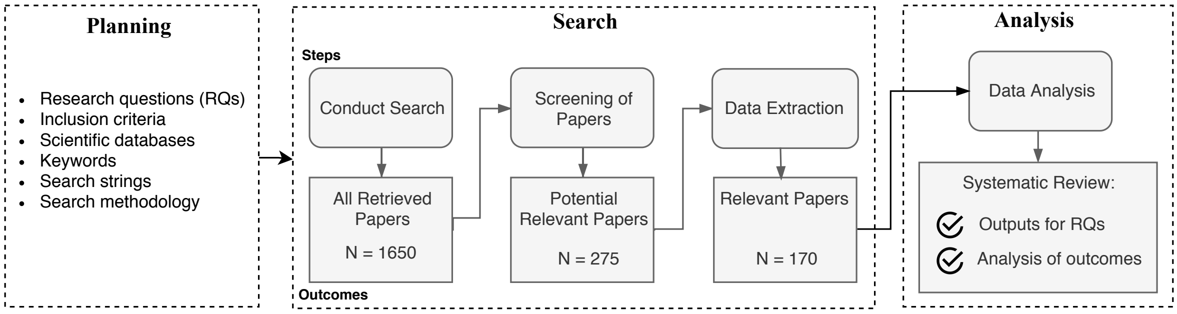


Figure 3.1 Steps of the SLR process.

RQ1: What is context from the perspective of a cache replacement policy?
RQ2: Which are the context characteristics used by the policies?
RQ3: Which are the cache replacement strategies applied for in-networking caching in ICN?
RQ4: What context variations had effects on the performance of the cache replacement strategies?

Notice that the research questions correlate with each other in the sense that they rely on each other's outputs in different ways: the first three questions are requisites to answer the last question; to answer RQ2 it is necessary a primary overview direction for RQ1 and also the output for RQ3; the complete delimitation of context that answers RQ1 is an iterative process that relies on RQ2 and RQ3 outcomes.

After the definition of the research questions, we specified a list of relevant keywords based on the analysis of manually selected papers, and we defined correspondent search strings using AND and OR operators, as exemplified in Table 3.1. The search strings were meant to drive automatic searches on relevant research engines. Table 3.2 contains the list of the selected scientific databases. We adapted the search strings according to the syntax of the scientific databases.

(”ICN” OR ”NDN” OR ”CCN” OR ”information centric” OR ”information-centric” OR ”named data” OR ”named-data” OR ”content centric” OR ”content-centric”) AND (“cache” OR “caching”) AND (“replacement” OR “eviction” OR “performance” OR “management” OR “policy” OR “policies”)

Table 3.1 Search string example.

The selection criteria included works written in English, addressing any aspects of cache replacement policy comparisons in ICNs. We also had the papers approaching new schemes for the eviction process for ICNs as part of the contributions.

After the planning phase, we applied the search activities. The first step of the searching process was applying the automatic searches as specified in the planning phase. We did not set a lower year threshold in the search databases for the publication year

Engine	**URL**
ACM Digital Library	https://dl.acm.org
Elsevier - Science Direct	https://www.sciencedirect.com
IEEE Xplore Digital Library	https://ieeexplore.ieee.org
Springer	https://link.springer.com

Table 3.2 Academic search databases.

range, and the upper bound was set to 2019. We cataloged a total of 1650 papers in this phase. In the following, the screening process comprehended abstract reading and analysis of all matched papers, to filter according to the inclusion criteria. Upon abstract filtering, we obtained 275 papers. Those were potential works where we could find answers to the predefined research questions. Finally, we performed full paper reading and analysis of the potential works to extract relevant information and evidence about the research questions. As a result, we reached a total of 168 papers pertinent to our research. Additionally, we incremented the results by carrying out a non-systematic snowballing research process on the read papers and search engines to update with new works not covered in the first search. This process resulted in the addition of two relevant documents.

Finally, the resulting papers and their correspondent extracted data consisted the input for our study. In the analysis phase, we have categorized and correlated data from different papers to empirically mining relevant information patterns. We report our main findings regarding the research questions in the following section.

3.3 SLR RESULTS AND ANALYSIS

The SLR process described in the previous section enabled us to answer the main research questions introduced in this manuscript. The following subsections describe the process to accomplished this.

3.3.1 Research Question 01 - Context Dimensions

As a result of the literature review analysis process, our definition of *context* comprises, in a broad sense, information that can be used by the policy as input data to direct the eviction process. Also, it includes information “external” to the policy that can be used within a computational environment and could influence the policy’s performance.

To understand and delimit what entities could represent the context from the perspective of cache replacement strategies, we direct the paper reading and extraction of possibly relevant information based on leading questions related to the content. Since the process of dealing with contents is the overall purpose of having caching policies, we placed content as a feedstock for caching policies, and we defined questions from the content’s point of view, as follows:

- *What content is being requested?* In this dimension, we seek for characteristics of the content itself (and the application), such as content size, popularity, type;

- *When is the content requested?* This dimension specifies time-related information regarding the content and its relation to the user—for instance, time of access, time of creation, or user delay to receive the content.

- *Where is the content located and distributed?* This dimension specifies network characteristics, such as topology and link capacity, and features about the node/routers that store the content, such as cache capacity and the number of interfaces.

- *Who is requesting the content? Also, who is publishing the content?* This dimension relates to the human aspect, in which preferences, behavior, and routines are mapped as a context dimension. The dimension can also refer to machine-to-machine communication, but, in this case, the characteristics overlap with information of the node contemplated in the previous dimension.

Therefore, we extracted relevant information that would apply to these dimensions and correlate with the cache replacement schemes. Based on the extracted data, we characterized context dimensions according to four main categories: *network, node, content*, and *human*. Figure 3.2 illustrates the hierarchy of our classification. A *context view* is represented by current information of cached content in a particular node, which belongs to a network, and is accessed or produced by a user. Each of these dimensions contains properties related to the cache eviction process in one or more of the surveyed papers. We detail the list of properties in the next subsection.

Additionally, we also consider ICN architecture decisions as part of the context. The other cache-related protocols, such as placement policies or naming schemes, are relevant aspects and should be included as part of the context. This dissertation surveyed the impacts of different architecture decisions on the replacement schemes; however, the discussion of specific caching protocols properties is out of the scope of this work.

3.3.2 Research Question 02 - Context Characteristics

Our second research question aims at identifying the context characteristics directly related to the policies. To this end, we collected the types of information used as input data for the replacement schemes and classified correspondent properties for the main context dimensions of Figure 3.2. We further discuss the context characteristics as follows:

- The **content** dimension is subcategorized into four types of properties: *feature, popularity, time-related*, and *type-specific*. The *feature* properties are global ones, i.e., are inherent to the content and usually do not vary according to the other context dimensions. Conversely, *popularity* and *time-related* properties are related to the node caching the content, and consequently, their values differ from node to node. The *type-specific* subcategory is reserved for specifying singular aspects of data or application types. Figure 3.3 contains a list of properties extracted from the surveyed papers for the content dimension. In this case, the type-specific properties are mainly about video content, for illustrative purposes.

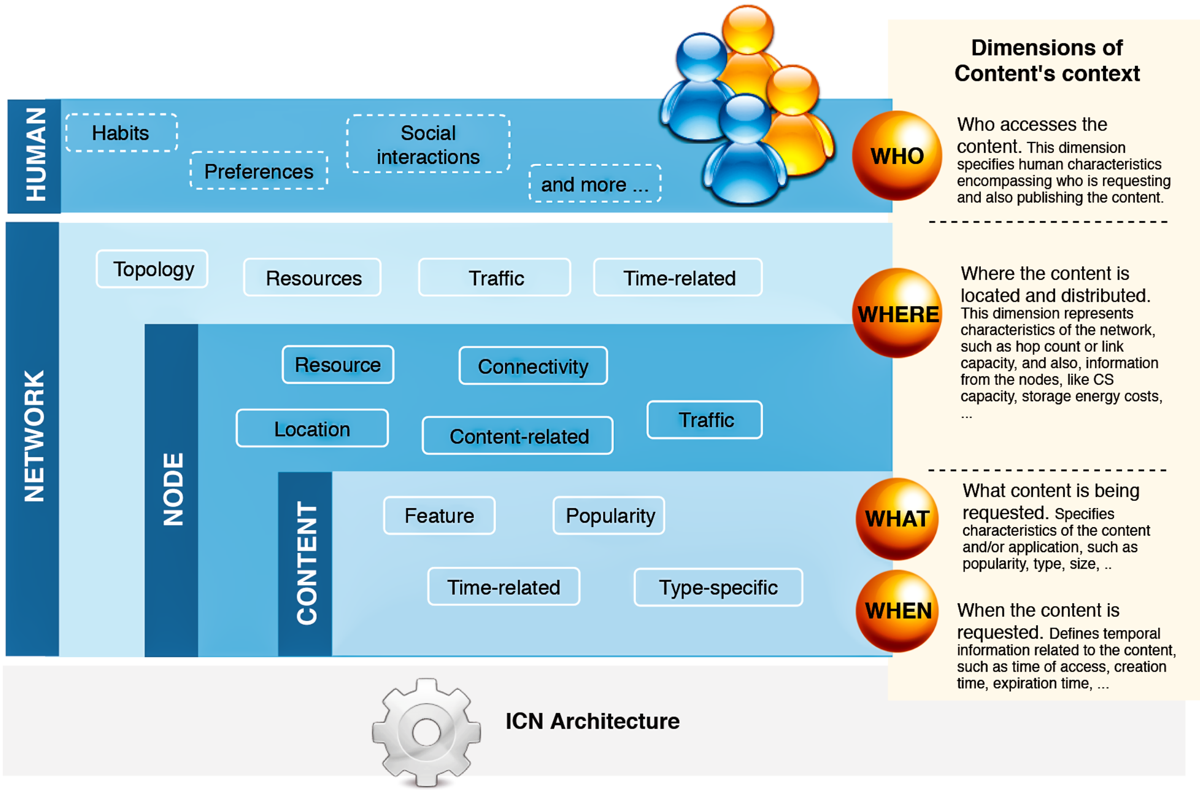


Figure 3.2 The hierarchy of context dimensions identified from the surveyed papers and the proposed classification for the correspondent characteristics associated with the cache replacement schemes.

- The **node** dimension is subcategorized in *resource, connectivity, location, content-related*, and *traffic*. Accordingly, the *resource* properties are inherent to the node; *connectivity* and *location* features are mostly related to neighbor nodes and the position of the node into the topology. The *content-related* represents the intersection with the content's dimension and gathers content's information in a broader granularity. The *traffic* properties are related to the flows of data traffic passing through the node. Figure 3.4 shows the list of properties extracted from the surveyed papers for the node dimension.

- The **network** dimension represents properties common to general network types. The properties are categorized into four classes: *resource, topology, traffic*, and *time-related*. The *resource* class groups the overall network capabilities, such as bandwidth, link capacity, and fetching content costs. The *topology* properties are more specific about network's size, represented mainly with the distances between nodes. The *traffic* class has the same connotation as in the node dimension but differs in granularity, and the *time-related* class defines temporal properties. The presented properties in the time-related class are similar to some of the topology properties. They are related to the distance between nodes measured in time units to reflect the delay to retrieve content. Figure 3.5 presents the list of properties extracted from the surveyed papers for the network dimension.

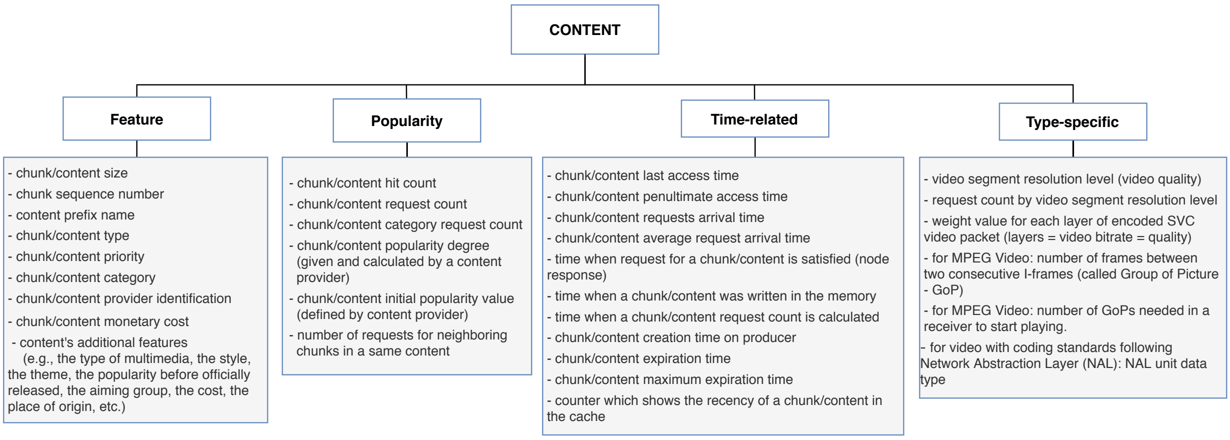


Figure 3.3 Properties from content dimension extracted from the cache replacement schemes for ICNs.

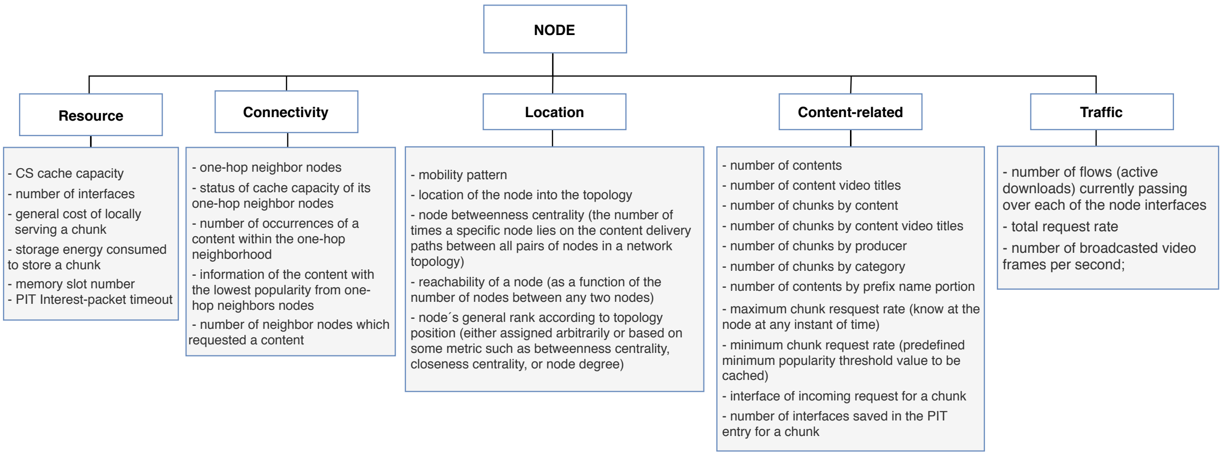


Figure 3.4 Properties from node dimension extracted from the cache replacement schemes for ICNs.

The previous list of properties is a broad definition of context characteristics to assist in the analysis of cache replacement schemes. It helps to visualize what dimensions are directly related to the policies and could significantly impact the applied network context. However, it is not a static list and can be increased as new information becomes available and relevant to a specific ICN instance. Furthermore, some of those properties are closely related to more than one context dimension. It is possible to change their perspective

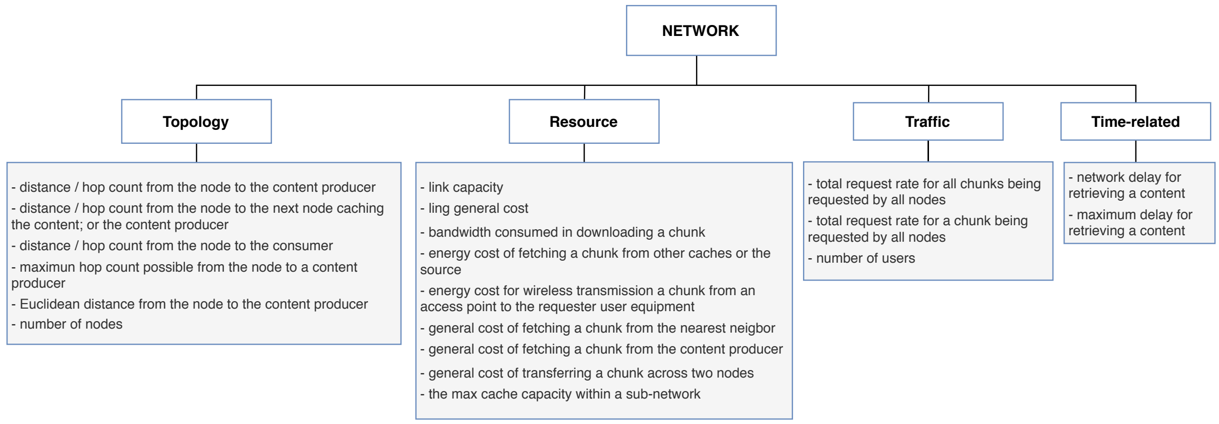


Figure 3.5 Properties from network dimension extracted from the cache replacement schemes for ICNs.

in terms of classification to represent a given ICN context. Moreover, the unified view of properties can substantiate the design of novel cache solutions by helping to identify potential gaps for new situations.

The **human** dimension is an emergent and new approach to be explored as part of the context. In this way, human attributes are potential drivers in the design of network solutions. The many examples of human data, such as *behaviors, interests, personality, character, social interactions, humor, daily routines, gender, age, etc.*, opens up a range of possibilities to be explored. We have performed experiments with real user data and associated distinct user habits with different cache replacement policies (See Chapter 4). The experiment reinforces the relevance of the human dimension for network configuration. However, it is an incipient research field, and there is still a lack of studies intersecting human features with caching policies. Thus, it was unsuitable for proposing a proper classification of properties for the human dimension in the current research.

3.3.3 Research Question 03 - Cache Replacement Schemes for ICNs

The literature shows various proposed replacement schemes for ICNs exploring beyond the context of the content and adding properties of node and network's dimensions. In this direction, we cataloged the replacement schemes applied to the surveyed papers to collect context features and understand their correlations. To better readability, we classified the schemes according to the classes of context information they used. They are classified in: *content-based; content and network-based; content and node-based; content, network and node-based;* and *network and/or node-based* schemes. Tables 3.3, 3.4, 3.5, 3.6, and 3.7 contain the lists of the cache replacement schemes in each class, respectively.

The tables also detail the correspondent context property categories used by the policies, which reveal the diversity of context combinations explored in the literature. We grouped the policies accordingly. This classification provides a comprehensive view of what context information the techniques required. Therefore, it is the first guide to

Content property categories	Replacement schemes
Popularity	(ROSSI; ROSSINI, 2011),(CHAO et al., 2013),(RAN et al., 2013),(YEH et al., 2015),(NAKAYAMA; ATA; OKA, 2015),(LIU; ZHU; MA, 2016),(ZHAO et al., 2017),(KALGHOUM; GAMMAR, 2017),(SINKY et al., 2018),(LI; YU; LI, 2018),(KALGHOUM; SAIDANE, 2019).
Time-related	(RAVI; RAMANATHAN; SIVALINGAM, 2014),(LI; MA; HU, 2015),(REZAZAD; TAY, 2015),(RHAIEM; FOURATI; AJIB, 2016),(SHUKLA; ABOUZEID, 2017),(DHIAB et al., 2017),(VURAL et al., 2017),(HOU et al., 2019),(MEDDEB et al., 2019),(DIN et al., 2019).
Popularity and Time-related	(WANG et al., 2012),(SANTOS et al., 2013),(QIAN et al., 2014),(CHEN et al., 2014),(ABIDI; GAMMAR, 2015),(XIN et al., 2016),(YAO et al., 2018),(CHOOTONG; THAENTHONG, 2017),(ZHANG; TAN; LI, 2018),(HUANG et al., 2018),(TANG et al., 2019).
Popularity, Time-related and Feature	(KANG; LEE; KO, 2012),(BILAL; KANG, 2014),(HAN et al., 2014),(BILAL; KANG, 2017),(PRAKASH; MOHARIR, 2018),(SERTBAŞ et al., 2018).
Time-related and Feature	(THOMAS; XYLOMENOS, 2014),(RAO; SCHELEN; LINDGREN, 2016),(WU et al., 2014),(TARNOI et al., 2019).
Popularity and Feature	(CHANDRASEKARAN; WANG; TAFAZOLLI, 2015; CHANDRASEKARAN et al., 2018),(LEE; HONG, 2017).
Popularity and Type-specific	(JIA et al., 2016),(GE et al., 2016).
Time-related and Type-specific	(ZHANG et al., 2017)
Time-related, Type-specific and Feature	(GHAHFAROKHI; MOGHIM; EFTEKHARI, 2017)
Type-specific and Feature	(LEE; LIM; YOO, 2013)
Popularity, Time-related, Type-specific and Feature	(LEE; LIM; YOO, 2013)

Table 3.3 Content-based cache replacement schemes.

map which context variances could directly influence the performance of the technique.

The replacement policies explore one or more features listed in Figures 3.3, 3.4, and 3.5, according to the classification. Naturally, almost all the schemes further explore the content dimension; however, we also found methods dealing only with network and node features to assist the eviction process. Figure 3.6 illustrates the usage distribution of context properties by their categories. We ranked the context categories according to the number of policies that used one or more of the corresponding category properties. It is important to remark that for the classification of policies, we did not account for the general use of node CS cache capacity and the number of interface information, since it can usually be part of the caching process.

3.3.4 Research Question 04 - Effects of Context Variation

Our objective in this section is to carry out an evidence-based analysis and identify what context dimensions can affect the policies' performance. In this way, we collected

Content property categories	Node property categories	Replacement schemes
Popularity	Location	(WEI et al., 2014),(CHEN et al., 2016),(MICK; TOURANI; MISRA, 2016),(LAL; KUMAR, 2019).
	Content-related	(LAL; KUMAR, 2016),(ZHANG; TAN; LI, 2017),(BAUGH; GUO, 2018).
	Traffic	(SALTARIN et al., 2018)
	Traffic and Connectivity	(YANG; CHOI, 2018)
	Traffic and Location	(LIU et al., 2019)
Popularity and Time-related	Traffic	(KARAMI; GUERRERO-ZAPATA, 2015),(ROCHA et al., 2016),(ZHOU; YE, 2017),(KHAN; KHAN, 2017),(QU et al., 2018).
	Connectivity	(AN; LUO, 2018)
	Content-related and Traffic	(YAO et al., 2019)
Time-related and Feature	Content-related	(HAHM et al., 2016)
	Connectivity and Location	(AOKI; SHIGEYASU, 2017)
Popularity, Time-related and Feature	Connectivity	(WOOD et al., 2013)
	Content-related and Traffic	(ONG et al., 2014)
Popularity and Feature	Content-related	(LI et al., 2015),(DRON et al., 2013).

Table 3.4 Content and Node-based cache replacement schemes.

Content property categories	Network property categories	Replacement schemes
Popularity	Topology	(WANG et al., 2011; WANG; BI; WU, 2012),(MING; XU; WANG, 2012),(REN et al., 2014),(HU et al., 2015), (HUANG et al., 2017),(KHAN et al., 2018).
	Resource	(CAARLS; HARGREAVES; MENASCHÉ, 2015)
	Traffic and Time-related	(SINKY et al., 2018)
Popularity and Time-related	Topology	(CHEN; FAN; YIN, 2013),(OSTROVSKAYA et al., 2018).
	Time-related	(YOKOTA et al., 2016)
	Resource	(PAL; KANT, 2017)
Popularity and Feature	Resource	(WANG; BAYHAN; KANGASHARJU, 2015)
	Time-related	(SUN; WANG, 2015)
	Resource and Time-related	(NDIKUMANA et al., 2018)
Popularity, Time-related and Feature	Topology	(DUAN et al., 2013)
Time-related	Time-related	(DAI et al., 2017)
Feature	Resource	(XING et al., 2017)

Table 3.5 Content and Network-based cache replacement schemes.

reported evidence from the surveyed papers about the effects of context variations on replacement schemes' performance.

We have found policy comparisons in different network types with variations of many

Content property categories	**Node property categories**	**Network property categories**	**Replacement schemes**
Popularity and Feature	Content-related and Location	Topology	(Panigrahi et al., 2014)
	Content-related and Traffic	Traffic	(LIU et al., 2018)
	Traffic	Resource	(BADOV et al., 2014)
	Resource	Time-related and Resource	(GÜR, 2015)
Popularity and Time-related	Content-related	Topology	(RATH; PANIGRAHI; SIMHA, 2016)
	Traffic and Location	Topology and Time-related	(AL-TURJMAN; AL-FAGIH; HASSANEIN, 2013)
Popularity	Traffic	Topology	(CHEN et al., 2017)
	Connectivity	Topology and Resource	(ZHANG et al., 2016)
Time-related	Resource	Topology and Resource	(LLORCA et al., 2015)
	Location	Topology	(NAZ; RAIS; QAYYUM, 2016)
Popularity, Time-related and Feature	Traffic	Topology and Time-related	(AL-TURJMAN, 2017)

Table 3.6 Content, Node, and Network-based cache replacement schemes.

Node property categories	**Network property categories**	**Replacement schemes**
Content-related and Location	Topology	(WANG; BENSAOU, 2012a; WANG; BENSAOU, 2012b; YANUAR; MANAF, 2017)
Content-related and Resource	Time-related	(SURESHJANI; MOGHIM, 2018)
Resource	Resource	(WANG et al., 2014)
-	Resource	(IOANNIDIS; YEH, 2016; IOANNIDIS; YEH, 2018)

Table 3.7 Node and/or Network-based cache replacement schemes.

aspects such as request rates, forwarding strategies, number of consumers, number of contents, and overall topology. Nevertheless, in summary, we found that variations in the *node location, cache size, cache placement policy* and *content popularity* had some relevant effect on the policies' performance. The first three presented variations resulting in different choices of replacement policies. Also, beyond the impact on the choosing point of which cache replacement schemes to apply, variations in *cache size* and *content popularity* presented other relevant effects related to the policies' performance. We discuss the context variations separately in the following.

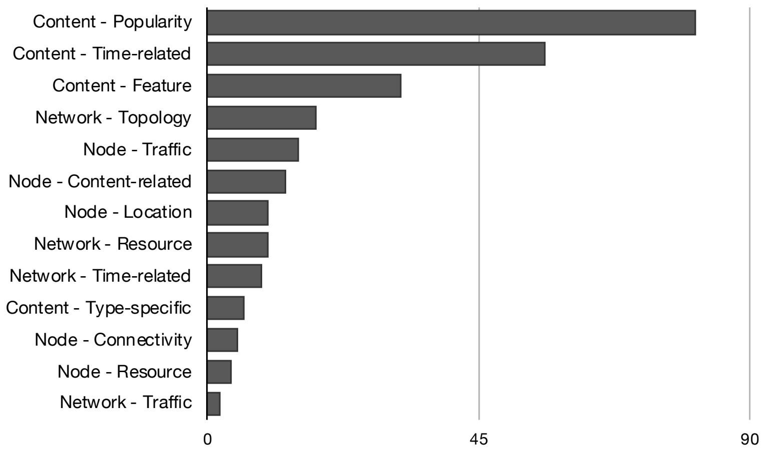


Figure 3.6 Distribution of context properties categories according to the number of policies that used the correspondent properties in their eviction logic.

	NETWORK	**NODE**		**CONTENT**	**ICN ARCHITECTURE**			
Ref.	**Topology**	**Node location**	**Cache size**	**Popularity (Zipf(a))**	**Placement policy**	**Eviction policy**	**Metrics**	**Effect**
(WANG; BENSAOU, 2012a)	Internet-like; 32 CRs.	Edge and Intermediate	[100-8000] chunks	[0,92; 0,78] and [0,96; 0,74]	LCE and PCP	LRU; Proposed edge and core.	Hit rate; Hit gain; Path stretch.	[*Node-location*] Different policies for different node locations; Limited evaluation.
(LI; SIMON; GRAVEY, 2012)	Internet-like; 40 CRs.	Edge and Intermediate	100 chunks	1.0	LCE	LRU, LFU and LRFU multi-γ.	Hit rate; Number of access to server.	[*Node-location*] Different configurations of LRFU for different node locations.
(TARNOI et al., 2014)	Cascade; 5 CRs.	Edge and Intermediate	10, 20, 50, and 100 objects.	[0.4-1.6]	LCE and Prob	LRU, LFU and Random	Hit rate; Server load; Round trip hop distance.	[*Node-location*] LRU and Random interchange positions for different node locations; [*Placement-policy*] Different eviction policies for different placement policies.
(TARNOI et al., 2014)	Internet-like; 50 CRs.	Edge and Intermediate	80, 160, 400, and 800 objects.	[0.4-1.6]	LCE and Prob	LRU, LFU and Random	Hit rate; Server load; Round trip hop distance.	*[Node-location]* LRU and Random interchange positions for different node locations. [*Placement-policy*] Different eviction policies for different placement policies.

Continued on next page

Table 3.8 – continued from previous page

	NETWORK	NODE		CONTENT	ICN ARCHITECTURE			
Ref.	**Topology**	**Node location**	**Cache size**	**Popularity (Zipf(a))**	**Placement policy**	**Eviction policy**	**Metrics**	**Effect**
(GALLO et al., 2014)	Tree; 3 CRs.	Edge and Root	[10-100] objects	1.7	LCE	LRU and Random	Miss probability	*[Node-location]* LRU and Random interchange positions for different node locations.
(NEWBERRY; ZHANG, 2019)	Data Center Fat-tree; 80 CRs.	Core, aggregation, edge	64, 128, 256, 512 and 1024 MB	-	LCE	LRU, 2Q, ARC, LIRS and MQ	Total network traffic	[*Node-location*] [*Cache-size*] Different policies for different node locations, for different applications, and for different cache sizes.
(CHAO et al., 2013)	-	-	[5-75] objects	1.0	-	FCDC, LRU and RUF	Hit rate.	[*Cache-size*] LRU and FCDC interchange positions for different cache sizes.
(WANG et al., 2014)	Cascade; 5 CRs.	-	aprox. [5-40] objects	0.8, 1.5, and 2.0	LCE and EV Placement	LRU, EV Replacement, and Popu.	Energy efficiency	[*Cache-size*] Different combinations of placement and replacement policies for different cache sizes.
(LIU et al., 2019)	Tree; 7 CRs.	Edge and Intermediate	[10-90] MB	0.7	LCE	PBRS, LRU, LFU and FIFO	Hit rate	[*Cache-size*] LFU and PBRS interchange positions for different cache sizes.
(SUN et al., 2014)	Internet-like; 80K CRs.	Edge, Middle, and Core	1GB, 10GB, 100GB, and 1TB	1.174	LCE, LCD, Rand, Prob, PProb, Centrality and Cross	LRU, LFU, FIFO, TTL and Size	Hit rate; Traffic reduction; Server load reduction	[*Cache-size*] [*Placement-policy*] Different eviction policies for different cache sizes, and for different placement policies.
(CHEN et al., 2016)	Wireless Mesh; 15 CRs.	-	210 bytes	0.8	LCE, LCP, LCD, and LCB	LRU, LFU, Random, and FIFO	Hit ratio; Energy consumption.	[*Placement-policy*] Different eviction policies for different placement policies.

Table 3.8: Scenarios concerning replacement policies evaluations with different effects on the policy choice. CR = Content Router; N.Cons. / N. Prod. = Number of content consumers / Number of content producers.

3.3.4.1 Cache Node Position in Topology The works from Wang e Bensaou (2012a), Tarnoi et al. (2014), Gallo et al. (2014), Li, Simon e Gravey (2012), Newberry e Zhang (2019) presented evidence of the impact of node's location on cache replacement scheme choice. Table 3.8 summarizes the reported underlying system parameters. Those are the context characteristics that supported the analyses. In the following, we discuss the reported impacts:

- Wang e Bensaou (2012a) proposed two complementary replacement algorithms to handle different workload characteristics observed by both edge and intermediate router nodes. The eviction logic uses the hop count factor to prioritize the maintenance of more distant contents and, consequently, reduce network resource consumption. Besides the hop count, the replacement algorithm for intermediate nodes considers the number of node's interfaces saved in the PIT entry for a content to estimate the diversity of the content requests. The proposed solution outperforms homogeneous configuration with LCE+LRU, and the results emphasize the benefits of using heterogeneous replacement policies according to the location of the node into the topology. However, the eviction solutions were evaluated only in conjunction with a proposed placement policy named PCP, limiting the analysis of the heterogeneous eviction solution separately. The proposed replacement schemes logic would be able to work with other location policies, such as LCE.

- Li, Simon e Gravey (2012) used the LRFU policy with a weighting parameter y to represent a multi-policy caching where every content router implements its caching policy according to its location in the network. The LRFU behavior can switch to be more closely similar to LRU or LFU according to the value of y. The router location is relative to his position between users and servers. The routers (CRs) are classified according to a defined "entering degree", which represents the number of the shortest path connecting front-end CRs with servers via a CR. The reasoning to configure different values of LRFU parameter y comes from an experiment under an emulated European Backbone Ebone topology with 40 nodes, in which they performed experiments with homogeneous configurations of y in all routers. They observed that the routers with lower hit rate achieved their best performance with higher values of y, and on the contrary, routers with higher hit rates achieved their best performance with lower values of y. Allied to that, they also observed that the position of the router in the hit rate rank is directly proportional to his position in the topology, in the sense that the closer to the edge, the higher is the hit ratio performance.

- The experiments of Tarnoi et al. (2014) reveal the difference of performances between LRU and Random according to the node position. For the experiment with a cascade network and one content requester, LRU and Random, in combination with LCE placement policy, interchange positions on the rank of the cache hit performance: for the level 1, LRU outperforms Random, but from level 2 onward, LRU performance decreases drastically and Random also slightly decreases but now with better performance than LRU. The difference in the rank of cache hit rate is similar for the experiment variation with multiple content requests, but LRU and Random interchange position after the third level node. For the Internet topology, the result groups edge and core nodes, and again, LRU presented the best results for edge nodes while Random for core nodes.

- Continuing the discussion about LRU and Random replacement policies, Gallo et al. (2014) came to a similar conclusion in terms of the difference in performance when varying node locations. For that, the authors presented an analysis of cache miss probability depending on the content popularity distribution. The analysis suggest that LRU and Random have significantly different performances only for popularity distributions highly concentrated on a relatively small number of objects. That difference is also relative to the position of the node in the topology. The more popular objects are more likely to be found at the edge node when using LRU, but those more popular objects can be more evenly distributed when using Random across the path. Due to the randomness of the eviction logic and its independence regarding other caches evictions, Random increases content diversity in the network. Also, the evaluation presents heterogeneous configuration for the leaves and root levels of a tree topology: LRU-Random and Random-LRU, also LRU-LRU and Random-Random. The heterogeneous LRU-Random configuration achieved better performance than the other configuration options, i.e., LRU and Random configured respectively in the edge and intermediate levels.

- While evaluating the advantages of integrating big data applications in an ICN-like architecture, Newberry e Zhang (2019) argue the benefits of using different cache replacement policies at each layer of a data center fat-tree topology. They compared the performance of homogeneous and heterogeneous policy configurations, placing the cache in each node of a fat-tree topology with three layers, composed of 16 core, 32 aggregation, and 32 edge switches. They performed combinations of the policies LRU, 2Q, ARC, LIRS, and MQ, on the levels of the tree topology, totaling 125 combinations for each variation of cache size. The results could reveal the different behaviors at different layers of the topology and the suitability of different policies at each level. However, the gain of the reported best heterogeneous configurations regarding the best homogeneous configuration is not explicit in the paper.

All the network contexts discussed in this subsection concluded that heterogeneous policy configurations achieved the highest performances than the homogeneous configurations. Whether for small topologies (TARNOI et al., 2014; GALLO et al., 2014) or larger topologies (WANG; BENSAOU, 2012a; LI; SIMON; GRAVEY, 2012; TARNOI et al., 2014; NEWBERRY; ZHANG, 2019), the works observed different traffic characteristics in the different nodes. They attributed this difference to the node position and associated different policies to different traffic profiles.

Multiple levels of caches naturally present that difference in traffic characteristics by cache-level due to the knowing *filtering-effect*. The filtering-effect happens any time a lower-level cache hits a content request. The cache does not propagate that request to the rest of the network and propagates only the miss requests to upper-level caches. This behavior modifies the original characteristics of the traffic. Many studies have been addressing the progressive filtering effect in hierarchical web caches (WILLIAMSON, 2002;

ZHOU et al., 2013; MELAZZI et al., 2014). That filtering has a direct impact on the *temporal locality* of the requests (JIN; BESTAVROS, 1999). Temporal locality refers to the property that recently accessed objects are likely to be reaccessed in the near future. As cache levels filter requests, the temporal locality intensity becomes gradually weakening, and the traffic profile at upper-level caches becomes more random (JIN; BESTAVROS, 1999). Besides the content diversity in the network obtained with the Random policy, the temporal locality effect also explains why Random achieved better performances for intermediate nodes in some of the discussed works. As expected, workloads with temporal locality property have a strong correlation with caching policies (GARETTO; LEONARDI; MARTINA, 2016), and variations in the temporal locality patterns directly impact the variations of caching policies performances.

Regarding the context attributes explored by the replacement schemes, only two of the works presented evaluations including context features in the eviction logic that helped differentiate the node's position: the *node's number of interfaces* (WANG; BENSAOU, 2012a) and the *node degree* as a general rank according to the topology (LI; SIMON; GRAVEY, 2012). However, other works are exploring those, and other context attributes that could be helpful. The context attributes with their respective classification and reference works are:

- Node-Location: node betweenness centrality (CHEN et al., 2016; LIU et al., 2019);
- Node-Location: reachability of a node (Panigrahi et al., 2014);
- Node-Location: node's general rank according to topology position (MICK; TOURANI; MISRA, 2016; AOKI; SHIGEYASU, 2017; NAZ; RAIS; QAYYUM, 2016);
- Node-Content-related: number of interfaces saved in PIT entry for a chunk (WANG; BENSAOU, 2012a);
- Node-Connectivity: one-hop neighbor nodes (ZHANG et al., 2016);
- Node-Resource: number of interfaces (WANG; BENSAOU, 2012a; BAUGH; GUO, 2018).

Although the node's location is a context that should be considered when selecting a replacement policy, it is not easy to foresee a straight map between policies and node positions. First, because there are many policies and diversity of topologies with different requirements, but mostly because there are other contextual factors that can also impact the performance of the policies. As we continue to show in the next sections, this SLR was able to pinpoint some of these factors.

3.3.4.2 Cache Size The works from Chao et al. (2013), Wang et al. (2014), Sun et al. (2014), Newberry e Zhang (2019), Liu et al. (2019) contains evidence of cache size variations on the performance ranking variations of cache replacement policies. Table 3.8 summarizes the reported underlying system parameters. Those are the context characteristics that supported the analyses. In the following, we discuss the reported impacts:

- According to Sun et al. (2014), the replacement scheme's optimal choice depends on the cache size and the placement policy. The authors combined seven placement policies with five replacement policies - LRU, LFU, FIFO, TTL, Size - and cache size variations of 0.0007%, 0.007%, 0.07%, and 0.7% of the unique contents. The content routers have homogeneous cache sizes for all experiments. We observe that the most significant impact on the replacement scheme choice happens when passing from 0.0007% to 0.007% of cache sizes. That is, for all combinations of placement policies, the best choice of replacement scheme changed when the cache size moved from 0.0007% to 0.007%. Meanwhile, for most combinations of placement policies, the experiments running with 0.007%, 0.07%, and 0.7% of cache sizes presented their highest performance values with the same replacement policy. For example, combined with LCE, LRU and TTL achieved the highest performances for 0.007% of cache size, while LFU stands out for the other sizes.

- Chao et al. (2013) also show evidence that variations on cache size can lead to variations on the policy with the best performance. This work presents a content-based replacement policy named FCDC that manages the content popularity property - request count - to classify and replace contents according to popularity categories. The evaluation shows comparisons of the proposed scheme against LRU and RUF policies. According to the results, FCDC presents a better cache hit rate than LRU and RUF when the cache memory is less than 5%. Yet, the performance rank changed for cache sizes larger than 10%, and LRU performed slightly better than FCDC. The authors attribute this behavior to each policy's property, in which FCDC can keep track of content popularity and maintain the most popular content better than LRU for small cache sizes. At the same time, LRU prioritizes most recently accessed over the most accessed and popular content. However, this does not directly correlate to the performance differences according to the cache sizes. FCDC deals with dynamic changes of content popularity and does not directly rely on node information.

- Furthermore, the experiments performed by Wang et al. (2014) also reveal differences in policy performance rank while varying the cache size. The work proposes the EV policy, a node-based replacement scheme coupled with a placement scheme. EV was evaluated and compared against LCE+LRU and LCE+Popu - a referenced popularity-based policy. The configuration of the content popularity follows a Zipf distribution, and besides the impact of different cache sizes, the results also reveal a correlation with the popularity skewness factor. For α skewness factor equals 0.8, EV and Popu had similar performances for all cache size variations. Meanwhile, for $\alpha = 1.5$ or 2.0, the policies interchanged positions in the rank of average total energy consumption for different cache sizes: Popu achieved better performance than EV for cache sizes between 10 to 20% of total contents; for larger cache sizes, EV turns to be the better choice. The work does not provide an analysis of this effect. The results show the impact of cache size on placement and replacement schemes combined, limiting the evidence of the eviction scheme solely.

- Similarly, Liu et al. (2019) presented evidences of variations in the rank of hit ratio of the policies for different cache sizes. The work shows evaluations of a proposed replacement policy named PBRS against LRU, LFU, and FIFO. PBRS and LFU interchange positions for different cache sizes in a tree topology. This effect is most evident for intermediate nodes, in which LFU presented better results for cache sizes between 10MB to approximately 50MB, and PBRS presented better cache hit values for larger cache sizes. Both policies rely on content popularity, but LFU computes the popularity directly to count the number of requests, while PBRS increments the computation by adding different weights associated with the nodes.

- Finally, besides the effect of heterogeneous policies for different node locations in a fat-tree topology observed by Newberry e Zhang (2019), we also observed variations in policy performances' rank while varying cache sizes. The work evaluated LRU and other replacement policies named 2Q, ARQ, LIRS, and MQ policies. For a homogeneous policy configuration in all levels of the topology, the rank of policy performances did not change when using cache sizes from 64 to 512MB. However, when cache sizes varied from 512MB to 1GB, a couple of changes happened in the rank: first, LRU and 2Q interchanged positions, in which 2Q achieved better results than LRU up to 512MB, but LRU presented better results for 1GB; second, ARQ and MQ changed positions, with MQ presenting better results up to 512MB and ARQ with 1GB; and finally, LIRS and ARQ also changed positions in the rank, with LIRS presenting better results than all other policies up to 512MB, but ARQ achieved better performance with 1GB of cache size. For a heterogeneous policy configuration, the results presented similar effects on the rank. Without going into specific characteristics of policies, this work has evidence of the influence of cache size and the lack of simple patterns that associate the performance of cache policies with the size of the cache.

Regarding the impact on the replacement policy choice, in none of the presented works it is evident why variations in cache size led to different policy choices. Also, the analysis of the works does not reveal potential patterns due to the heterogeneity of the context factors. The works range from country-wide router-level topology with around 80K routers to a small and straightforward linear topology, with variations of placement and replacement policies, and different ranges of cache size evaluations. Although the evidence clearly shows the relevance of cache size in particular works, it is not sufficiently conclusive the why.

Yet, we cataloged other effects regarding variations on cache size and the performance of the policies. It is natural to expect an increase in the cache size should increase the performance gain for any caching policy since there is more space to store contents. In practice, the constraints of memory access speed or node devices' power will limit cache size. However, evidence shows that *caching policies' performance gain is not linear to the cache size increase* (HAN et al., 2014; CHEN et al., 2014; ONG et al., 2014; SUN et al., 2014; PIRES et al., 2018; MANGILI; MARTIGNON; CAPONE, 2013). In this way, adding cache resources on the network could not be the most suitable solution to improve the performance. The observed effect is because size allocation is a function of the content's popularity distribution. For example, for large amounts of non-popular content, the cache size may be small because the gain in caching is restrictive. On the contrary, for large amounts of popular content, the benefits will be best achieved for larger cache sizes. In this way, balancing optimal cache size in terms of cost and effectiveness of policies shall be done considering the fluctuations in content popularity.

Another observed effect of the relationship between cache size and replacement policy gain is that *as the relative cache size increases, the performance difference among the techniques decreases* (CHARPINEL et al., 2016; HAN et al., 2014; NAKAYAMA; ATA; OKA, 2015; BILAL; KANG, 2017; XING et al., 2017; Panigrahi et al., 2014; LI; SIMON; GRAVEY, 2012; FRICKER et al., 2012; NEWBERRY; ZHANG, 2019). That means the performances tend to converge eventually. Such an effect is in line with Che's approximation (CHE; TUNG; WANG, 2002), which we briefly discuss here. The longest possible time between two sequential hits for a content c present in the cache, i.e., before removing c from the cache, is expected to be random and related to c. That is the *cache eviction time* for content c. However, Che's approximation stands that, for reasonably large cache sizes, this cache eviction time tends do be deterministic to the point of being a constant irrespective of the content. Therefore, as cache size increases, the dependence on c decreases and becomes negligible. Following this direction, we infer that if the dependence on the content decreases, the dependence on which content will be removed also decreases since all contents converge to the same relevance in terms of eviction time. Although Che's approximation has been proposed for network contexts with LRU under Independent Reference Model (IRM), other extensions and generalizations also show the approximation's validity to other context' variations (GARETTO; LEONARDI; MARTINA, 2016; FRICKER; ROBERT; ROBERTS, 2012; ARALDO; ROSSI; MARTIGNON, 2015).

3.3.4.3 Content Placement Policy ICN in-path cache works as an opportunistic cache to distribute the content along with the network, and that opportunistic characteristic makes more flexible the distribution of caches on network nodes and the content location choices. Once there is a cache, though, the replacement scheme is mandatory for all cache nodes. Nevertheless, both content placement and replacement decisions are closely correlated and influence each other behaviors. The decisions can be implemented separately and combined according to the network requirements. Each combination of placement and replacement policies can lead to different behaviors. On the other hand, both placement and replacement strategies may complement each other. Some of the replacement schemes reported in ICN literature are already coupled with a placement strategy (SANTOS et al., 2013; SINKY et al., 2018; REN et al., 2014; HU et al., 2015; PAL; KANT, 2017; XING et al., 2017; MICK; TOURANI; MISRA, 2016; ZHANG; TAN; LI, 2017; WANG et al., 2014; CHEN et al., 2017; KHAN; KHAN, 2017) and deployed in conjunction.

In this work, we chose to look at the placement policy as a context factor that influences the replacement policy choice. This subsection presents the works (CHEN et al., 2016; TARNOI et al., 2014; SUN et al., 2014) in which variations in the placement policies led to different choices of replacement schemes. Table 3.8 summarizes the reported underlying system parameters. Those are the context characteristics that supported the analyses.

- Chen et al. (2016) develop an ICN-Wireless Sensor Network (WSN) system in which they tested 16 combinations between four placement strategies - LCE, Prob (i.e., LCP), LCD, and Betw (i.e., LCB) - and four replacement policies - Random, FIFO, LFU, and LRU - in a WSN with 15 nodes. The results reveal a significant variation in the rank of policies for different combinations of placement policies and comparison metrics. Considering the metric cache hit rate, LCE and Prob achieved their best results combined with LFU, while LCD and Betw with Random; Yet, when considering the metric energy consumption, LCE and Prob achieved their best results with FIFO, while LCD with LRU, and LCB with Random.

- In addition to analyzing the effect of heterogeneous policies configuration by node locations, Tarnoi et al. (2014) also analyzed variations on the replacement scheme choice according to the different placement policies. The work shows how the probabilistic caching placement behavior varies as a function of the replacement scheme. The authors evaluated combinations of LRU, LFU, and Random policies with LCE and Prob. In general, for both cascade and Internet-like topologies, and considering both server load and round trip-hop distance evaluation metrics, the results show that Prob can improve the performance of the network and achieve its best performance only when combined with LRU, while LCE achieves its best performance when in conjunction with LFU.

- Finally, as we mentioned earlier, the results reported by Sun et al. (2014) show that the optimal choice of the replacement scheme depends on the cache size and the placement policy. Regarding the variations of placement policies, the work combined seven placement policies - LCE, LCD, Rand, Prob, ProbCache, Betw (i.e., Centrality), and CRCache (i.e., Cross) - with five replacement policies - LRU, LFU, FIFO, TTL, and Size - and the results presented evidence of the difference in performance ranks for each combination. For example, considering the metric server load reduction and 1G of cache size, LCE, Rand, Prob, and ProbCache achieved its highest values when combined with TTL; while LCD with FIFO; Betw with LRU; and CRCache with TTL or LRU. However, for cache sizes of 100G and 1T, all placement policies presented their best results with LFU, except for LCD, which achieved the best results combined with LRU or TTL. The work also stands for a dominant strategy among the compared ones in terms of caching metrics. Partially in line with Chen et al. (2016), and contrary to the analysis presented by Tarnoi et al. (2014), the authors place Prob+LFU as the closest to the best strategy for their context. However, the analysis between the different results is limited because the two works (CHEN et al., 2016; SUN et al., 2014) did not mention the probability value used for caching contents. The Prob performance may vary according to the configured probability value.

Reinforcing the intrinsic correlation property between content placement and replacement decisions, all the works presented in this section show evidence of the different and unique effects of each policy's combinations for distinct network contexts. Different placement policies can have a different impact when changing a replacement scheme (REZAZAD; TAY, 2015; TARNOI et al., 2015; ZHANG et al., 2017; MEDDEB et al., 2017). This way, each placement strategy requires evaluation of what replacement scheme performs the better. Each placement policy has a different requirement in terms of evictions, and the more is the number of evictions, the more the placement policy relies on the replacement scheme and, therefore, is affected accordingly.

3.3.4.4 Content Popularity One of the behaviors we were expecting to find evidence for was the impact of content popularity variation on the replacement policy choice, especially on the choice between frequency-based policies, e.g., LFU, and others, such as recency-based policies. That reasoning relies on the argument of many works that frequency-based policies suit better content populations with high popularity skewness, while with low popularity skewness would suit other policies (BECK et al., 2017).

However, while analyzing the variations of popularity skewness during the comparative evaluation of the replacement schemes, we found works in which *popularity skewness variations did not influence policies' rank* (WANG et al., 2011; GÜR, 2015; HUANG et al., 2017; ZHANG; TAN; LI, 2018; AN; LUO, 2018; JEON; LEE; SONG, 2013; SHAILENDRA et al., 2016; LIU et al., 2017; TARNOI et al., 2014; GALLO et al., 2014; YOKOTA et al., 2016; ZHANG; TAN; LI, 2017; SINKY et al., 2018; YAO et al., 2016). Those comprehend works under Zipf popularity distribution, with different variations of the skew factor from, for example, 0 to 2, with conventional policies like LRU and LFU as well as more recently proposed policies, but the performance rank among the policies remained unchanged. Variations in the skew factor represent variations in the distribution of contents' popularity. The increase in the factor leads to an increase in the number of popular content. It is also associated with the diversity of contents distributed in a network of caches. The increase in the number of popular contents reduces the diversity of the contents stored in the caches since popular contents are more conducive to be accessed and occupy cache spaces for relatively long times.

Also, we observed a similar effect as the one about the increasing of cache size discussed earlier: under variations of the skew factor solely, *as the skew factor increases, the difference of performance among the techniques decreases* (BADOV et al., 2014; YOKOTA et al., 2016; ZHANG; TAN; LI, 2017; ZHANG; TAN; LI, 2018; SINKY et al., 2018; YAO et al., 2016).

3.4 POLICY CATEGORIES FOR ICN APPLICATION AREAS

This subsection presents potential context characteristics to enhance the eviction performance in emergent networks. We correlated characteristics of emergent networks with the context characteristics relevant to the choice of suitable cache replacement schemes. In the following paragraphs, we highlight the most suitable context characteristics for generic network contexts on information-centric IoT (ARSHAD et al., 2018; DONG; WANG, 2016), vehicular named-data networking (KHELIFI et al., 2020), and ICN-enable edge and core networks (ZHOU et al., 2017; ZHANG et al., 2018). Table 3.9 summarizes the discussion.

Cache-enable network	Characteristics and/or requirements	Policy category	Correlation of requirements with context dimensions
IoT (Smart home, home care...)	High heterogeneity among IoT devices with different priorities; High ephemerality of contents; Limited resources.	Content and Node-based	Content features, like content provider identification, priority, and time-related properties
VANETs	High intermittency of connections; Multi-path propagation; Different strategies for delay-sensitive data from safety applications and delay-tolerant data from infotainment applications.	Content and node-based	Node location properties like mobility pattern plus direction, node's rank according to topology position; Content features, like type, priority, and popularity and time-related properties.
Edge computing (Small-cells radio access; 5G; Device-to-Device (D2D) communication; UAVs)	High temporal and spatial correlation of content requests; Enables clusters by user similarities.	Content and future human-based	Content popularity properties; User preferences, habits, and social interaction.
Internet-scale networks	Globally content preferences; Heterogeneous link/node capacities; Long geographical distances.	Content, node and network-based	Content feature and popularity properties; Network topology, resource, and time-related properties; Node resource and traffic properties.

Table 3.9 Suggestion of cache replacement policy category for different ICN-enable scenarios.

Information-centric Internet of Things

The suitable kind of cache replacement schemes for information-centric IoTs should deal with the two most significant characteristics of IoT traffic: i) the large number of heterogeneous devices and ii) the ephemerality of the content produced by them. In the former, the different types of devices usually have different resources restrictions in terms of processing capabilities, memory, energy constraints, and they produce contents with different requirements regarding the context. For example, Smart Cities will need to integrate intelligent urban sensing services for many purposes, such as management of smart garbage collection, street lighting, parking, the monitoring of road conditions, urban noise, security cameras, and environmental conditions, among other possibilities. In this case, the infrastructure comprises a diversity of sensors with different content production rates and characteristics. The replacement scheme may apply different treatment to the contents according to the type of device by exploring both *content* and *node context dimensions*, with features like *content provider identification*, *content priority*, and *node resource features*.

The latter characteristic points out the typical time-restricted data generated by some IoT devices that periodically inform sensor measurements monitoring the environment. For example, the content periodically generated by temperature sensors and collected by distributed applications to monitor the ambient in urban areas can be usefully cached to serve user applications' requests. However, the most recent measure will usually be of interest to most applications, and there is no need to maintain the previous measures in the cache. The replacement scheme should also combine *time-related features* of the *content context dimension* in the eviction process logic. The combinations of the features mentioned above can help detect redundant contents from the same producer while increasing the techniques for stale content detection.

Vehicular Named-Data Networking

For vehicular-NDNs, the cache replacement scheme should consider the singular temporal and spatial characteristics of vehicular traffic because they can affect the local relevance of contents. For example, accident information's relevance is highly dependent on the vehicle location and the direction towards it was moving (de Sousa; Araújo; Sampaio, 2018). If the vehicle has passed the accident, that information may no longer be useful. The replacement schemes can handle this decision with *node location properties* like *mobility pattern*, plus *vehicle direction*, and *node's rank according to the current topology position*.

Moreover, and combined with node location properties, different strategies should be applied to deal with the different types of applications running in vehicular networks. For that, the strategy can explore *content features*, like *content type* and *content priority*. The road-traffic-related applications, such as road congestion notification, traffic monitoring, and accident warning, usually are delay-sensitive applications and are better handled by *content time-related properties* or even newly *type-specific properties*. Similarly, applications for code dissemination designed to support smart city infrastructures' upgrades can benefit from those properties. Meanwhile, the infotainment applications are mostly delay-tolerant and more suitable to be handled by *content popularity* features.

In-network Cache-based Data Offloading through Edge Computing

A fundamental characteristic created by the user's closeness in edge networks is the temporal and spatial correlation of content requests. In this way, one of the widely explored approaches at the network edge is user-centric clustering techniques (Ribeiro; Sampaio; Ziviani, 2018; HE; WANG; WANG, 2019; ELBAMBY et al., 2014). User characteristics are the input and motivation for virtual groupings, whether regarding the network structure or the users' connection to the network. As a consequence, user and their content requests can be grouped according to user behavior patterns.

Thus, the replacement schemes for in-network caching at the edge can benefit from *content-based properties*, especially *content popularity* features, and the exploration of a variety of *human properties* related to preferences, habits, and social interaction. Therefore, user behavior analysis is a relevant area in the future of edge-caching, fostering future human-based replacement policies.

ICN-enabled Core Network

Because of the considerable physical distances naturally presented in large-scale networks to connect content consumers and producers, requests typically have to traverse several nodes within the network. Therefore, the *network topology* context must be taken into account to optimize cache replacement policies in content-based core nodes. Context properties, in this case, are related to the *distance* connecting two end-nodes, like *hop count*, properties related to the *network resources*, like *packet transmission cost, link capacity*, and *time-related features* with *network delay for retrieving content*.

The cache replacement schemes should also explore *content* and *node* contexts to reflect globally content preferences and the different capacities of core nodes, respectively. The *content feature* and *popularity properties* and *node resource* and *traffic properties* may further increment the replacement policies' decision. On the other hand, there is a trade-off relating the performance while processing many context information, since core routers process requests at line speed.

3.5 FURTHER DISCUSSION ON LESSONS LEARNED

The SLR covered many works with evaluations of cache replacement policies that presented different behaviors according to variations in contexts. Contextual factors are triggering this difference in performance, and the SLR was able to identify some common factors in a set of works, as we exposed in the previous subsections. One important lesson of the overall review is the confirmation by many evidence that efficient utilization of cache resources relies on employing cache replacement policies according to some context.

The influence of some contextual factors was already evident when looking at individual works. However, one of our intentions with this SLR was to analyze the works that had similar effects, to look for patterns that could relate the contextual factors to the policy's properties. That cames in contrast with the diversity of context characteristics and evaluated policies, which limited the analysis.

Besides, there was no more in-depth analysis of why and how the effects happened, most of the works came to evidence by testing the context variations, and small changes in any context characteristic could have led to different results. In general, there was no explicit pattern in the surveyed works associating the context factor to the policies or their properties. That also limited a more in-depth analysis from the perspective of the proportion of impacts for different contexts, since the extent to which context characteristics affected cache replacement strategies varied for the different works.

We must also highlight that most of the works did not indicate the confidence interval in their experiments. A few of the differences between policies' performance measurements were relatively small, and a confidence interval would help investigate the significance of the difference values.

Due to the reasons mentioned above, the policy choosing process can not be reduced to rule-based schemes or related solutions. Instead, the choosing process is suitable for solutions that dynamically analyze context factors and perform large-scale correlations between the factors and policies, for example, with reinforcement learning techniques.

3.6 CHAPTER SUMMARY

This chapter detailed a comprehensive and systematic review of studies regarding cache replacement policies in ICNs. The literature presents a vast set of eviction strategies exploiting combinations of multi-dimension aspects of context information in different ways, aiming at making more customized and effective decisions about the relevance of contents. Among its contents, the chapter presents:

- The relevance of considering context's properties in choosing suitable replacement policies.
- The characterization the context factors correlated with the caching policies and the reported effect of context variations on cache replacement policies' performance.
- Features of the content, node, network and human dimensions that can be explored by the replacement schemes to evaluate the relevance of contents.
- A catalog of caching replacement policies classified according to the context information used by the policies.
- Evidence that the cache performance associated with replacement policies is influenced by features like cache size, location of the cache node into the topology, content popularity, and content location policy.
- Evidence to confirm the absence of a single context factor determining the choice of policies; there is no explicit pattern regarding context properties variations to support the choosing process of policies for different network contexts.
- An analysis of context features to be explored in emergent network scenarios such as Information-centric IoTs and Vehicular Named-Data Networking.

Chapter 4

THE INFLUENCE OF HUMAN HABITS ON CACHE REPLACEMENT POLICIES

This chapter presents a case study experiment to investigate the influence of user habits on the performance of cache replacement policies. We aim to consolidate by evidence the human dimension as a context aspect related to the caching policies. In the following sections, we first describe an outline of our case study (Sec. 4.1). Next, we detail the methods and protocols used to perform our experiment (Sec. 4.2). Then, we present the experiment results and an analysis of our main findings (Sec. 4.3). Last, we summarize the chapter (Sec. 4.4).

4.1 EXPLORATORY CASE STUDY OUTLINE

We carried out an exploratory case study based on the hypothesis that the most predominant profile among users in cluster-based networking approaches (MAO et al., 2017; KLAINE et al., 2017) may lead to different performances for different caching policies. Thus, each cluster can work with a best-fit customized policy according to a predominant user habit.

To confirm our hypothesis, we investigated habits of users on listening to music. We defined user profiles according to a group of users that share the same habits when using a music streaming service. The characterization of habits considers frequency of music requests. Then, the resulted profiles guided the building of the clusters, which were extracted from historical records from an on-line music streaming service. Methods of a data mining process assisted the construction of the clusters. We finally evaluated the performance of common used replacement policies under the actual requests of each cluster. The evaluation considered simulated NDN scenarios.

In addition to the cache performance analysis, we also investigated the correlation between user's profile and the songs popularities. For that, we explored the popularity distribution of the songs from the perspective of Benford's Law (PIETRONERO et al., 2001). Benford's law is a distribution of probability P observed empirically in numerical data of several natural processes. This research reveals that the popularity distribution of the songs follows an approximation of Benford's Law, and it is possible to differentiate profiles for different users according to the behavior of Benford curve for the accessed songs.

Hence, the case study contribution is threefold. It (i) identifies user profile relations with cache replacement policies based on evaluations with real datasets; (ii) ratifies the inclusion of human context dimension as a context factor that influences the choice of cache replacement policies; and (ii) substantiates the building of user profiles predictor systems by presenting a correlation model between user profiles and content popularity patterns.

4.2 CASE STUDY METHODOLOGY

The case study aims to analyzes how cache replacement policies perform according to different user profiles. To this end, we used a real dataset extracted from Last.FM[1], an online music stream service that stores traces of different users around the world. Figure 4.1 describes the whole process.

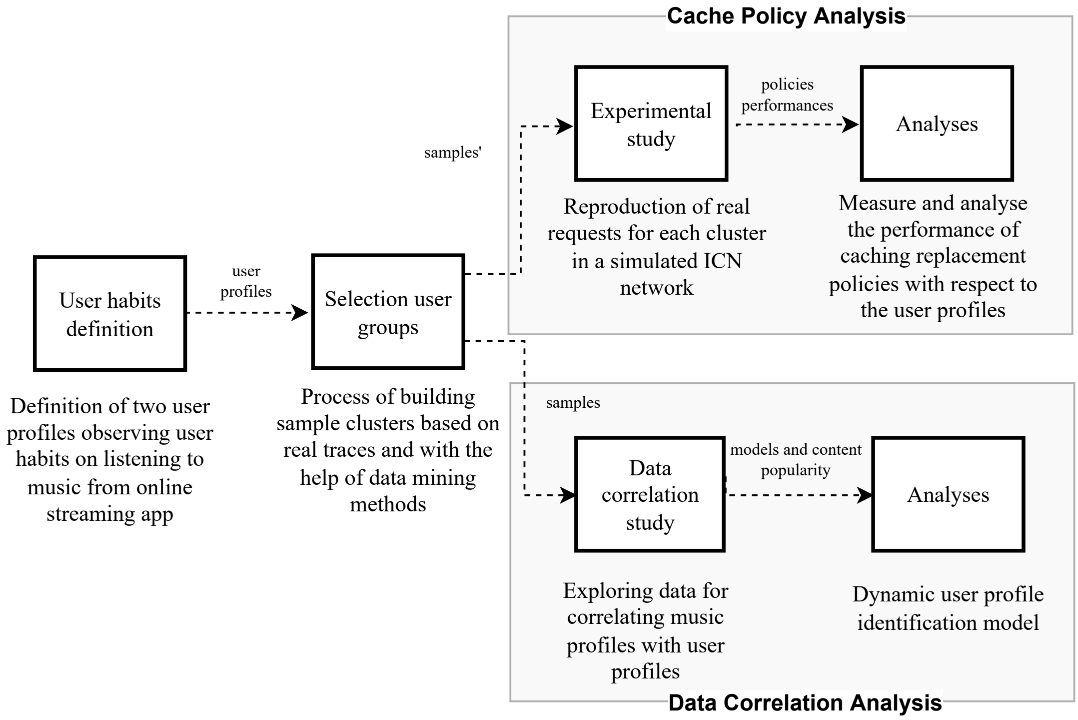


Figure 4.1 Steps of the whole evaluation approach adopted for the experiments.

1. We defined user classes based on song repetition habits empirically observed in a historical dataset from `Last.FM` (step 1)

[1]http://www.last.fm

2. We use data mining processes to select data samples that represent song requests from each class (step 2);

3. With the selected samples, we divided the process into two parallel branches:

   - The first uses part of the samples to reproduce an experimental cache replacement policy evaluation study;
   - The second branch uses the samples for a correlation analysis of user information with the popularity of the songs accessed.

The process steps are detailed in the following subsections.

4.2.1 User Habits Definition

In order to identify different characteristics of users, our study relied on the definition of two user profiles (P) based on their listening habits to build clusters (C). We mapped the users habits according to the level of repeatability of accessed music and defined two profiles as follows:

- P_1: *users that frequently request the same songs:* with this profile we infer habits of methodical and systematic people, used to listen to favorite playlists;

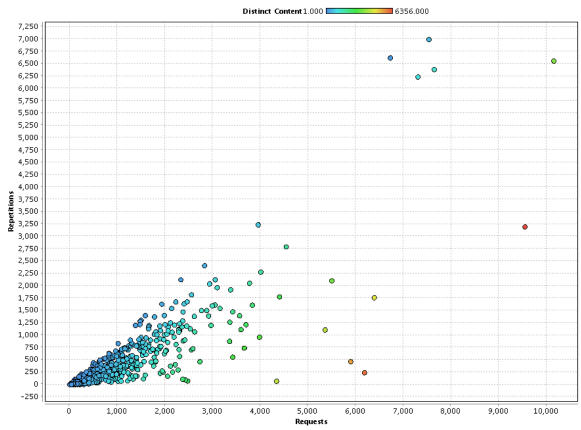
- P_2: *users that do not usually repeat the requested songs:* deducing habits of more dynamic and impulsive people, who almost never repeat their songs

Accordingly, the experiments rely on two user clusters (i.e., C_1 and C_2) built through the aforementioned profiles, P_1 and P_2, respectively. So, $P \longrightarrow C$. Additionally, we created a third cluster (i.e., C_3) composed by random users with mixed behaviors, that served as a baseline for comparison purposes.

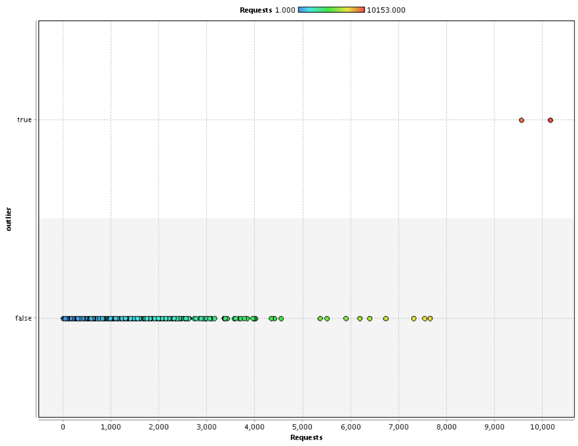
4.2.2 Selecting User Groups

We carried out the cluster selection step by following a data mining process that identifies groups of similar users existing in the Last.FM dataset, according to each profile. The dataset contains all music requested by users in a period of approximately four years with more than 19 million usage records. Each data record describes the song, data and time of request, artist, and user who requested the song. We manipulated this data to generate three clusters that served as inputs to the simulations. Two sets (i.e., C_1 and C_2) consisted of users that matched with definition of profiles P_1 and P_2, respectively. The whole procedure relied on pre-processing phase of data mining methods and was carried out through the following steps:

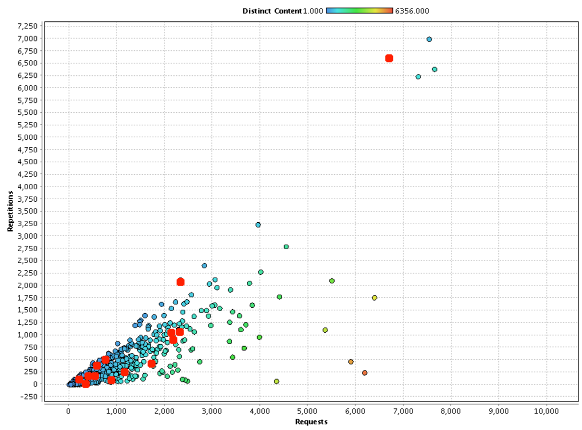
- *Pre-processing - Sample selection by time:* we filtered 10 random interval samples in periods of a month, totalizing a period of 10 months. Each month has an average of 594,483 requests, 191.703 distinct songs, and 622 distinct users;



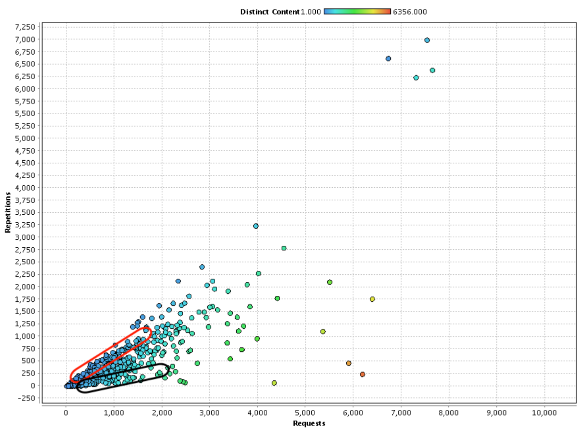
(a) Sample of the total requests of all users in the dataset in a period of a month



(b) Sample of pre-processing result for identification of outliers based on the number of requests



(c) Sample of random selection of 20 users without distinction of habits



(d) Samples of two clusters area of distinct users profiles

Figure 4.2 Building user's clusters: dataset selection sample.

- *Extract listening habits:* For each sample interval, we extract datasets to map the habits of users according to the total of music repetitions made during the month by each user. Figure 4.2(a) illustrates a sample in a period of a month. Each dot in a board is a distinct user, where x-axis represents the total amount of music requests (per user), and y-axis defines the total repetitions of music a user made in the same period. The total of user's repetitions is defined as follows: Let S_u be the set of n distinct songs accessed by user u, with $(s_1...s_n) \in S_u$. Yet, let $Q_u(s_i)$ be the total amount of access to song s_i by user u. The total repetitions R for a user u is defined as $R_u = \sum_{i=1}^{n} Q_u(s_i)$, $\forall s_i \ Q_u(s_i) > 1$;

- *Data cleaning - Identification and removal of outliers:* To mitigate biases on randomness of the selection of users, we analyzed all users of the selected samples in regard to the total of requests, to cut off the ones considered as *outliers* for the corresponding sample. We have applied a data pre-processing algorithm (RAMASWAMY; RASTOGI; SHIM, 2000) that detects outliers based on the Euclidean distance from each record to its k-*th* nearest neighbor. Since it is hard to define the threshold that qualifies a user as an *outlier* due to the subjectivity of the context, and we aimed to do minimal interferences in the dataset, we configured the Euclidian distance parameters with minimal values. Figure 4.2(b) shows the result of a sample dataset classified with number of outliers = 2.

- *Data groups selection*: After filtering the outliers, we proceed a random selection of the users. For each sample interval, we have selected three sets of 20 random users. The first set is a general set, without distinction of listening habits, to be used as a negative control on performance evaluation. Figure 4.2(c) shows an example of one selection of user clusters we carried out. The other two sets are meant to reflect distinct habits, related to different user profiles. The former represents the profile P_1 (the highlighted top area of the Figure 4.2(d)), and the other group represents the profile P_2 (the highlighted bottom area of the Figure 4.2(d)).

4.2.3 Cache Policy Analysis

After building sets, we reproduced all user's songs requests through simulations conducted during the experimental study. We evaluated the cache replacement policies through a simulation-based experimental study conducted over ndnSIM, an ns3-based NDN simulator (AFANASYEV et al., 2012). The study consisted in simulating user requests according to the clusters created, in order to evaluate the influence of their listening habits on the performance of each evaluated cache replacement policy. Thus, this section presents the details of the experimental study, including the simulation environment and the evaluated metrics.

4.2.3.1 Simulation Environment Through the ndnSIM simulator, we built a simulation environment which enabled us to reproduce a caching networking architecture. Each record of a song requested in the dataset corresponds to an *Interest* packet sent to network. Besides, the simulation reproduced the sequence and time of requests exactly as recorded in timestamps of the sample data.

Figure 4.3 depicts the ICN scenario through which the simulation took place. It consists of one router with cache capacity to intermediate requests and one music server (i.e., the producer of contents). The simulation performs user requests of each cluster at once and according to the recorded timestamps. Each experiment runs one individual cluster at time. Since the communication process follows an ICN architecture, the content is initially searched in the cache router. In case of it being found, a *cache-hit* is counted and the content is immediately returned to the user. Otherwise, i.e. the content is not founded in the cache, the request is forwarded to the server.

4.2.3.2 Performance Metrics and Simulation Parameters We choose the cache hit ratio as performance metric to evaluate the replacement policies under analysis, that

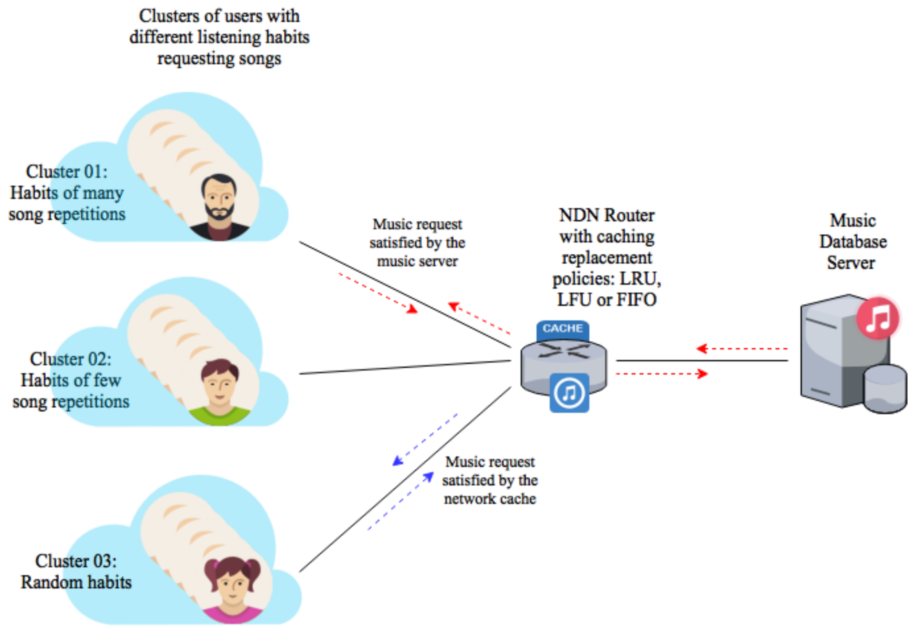


Figure 4.3 ICN scenario reproduced by simulations.

is: the capacity of cache solving the requests locally, instead of downloading contents from the server, thereby saving network resources. This metric was measured through Equation 4.1, where *totalHit* is the number of music requests satisfied by the network cache, and *totalRequests* represents the number of requests received by the cache. As we extracted 1-month interval subsets (total of 10 different months), we measured the cache-hit for each sample, and the results correspond to an average of the measures of each sample separately.

$$HitRatio = \frac{totalHit}{(totalRequests)} \times 100 \tag{4.1}$$

Table 4.1 summarizes simulation parameters adopted in experiments.

4.2.4 Data Correlation Analysis

The listening habits modeled herein are intuitively related to the content's popularity. Therefore, we examined the popularity distribution of the songs in the dataset by analyzing one-month samples previously extracted. For each song in the sample, we computed its popularity by counting all the requests of all existing users in the dataset.

By analyzing the relationship among the data, we notice that the popularity distribution of the songs follows an approximation of the Benford's law. The law states that given a numerical set on a decimal basis, the proportion of the first most significant digit d of any number is approximately equal to the probability function:

[2]LFU *with Dynamic Aging* (ARLITT et al., 2000).

Table 4.1 Simulation parameters.

Parameter	**Value**
Quantity of users (by cluster)	20
Quantity of routers	1
Mean of distinct contents (cluster 1)	3221
Mean of distinct contents (cluster 2)	10125
Mean of distinct contents (cluster 3)	7059
Caching Policy	FIFO, LRU, LFU e LFU-DA[2]
Caching Size	5%, 15% and 30% of distinct content
Data rate	1Mbps
Delay	10ms
Simulation time	accordingly to the trace

$$\begin{aligned} P(d) &= \log_{10}(d+1) - \log_{10}(d) \\ &= \log_{10}\left(\frac{d+1}{d}\right) \\ &= \log_{10}\left(1+\frac{1}{d}\right) \end{aligned} \tag{4.2}$$

where $\forall d \in \{1, 2, 3, ..., 9\}$. In this way, we calculate the proportion of digits d in a base containing the numerical value of the popularity of all songs in the sample.

Benford's law can be applied in several areas of knowledge for different purposes. For instance, in computer science, it has been used for network anomalies detection (ARSHADI; JAHANGIR, 2014).

The investigation led us to evidence of a correlation between users' profile and song popularity, and to a popularity model that identifies users' profile. The results of the correlation analysis are detailed in Section 4.3.2.

4.3 CASE STUDY RESULTS

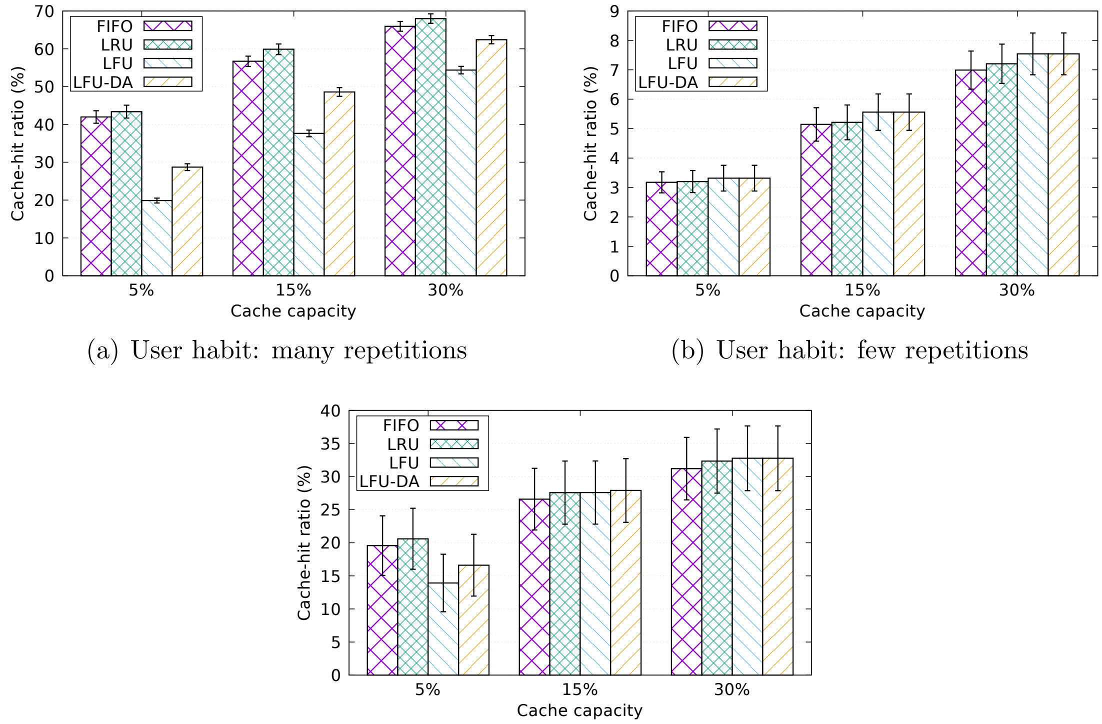
4.3.1 Cache Policy Results

The results obtained through the experimental study allow us to evaluate the influence of the clusters on the performance of the caching policies. They describe the cache hit ratio obtained when using LRU, LFU, LFU-DA, and FIFO replacement policies, varying cache size from 5%, to 15% and to 30%. In addition to hit ratio analysis, we also discuss the cache storage consumption. Such results show different performance outcomes for each user profile as we discuss in the following subsections.

4.3.1.1 Impact of User Habits Figure 4.4 presents the results from the experiments that supported our analysis in respect to the listening habits of users. Figure

4.4(a) describes the hit ratio obtained when the cluster is formed by users with profile P_1 (i.e., users that frequently request the same songs). As it shows, LRU presents better performance than the other policies when users frequently repeat song's requests. Conversely, for the other network scenarios, this advantage does not exist. As Figures 4.4(b) and 4.4(c) show, despite of a slight improvement on the LFU's hit ratio mean, when it is compared to the other policies, they are statistically equal, if we consider the confidence interval. We remark that the random-habits cluster presented an even higher confidence interval. Certainly, this is due to the group of users that present random listening habits that lead to a big variation in the hit ratio.

Such results reveal that the user profile can influence the performance of cache replacement policies. The hit ratio varies from both profiles P_1 and P_2, and the whole set of policies has changed their performance only by varying user profile, irrespectively the difference among them. So, the user habits should be considered when choosing a replacement policy in order to achieve optimal performance. Figure 4.5 presents the same results of cache hit ratio but from another perspective of analysis. As it shows, irrespectively the cache capacity, the advantage of clustering users that frequently request the same songs is always bigger when compared to the non-cluster approach (baseline group), whereas clustering users with few song repetitions is worst. This discussion leads to another finding: clustering users with similar preferences is the best option to obtain higher hit ratios, in particular for groups of users that frequently request the same songs.



(a) User habit: many repetitions

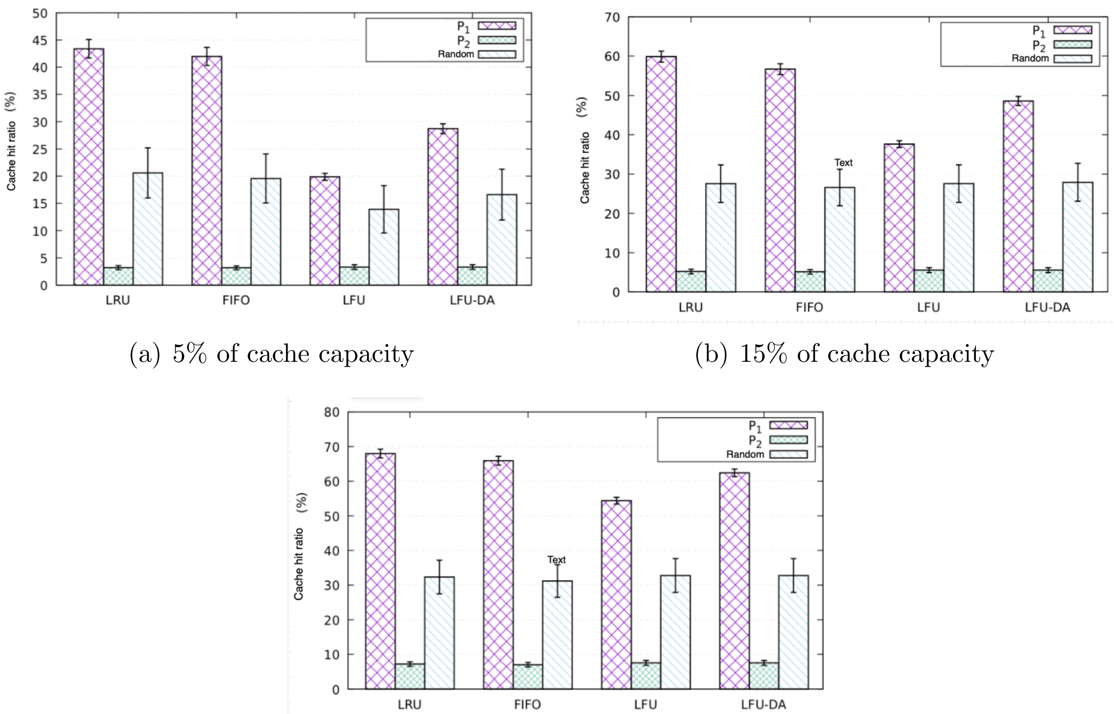
(b) User habit: few repetitions

(c) Baseline trace: Random Users

Figure 4.4 Cache hit ratio for each cluster of users.

4.3.1.2 Impact of Cache Size In addition, the experiment contributes to the cost-benefit analysis of cache size. The results show that the increase in cache capacity is not proportional to the increment in the hit rate. This way, just adding cache resource on the network could not be the most suitable solution to improve the performance.

Furthermore, the results reveal that the choice of replacement policy does not rely on cache capacity, but on user profile. That is, the best replacement policy may be different for different profiles, but for most cases, remains the same for different cache sizes.



(a) 5% of cache capacity

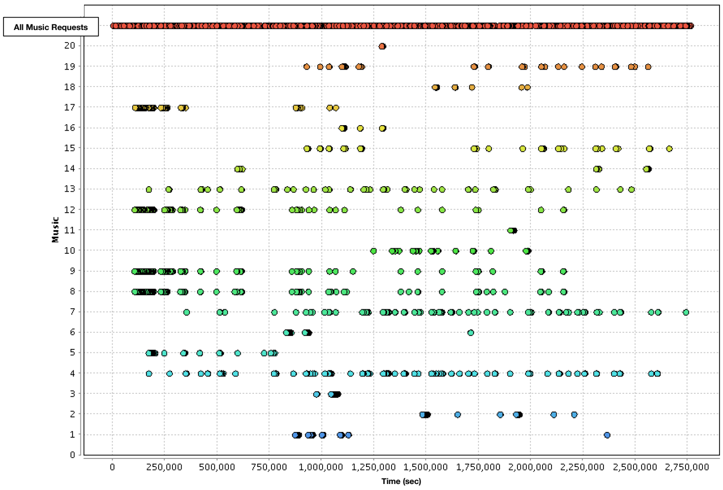
(b) 15% of cache capacity

(c) 30% of cache capacity

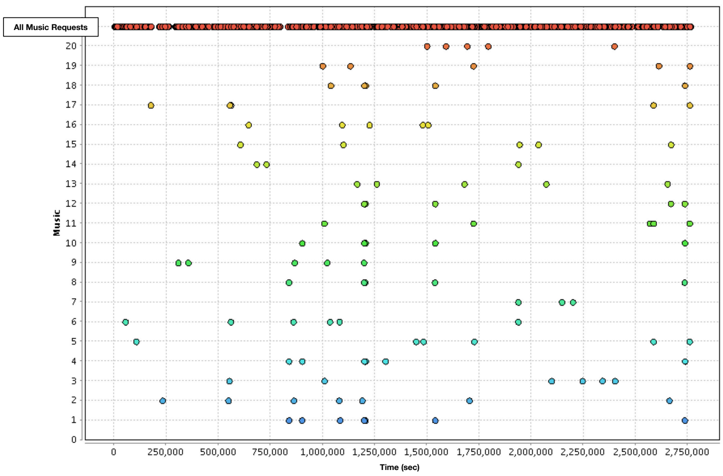
Figure 4.5 Cache hit ratio for each cache size.

4.3.1.3 Cache Behavior Analysis Finally, in order to further investigate cache behavior, Figures 4.6(a) and 4.6(b) depict sample sets of song requests from profiles P_1 and P_2 respectively. They show the 20 most requested songs in a period of a month, and each dot represent a song request, i.e., each horizontal line shows the time one music was repeatedly requested by any user in the trace. The upper line contains all song requests in the trace to be used as baseline.

By analyzing Figure 4.6(a) we can see the high frequency of song repetitions, but the interval between the repetitions is too short in most cases. For instance, *song 1* was requested 173 times in a short interval, while *song 2* was requested 165 times, and *song 3* had 157 requests. Since we are evaluating a group composed by users that have a lot of repetitions, it may be expected that LFU would have the best results, because data are requested more frequently. However, in that application, the time between the requests is more important than the frequency of requests. Thus, the two policies that consider time of request have better results: FIFO and LRU.



(a) Music requests from profile P1

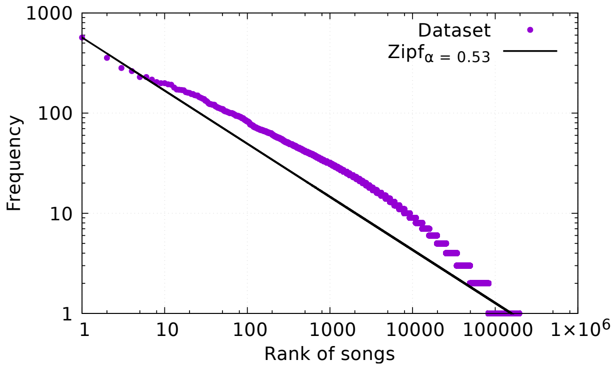


(b) Music requests from profile P2

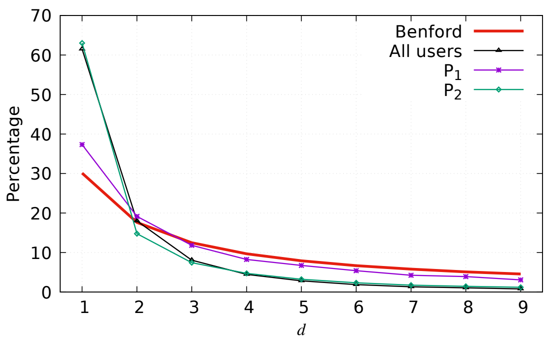
Figure 4.6 Cache behavior analysis: sample of the most repeated songs.

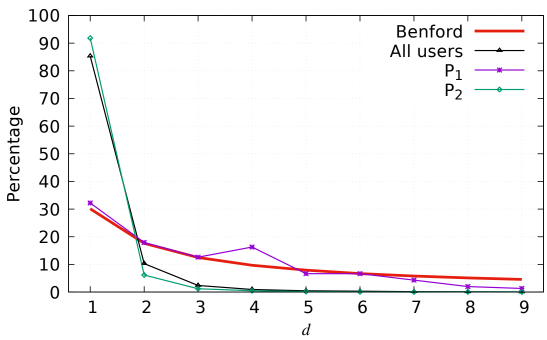
4.3.2 Data Correlation Results and Dynamic Profile Identification

Figure 4.7(a) illustrates the popularity of the songs in a sample for a period of one month. It shows a log-log scale of the number of requests for each song versus the ranking of the songs. It is possible to note that the first i-ths in the rank are the most popular songs. Besides that, the plot demonstrates the accordance of the dataset with the Zipf-like distribution Ω/i^{α} with $\alpha = 0.53$. It shows a smoothing on the difference of the popularity among songs in the ranking when compared to the plain Zipf (i.e., $\alpha = 1$), through which the second most popular content has exactly half of the popularity of the first one in rank, and so on.



(a) One-month sample that shows a Zipf-like distribution with the number of requests for each song versus the ranking of the songs.





(b) One-month sample containing the distribution of the first most significant digit of songs popularity.

(c) One-day sample containing the distribution of the first most significant digit of songs popularity.

Figure 4.7 Popularity distribution of songs in a sample set that follows the Zipf distribution and the Benford's law.

Figures 4.7(b) and 4.7(c) depict the distribution of digits that follows the Benford's law, as well as the popularity distribution of all songs accessed by all users existing in a sample of one-month and one-day periods. The x-axis refers to the first digit of the total request value of a song by all users (song popularity). It is possible to observe that the popularity of songs follows a derivation of Benford's law, with significant change for the songs with popularity whose first digit is equal to 1 (in the majority are the songs of low popularity).

To investigate the existence of a relationship between user habits and the popularity of his/her accessed songs, we analyzed how songs are accessed by user profile. Figures 4.7(b) and 4.7(c) show that the profiles are similar for most "high-popularity" content, but differ in access proportion to most "low-popularity" content. By evaluating the behavior of the curves, we can differentiate user habits from its access proportion to content with "low popularity". For example, deriving the following rules: (i) Users with profile P_1 access less than 40% of contents with popularities whose first significant digit is 1; (ii) Users with profile P_2 access more than 60% of contents with popularities whose first significant digit is 1.

Hence, the analysis shows that it is possible to infer the profile of the user by observing only the global popularity of the content that he accesses, without the need of counting the repetition level of each individual. That is, it enables the online identification of user profiles without higher computational costs. Based on this correlation, we propose to use the residual sum of squares (RSS) as a metric to define user profile. In statistics, RSS is the sum of the squares of residuals (i.e., deviations predicted from actual empirical values of data). It is a measure of the discrepancy between the data and an estimation model. A small RSS indicates a tight fit of the model to the data. In this analysis, RSS metric evaluates the distance between the user profile and Benford model through Equation 4.3:

$$RSS = \sum_{d=1}^{9} (y_d - \hat{y}_d)^2 \tag{4.3}$$

where $\hat{y}_d$ is the expected proportion for music with popularity on first digit d, following Benford's law, and y_d is the measured one.

Figure 4.8 contains the RSS values measured by user profile. Users from profile P_1 (many repetitions) have significantly different RSS values from profile P_2 (few repetitions). The RSS difference between the profiles is resulted from the difference of popularity distribution of the songs that each profile usually accesses, and therefore can be calculated by means of the RSS. Its value allows determining the profile of the user by observing only the proportion of the popularity of the content that a user requests. That is, based on RSS, it is possible to build a Benford's law based predictor capable of inferring user profile only through the information about his/her playlist.

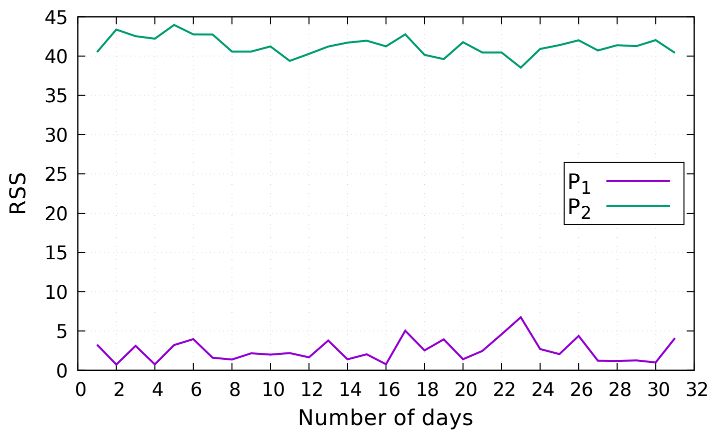


Figure 4.8 RSS for the modeled profiles computed through Equation 4.3 by day measurements.

4.4 CHAPTER SUMMARY

We have presented an experimental case-study which shows the potential gains in adopting cache replacement policies according to the most predominant profile of users they serve. The case-study contributed with an investigation of the concept of human-aware information-centered networks, and presented an analysis of the habits of users of online music streaming applications. The study showed that the performance of cache replacement policies can be optimized when the policy is chosen according to user habit. We can improve the hit ratio by around 30% when a cache policy is used according to the profile of the users. Therefore, user habits should not be overlooked when adopting a cache policy for maintaining contents in content-routers close to the user.

Additionally, the analysis of user's habits on using online music streaming application reveals that the popularity distribution of the songs follows an approximation of Benford's law. The case study reveals the suitability of the proposed model for an online predictor, capable of classifying the user according to listening habit and, therefore, guiding the design of human-aware caching strategies in many networking settings.

Together with the literature review presented in Chapter 3, this chapter added evidence of context aspects related to caching policies. In the following chapter, we present a new context-aware method to choose suitable caching replacement policies online.

Chapter 5

A CACHING META-POLICY STRATEGY

Sets of context information have been raised from multi-dimension aspects that can directly influence the behavior of cache eviction policies. We have seen that there is a vast set of strategies exploiting combinations of those information in different ways, aiming to make more customized and effective decisions about the relevance of contents. However, we still don't know how to choose a suitable policy regarding a given scenario. As we stand previously, there is a lack of research efforts to assist the process of gathering which caching policy should be deployed in a given network scenario.

Traditionally, the choice of caching policies requires some empirical knowledge associating context characteristics of the cache production environment with the policy type that should be adopted. For example, some works point to recency-based policies as more suitable for traffics with strong temporal correlation, such as video streaming (EUM et al., 2015); or frequency-based policies for more stationary traffic patterns (DRäXLER; KARL, 2012); other works point to random policies as an efficient choice to handle filtered traffic from a hierarchical caching structure (GALLO et al., 2012). However, it is not always possible to previously know the characteristics of the workload scenario.

Besides, traffic characteristics can possibly change over time, leading to changes in which caching policy would better fit the cache. Changes in the number of users, predominant user habits, and applications type (among other contexts) can dynamically change the characteristics of the network at different times. Also, user mobility imposes a particular challenge leading to on-demand changes in the network topology, node connectivity, and workload aspects. Variations in context aspects may affect policies' performance over time, and the cache must adapt to those changes to ensure optimal performance.

As described in the Introduction section (1.2), the research question of this work that addresses the issues mentioned above is: *How to explore the cache replacement policies and instantiate best-fitting strategies dynamically adapting to on-demand changes, considering the available context characteristics of the overall scenario.*

Toward answer this question, we take advantage of the diversity of policies presented in the literature and model the caching policy choice problem as an online learning problem.

We assume that the cache has multiple cache replacement policies potentially suitable and operable by the cache. Thus, we formalize the decision-making process of choosing the most suitable policy from a finite set of possible policies.

In the following subsection, we introduce online learning concepts for caching systems (Sec. 5.1). Then, we present the system model used by our strategy to assist the choosing process of cache replacement schemes (Sec. 5.2). Next, we describe the proposed meta-caching policy strategy (Sec. 5.3) and detail the main components in separate subsections. In addition, we discuss the generic aspects of the caching meta-policy strategy (Sec. 5.4). We conclude the chapter by presenting the chapter summary (Sec. 5.5).

5.1 ONLINE LEARNING FOR CACHING SYSTEMS

Online learning models represent a subset of machine learning techniques to tackle online prediction problems. An agent interacts with the environment in successive online rounds by taking one action at each round over a range of possibilities. Different actions cause different impacts on the environment measured by corresponding reward values. Each iteration is a new decision instance, and the problem is to predict which action to execute, aiming to maximize the cumulative rewards in a time horizon (or reduce the regret when not choosing the best action). The model of choosing over a finite set of actions is known as learning from expert advice, as each action plays as an expert by returning numerical feedback information used to improve future predictions. Two usual feedback models are the *full information*, in which the agent has access to the feedback of all actions, and the *partial information*, in which only the played action yields its feedback value. This partial feedback setting is also known as bandit feedback.

Bandit models are widely explored to solve resource allocation problems in caching systems (BLASCO; GÜNDÜZ, 2014; MÜLLER et al., 2016; ZHANG; REN; DU, 2017; CHEN et al., 2019; BITAGHSIR et al., 2019; DAI et al., 2019). In particular, the content allocation problem in cache-enabled cellular infrastructures can be modeled as bandit applications. For instance, Blasco e Gündüz (2014) proposed a bandit model for content placement in wireless small base stations (sBS). The model places an agent as the manager of an sBS and the contents present in a connected macro base station (mBS) as the actions. In summary, the bandit problem is to select a subset of popular contents to cache at the sBS by accounting for the popularity of the contents present in the sBS only. The agent has no information about the popularity of all contents placed at the mBS. This way, the strategy needs to choose the best subsets of popular content.

Moreover, the authors in Paschos et al. (2019) employ an online gradient ascent method as a caching policy to address the content allocation problem. Gradient ascent is a type of optimization algorithm also used to solve resource allocation problems. Following the content allocation problem's general goal, the strategy chooses a smaller subset of potential popular contents to store at the cache given a content set. Each arriving content request triggers the proposed algorithm to adapt the cached subset based only on the current cache composition and the most recent request.

Another study in Li et al. (2018) models cache content configurations as cache states and transition states as Markov. The work shapes caching policies as online distribution learning algorithms, in which each caching policy can be associated with a distinct popularity distribution of the cached contents. The authors also propose an adaptive policy directed to the learning process under non-stationary request models based on the study.

The employment of online learning techniques as a content replacement policy can be computationally costly since the cache has to trigger the learning agent at each content request arriving at the cache. Besides, the commonality of those and most proposed cache-related solutions is the focus on the content choice. Naturally, that is the central objective of the caching policies. We, instead, model the online caching problem with an upper abstraction level. We propose employing online learning to enhance the cache with the meta capacity to choose among an available set of potentially suitable policies. Our model overcomes the intensive iteration of learning agents as the agent needs to interact with the cache in predefined time intervals, instead of at each arriving content request. This setting is particularly conducive for in-network caching architectures with no restrictions on the content set.

In machine learning language, a sequence of actions chosen according to some learning algorithm is called *policy*. To avoid misunderstanding with the caching strategies, this work applies the term policy only to reference the caching replacement strategies.

5.2 SYSTEM MODEL

Consider a cache-enabled router CR with fixed capacity for n contents from a content library set L of unknown size, but it is known that $|L| \gg n$. The router responds to the content requests passing the network when the content is stored locally, thus counting a cache hit. Otherwise, the router forwards the request to another node on the network and counts a cache miss. Content packages passing through the cache can be opportunistically stored locally, but the cache space is always fully occupied in the steady-state. Therefore, the cache works with a cache replacement policy ω to keep the contents most likely to be reaccessed.

Given a discrete-time setting, we slotted the time into fixed intervals I. The cache efficiency of CR inside the interval I can be defined as:

$$CE(\omega)_I = \frac{H_I}{M_I + H_I} \tag{5.1}$$

in which H_I is the number of content interests satisfied by the cache in I and M_I is the number of missed requests in the same interval. The cache efficiency relies on the policy ω since different policies perform differently according to their eviction logic.

Consider a finite set $\Omega = (\omega_1, \omega_2, \omega_3, ..., \omega_m)$ of m content replacement policies feasible to the CR, for $m \in \mathbb{N}$. Without loss of generality, we assume that $CE(\omega_1)_I > CE(\omega_2)_I > CE(\omega_3)_I > ... > CE(\omega_m)_I$. Each $\omega \in \Omega$ can use a different set of contextual information for the eviction logic. Most of the contextual information used by the replacement policies are content related such as content access frequency or access time, but contextual information can also explore characteristics of the router node and the network.

We denote the best policy choice for CR in time interval I as

$$\omega_I^* = \omega_i \mid CE(\omega_i)_I > CE(\omega_j)_I, \forall j \leq m. \tag{5.2}$$

Network environments are dynamic, and changes such as variations in the content request pattern can lead to variations in policies' performances. Therefore, the best policy may unpredictably vary over time. The problem is choosing which policy should be executed by the CR at each interval I to maximize cache efficiency over a time horizon T.

Since we have continuous decision tasks during caching operations, the problem can be represented as an online learning problem. The cache has to learn which policy is more suitable to execute at each iteration. In our model, the cache operates only with one cache replacement policy at a time. Thus, the cache can measure the CE associated only with the running policy.

5.3 META-POLICY: LEARNING SUITABLE CACHING POLICIES ONLINE

Caching replacement policies behave differently according to their heuristic, but regardless of the policy, sequential $CE(\omega_i)$ measures could be seen as a sequence of random variables shaped by the running caching policy ω_i. CE measures are expected to be random since the sequence of future content requests is unknown. Also, the calculation of cache efficiency over a current interval does not consider efficiency values calculated over past intervals. This way, the measures are distributed random variables in the range [0,1].

Notice that the $CE(\omega_i)$ measures of each policy may follow distinct Gaussian distributions[1] since each policy implements particular eviction decisions. Therefore, our strategy's primary rationale is to model caching policies' efficiencies as distinct and fixed stochastic distributions. In this direction, we modeled the policy choice as an exploration and exploitation problem, typically covered by Online Learning with Partial Feedback (OLPF) algorithms.

The second rationale of our strategy is the decoupling of the content eviction strategy from managing the context information used by the policies. For that, we consider a Content and Context Management (CCM) module able to manage the cached contents and the context features associated with the set of content replacement policies Ω available in the cache.

We combined the two procedures to propose a caching meta-policy strategy capable of learning the suitable policies online and dynamically adapting to context variations that leads to changes in which policy is best. Figure 5.1 illustrates the two main components: an OLPF agent and a CCM module. The cache node works with the two components by implementing the caching meta-policy protocol described in Algorithm 1. In summary, the OLPF agent is responsible for choosing the policies to run in each interval, while the CCM module operates the cache and measures the cache efficiency CE. Then, the OPLF algorithm receives the CE measured to update its parameters used in the learning process.

[1] Observed from our experimental data (see Figure 6.2).

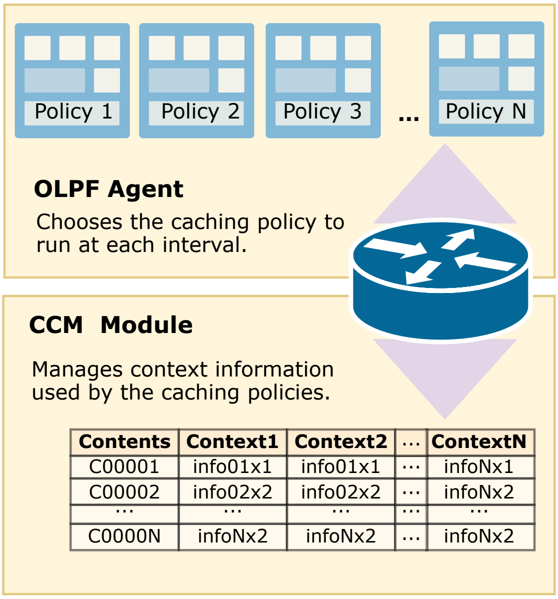


Figure 5.1 A meta-caching policy approach based on online learning with partial feedback and a content/context management module.

Algorithm 1: Caching Meta-policy protocol

---

foreach *iteration* I ***in*** *a time horizon* T **do**
  Choose policy $\omega_I \in \Omega$ according to the OLPF algorithm
  Call CCM(I, ω_I, Ω) to configure/operate the cache and compute the cache efficiency
  Receive the cache efficiency $CE(\omega_I)_I \in \mathbb{R}$
  Update policy choice parameters according to the OLPF algorithm
end

That model allows the cache node to act as an agent learning traditional stochastic Gaussian distributions regardless of the policy set. Moreover, the strategy enhances the cache with an upper abstraction level over the content eviction strategies, capable of adjusting the cache to network changes that influence the policies' performance. We detail the components in the following subsections.

5.3.1 Online Learning Agent with Partial Feedback

OLPF describes a set of sequential decision-making problems. An agent interacts with the environment online by deciding to execute one action over a finite set of actions. The chosen action yields a numerical reward after being executed, and the reward of the unplayed actions remains unknown. The agent has no previous knowledge about the mechanism generating the sequence of rewards for each action. This way, there is always uncertainty about whether the agent made the best choice. The agent can explore a new action at each iteration or exploit the best action learned in previous iterations.

The agent has to explore to learn its possibilities and also exploit to increase its gains in the long term. This exploration-exploitation dilemma has been the focus of extensive research over decades, giving rise to a diversity of OLPF algorithms targeted to variations of online learning problems.

In the caching meta-policy strategy, the decision-maker agent is represented by the cache node, and each cache replacement policy is an option to be chosen. For in-network architectures, cache nodes are the network routers enhanced with storage capacity. The cache works with a fixed set of policies. Then, in a continuous task, the agent chooses a policy to run inside a predefined interval of time. At the end of each interval, the agent receives the reward associated with the policy and evaluates whether the same caching policy will be running in the next interval or another policy. The reward is the cache efficiency measured inside the interval and relies only on the running policy. This way, the OLPF agent uses the reward obtained in each interval to learn the policies' distributions, i.e., the cache efficiencies' distributions. The agent is agnostic to the caching policy eviction logic and does not associate the choice with any traffic pattern or network-related characteristic.

Caching policies' performance may vary according to traffic characteristics changes, especially for highly dynamic networks with intermittent wireless communication. In that context, the distributions are expected to be non-stationary. Models with stationary distributions have no variation in which option will achieve the highest cumulative rewards over time. This way, the learning process focuses on finding which one is the best option. However, in non-stationary stochastic problems, the distributions may infrequently change in a time horizon. The learning process, thus, requires different strategies to adapt to possible variations of best arms. Therefore, the agent may employ algorithms able to deal with non-stationarity in stochastic models.

The stationarity degree and characteristics depend on the set of policies. The strategy effectiveness and adaptability rely on the employment of OLPF algorithms appropriate to the distributions' profile of changes.

5.3.2 Content and Context Management Module

Since the caching meta-policy strategy consults the OLPF agent at each iteration I to configure the cache with the policy chosen by the agent (see Algorithm 1), each iteration can be executed with a different policy ω.

A requirement to allow the dynamic change of policies during the cache operation is to maintain the context information associated with each policy's eviction logic in the set Ω. This way, the strategy implements a CCM module to store the context for all contents in the cache. Examples of context information commonly used by policies are the access frequency and last access time of contents. The management module keeps the context information used by all the policies in the cache policy set Ω regardless of which policy is executing. Besides, the module manages the content eviction engine by matching the stored context feature with the context feature used by the running replacement policy.

Upon the beginning of an iteration, a chosen policy begins its execution relying on the stored context. This way, it is possible to continue the cache operation from the current cache state left by the previous running policy. Otherwise, it would not be possible to change policies online. Algorithm 2 presents the pseudo-code of the CCM module.

The policy set can vary in number and replacement logic exploring different context aspects, such as content, router, and network properties. Several context factors can influence the performance of policies. Therefore, there is no single criterion for choosing the candidate caching policies to compose the set. Each cache on the network can work with a different set. The meta-policy strategy is a mechanism of choice, and the learning will converge to the most suitable policy among the options in the chosen set. In other words, the cache performance will converge to the performance obtained if only the best policy present in the set is executed.

Algorithm 2: CCM - Content and Context Management for the meta-policy strategy

Input: I, $\omega_I \in \Omega$, Ω
Output: $CE(\omega_I)_I$
Initialize cache miss and hit counters: $M = 0$, $H = 0$
foreach *request for content* c *during the iteration* I **do**
    if c *is not in the cache* **then**
        Increment cache miss counter: $M = M + 1$
        Elect content c' to evict from the cache according to ω_I eviction logic
        Add new content c in the cache
        foreach $\omega_i \in \Omega$ **do**
            Remove context data used by ω_i and related to c'
        end
    else
        Increment cache hit counter: $H = H + 1$
    end
    foreach $\omega_i \in \Omega$ **do**
        Update context data used by ω_i and related to c
    end
end
Compute and return $CE(\omega_I)_I = \frac{H}{M+H}$

5.4 META-POLICY GENERALITY

It is worth emphasizing the generic aspects of our proposed caching meta-policy (See Fig. 5.2). The strategy allows the employment of different policy sets and OLPF algorithms. It is also generic regarding the network type in which the cache operates. The strategy is suitable for caches operating in different settings such as ICNs, CDNs, and proxy caches. The next chapter presents a proof-of-concept evaluation in an ICN setting with traditional caching replacement policies and basic non-stationary stochastic bandit algorithms. The results reveal the potential for widespread use in different caching scenarios.

The meta-policy model can also be extended to collaborative caching systems. In that case, it is necessary to define a collaboration model and performance evaluation metrics for the overall network. We have pointed out research directions for collaborative caching models in section 7.3.

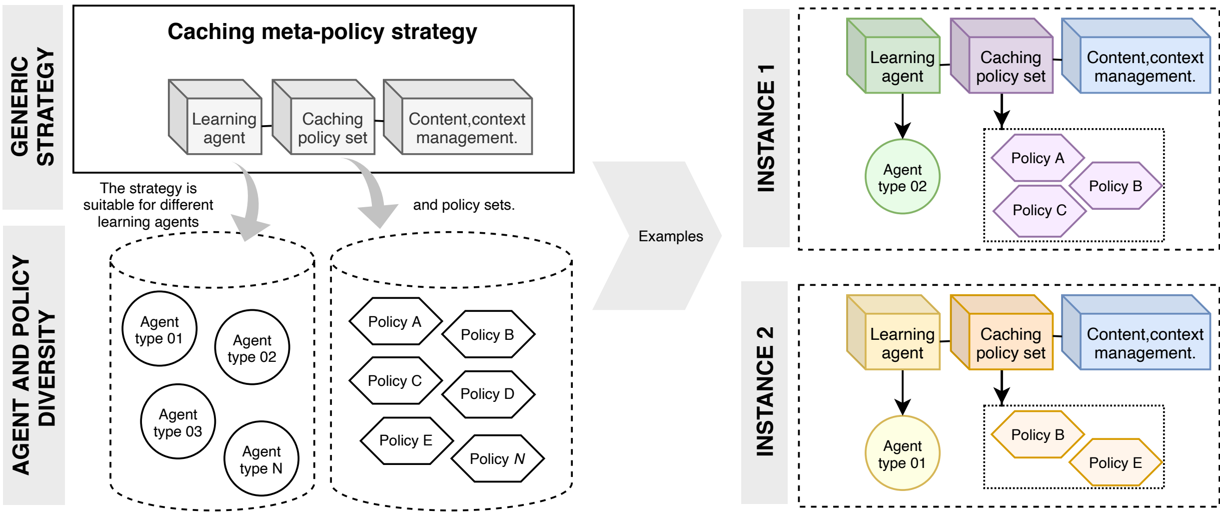


Figure 5.2 The generic aspects of the caching meta-policy encompass both the policy set with its corresponding context features and the OLPF algorithm.

5.5 CHAPTER SUMMARY

This chapter presented our proposed solution to address the problem of choosing caching policies during the design of caching systems. The solution models the caching policy choosing problem as an online learning problem with bandit feedback. Therefore, the chapter presents introductory concepts of online learning for caching systems and related works to position our proposed strategy. The strategy is a caching meta-policy that enhances the cache with an upper abstraction level above the logic of choosing content. To implement the strategy, the cache must work with a set of cache replacement policies. The meta-policy operates with an OLPF agent and a CCM module. In a continuous task, the OLPF agent chooses the policy to run at the cache and receives periodic feedback. The OLPF agent is designed to learn the best policy in the long run, and adapt to variations that impact the policy choice. Meanwhile, the CCM module guarantees the maintenance of the context information required to operate the policies. The meta-policy is a generic strategy and can support deploying a diverse set of self-contained caching policies in different scenarios.

Chapter 6

META-POLICY EXPERIMENTAL EVALUATION, RESULTS, AND ANALYSIS

To demonstrate the applicability and benefits of using the caching meta-policy strategy for choosing suitable caching policies, we have carried out a simulation-based study through the NDN architecture. NDN in-path cache works as an opportunistic cache to distribute contents across the network. This section details the evaluation methodology and discusses the experimentation settings.

6.1 EVALUATION ENVIRONMENT

We have implemented the meta-caching strategy in a modified version of the ndnSIM simulator (MASTORAKIS; AFANASYEV; ZHANG, 2017; MASTORAKIS et al., 2016) coupled with the ns3-gym framework (GAWŁOWICZ; ZUBOW, 2019).

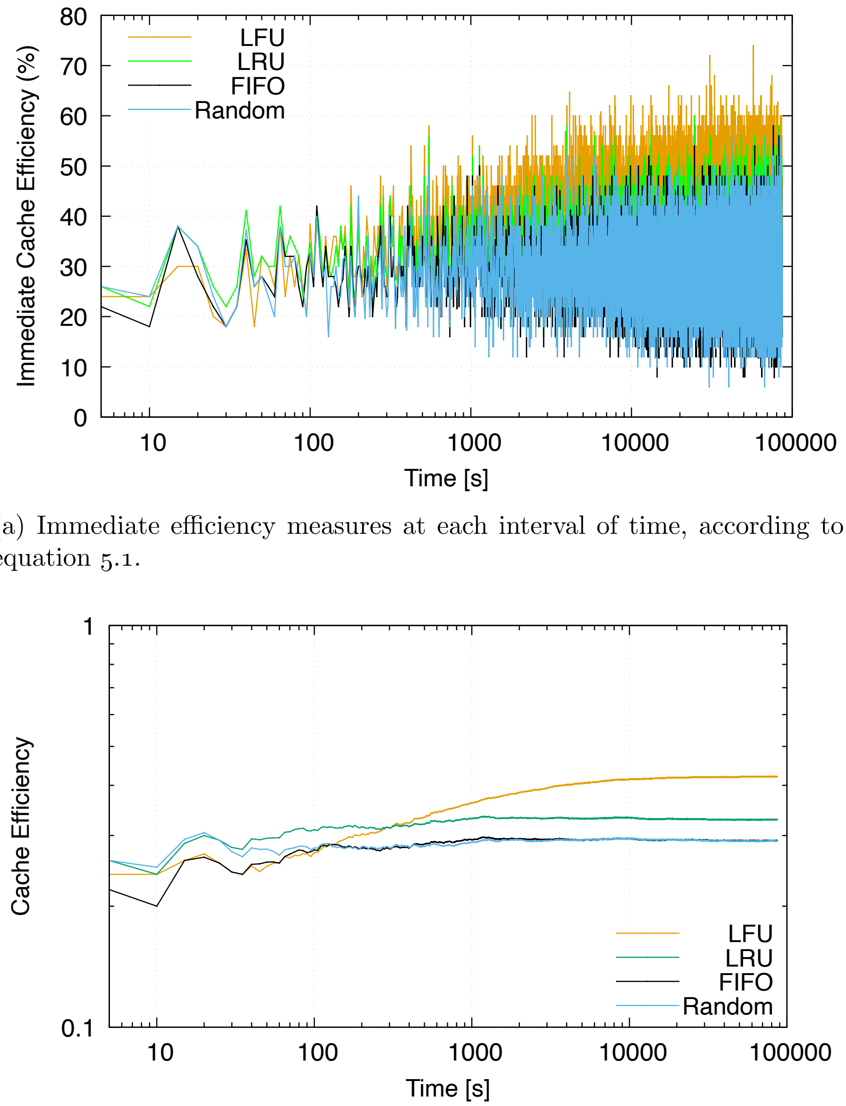
The ndnSIM is an open-source NDN simulator to reproduce discrete-event network scenarios. The simulator is a module of the Network Simulator (NS-3) framework (RILEY; HENDERSON, 2010), in which all network nodes implement the NDN protocol stack and the in-network cache structure. The cache works with a cache replacement policy configured at the beginning of the simulation. To implement our strategy, we have adapted the simulator to enable the change of policies during the execution of a simulation scenario. We also adapted the simulator to reproduce requests from real datasets.

The ns3-gym framework is a module of NS-3 designed to support the interaction of machine learning agents with the network environment. This way, a learning agent based on the ns3-gym framework can interact with the NDN cache node created by the ndnSIM. To evaluate our proposed strategy, we have implemented cache replacement policies on the ndnSIM and online learning algorithms on the ns3-gym. In the following subsections, we describe the policies and algorithms and details the scenario settings.

6.2 EXPERIMENTATION SETTINGS

6.2.1 Cache Replacement Policies

We have implemented a meta-caching policy strategy for the evaluation through an agent that chooses over a fixed set of four policies: LRU, LFU, FIFO, and Random. In their eviction logic, LRU removes the last accessed content, LFU removes the last frequently used content, FIFO removes the oldest content, and Random removes one content randomly. Those are traditional replacement policies inherited from the memory management of operating systems and used in Web cache networks.



(a) Immediate efficiency measures at each interval of time, according to equation 5.1.

(b) Total cache efficiency over time, i.e., at each iteration, we measured the total content requests and total cache hit from the simulation's start to the corresponding iteration.

Figure 6.1 Distribution of policies's performance for a single cache under IRM model.

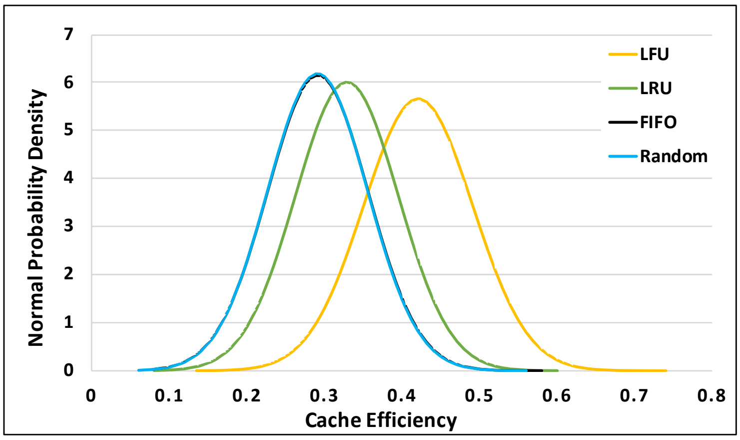


Figure 6.2 Graph of the probability density function for the distributions obtained by each policy under the IRM model.

To exemplify possible efficiency distributions for the distinct caching policies, Figures 6.1 and 6.2 illustrate the execution of LFU, LRU, FIFO, and Random, in a single cache scenario running individual policies separately. The scenario had a fixed catalog of 20.000 contents, static content popularity following the Zipf model with $\alpha = 0.8$, relative cache size of 5%, and request rate of 10 requests/s. We illustrate each policy's cache efficiency in a day, measured in 5 seconds intervals. The figures show the distribution of policies' performances from three different perspectives: the log-scale graph of all immediate measures by interval (Fig. 6.1 (a)), the total cache efficiency over time (Fig. 6.1 (b)), and the normal distribution for each policy (Fig. 6.2). Under the IRM model, LFU has optimal performance but requires some time to populate the cache with the most popular contents. Notice the variance in the first iterations, in which other policies performed better than LFU. Upon increasing the iterations number, LFU started to perform better, and the distributions became stationary. In this scenario, FIFO and Random obtained similar distributions, almost indistinguishable in the graph.

Different scenario settings would generate distinct distributions. Our proposed strategy stems from the analysis of similar distributions to learn which ones should be used by the cache. In the following, we present the online learning strategies we have employed in that learning process.

6.2.2 OLPF Algorithms

The OLPF problems describe bandit problems and can be tackled by a class of Multi-armed bandit (MAB) algorithms. Upper confidence bound (UCB) is a MAB strategy successfully used to solve stochastic bandit problems. In a stochastic setting, bandit algorithms usually estimate the arm's values by incrementally averaging its rewards in a time horizon. The more an action is taken, the more confidence we have that the average will reflect the action's actual value. UCB strategies add a fixed confidence interval to each arm's mean to estimate the expected arm's values optimistically. It is based on the concept of optimism in the face of uncertainty about the mean values. The strategy can gradually reduce the interval as the bandit gains more confidence in the mean values.

The literature presents several variations of UCB algorithms. A standard UCB deals with stationary problems in which the average considers the entire distribution evenly. In non-stationary scenarios, it is possible to include a discount factor to give higher weights to more recent rewards, similar to the ones proposed by (GARIVIER; MOULINES, 2011) and (SUTTON; BARTO, 2018). Another alternative is to consider only the most recent measurements in a sliding window over time (GARIVIER; MOULINES, 2011).

The experiments considered UCB algorithms for non-stationary stochastic bandits that choose the policy ω for the interval I according to the following equation:

$$\omega_I = \operatorname*{argmax}_{\omega \in \Omega} \left[\overline{CE}_I(\omega) + c\sqrt{\frac{\ln I}{N_I(\omega)}} \right] \tag{6.1}$$

in which $\overline{CE}_I(\omega_t)$ is the average efficiency for policy ω obtained before iteration I, and the square root part is the confidence interval; $N_I(\omega)$ is the number of times policy w has been chosen, and c is a fixed parameter to tune the effect of the confidence interval thereby controlling the degree of exploration (SUTTON; BARTO, 2018).

The algorithms are (i) UCB with discount factors (UCB$_d$) and (ii) sliding-window UCB (SW-UCB). For the UCB$_d$, we have applied the incremental Exponential Recency-Weighted Average (ERWA), according to equation 6.2,

$$\overline{CE}_{I+1}(\omega) = \overline{CE}_I(\omega) + d\Big[CE_I(\omega) - \overline{CE}_I(\omega)\Big] \tag{6.2}$$

in which $d \in (0, 1]$ is a step-size parameter that works as a discount factor in the average learning process. The discount factor adjusts distinct weights over the reward distribution, wherein higher values emphasize recent rewards. So, in the experiments, we adopted $d = 0.2$ and $d = 0.8$ as two opposite values to evaluate the bandit's adaptability. The SW-UCB used the simple average and tuned the window size parameter according to Garivier e Moulines (2011).

Parameter	**Values**			
Caching policy set	LRU, LFU, FIFO, Random			
OLPF algorithms	UCB$_{d=2}$, UCB$_{d=8}$, SW-UCB			
Agent iteration interval	2, 3, 4, 5, 10, 15, 20, 25, 30 s.			
Number of cache nodes	1, 4, 9			
Node positions	edge, intermediate positions			
Cache size	2, 4, 5, 6, 8, 10, 12, 14, 16, 18, 20%			
Content request pattern	**IRM model**	**Boston sample trace** (1 day)	**Youtube sample trace** (1 day)	**DEC sample trace** (1 day)
Content library size	1.000, 20.000	≈ 20.000	≈ 30.000	≈ 600.000
Content Zipf(a)	0.8	≈ 0.80	≈ 0.53	≈ 0.63
Total content request	2, 5, 10 /s.	≈ 10.000	≈ 50.000	≈ 1.5 million

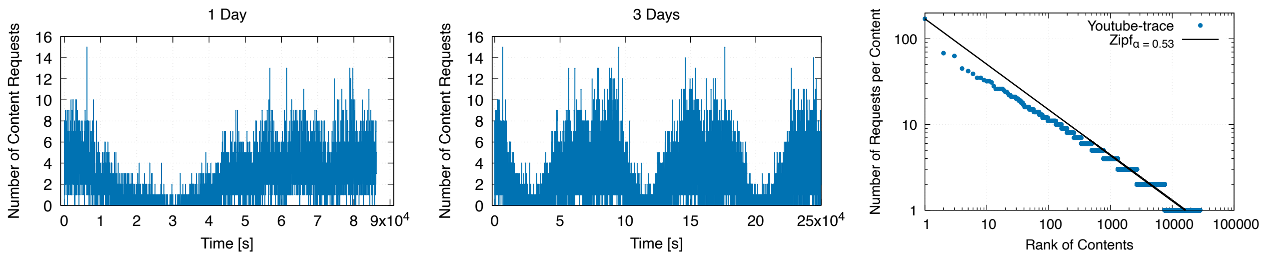
Table 6.1 Scenarios parameters.

6.2.3 Scenarios and Evaluation Metrics

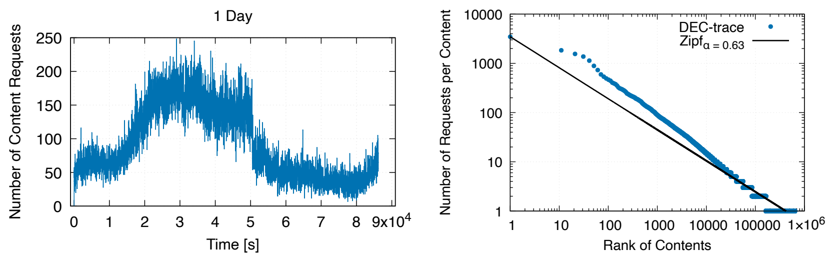
We have conducted experiments to analyze our strategy effectiveness in choosing the best policy for different scenarios. The scenarios contain variations on content request patterns, cache sizes, and locations of the cache node into the topology. We also analyzed the impact of different interval times that the learning agent interacts with the cache. Table 6.1 summarizes the scenarios parameters and their respective values.

We evaluate the convergence of the cache's performance to the performance of the best policy present in the policy set. We expect the performance of a cache executing the meta-policy to converge to the performance of the policy best suited to the context of the cache's operation.

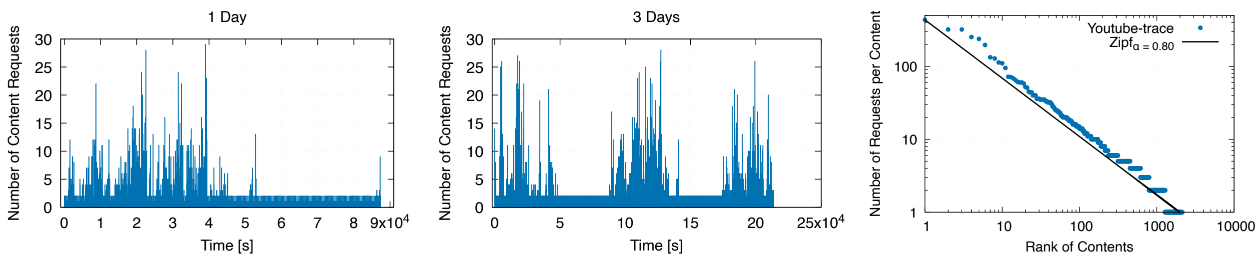
The strategy efficiency was compared with the correspondent benchmark scenario consisting of one replacement policy throughout the simulation period.



(a) Youtube.



(b) DEC.



(c) Boston.

Figure 6.3 Sample traces for three distinct real content request datasets. The number of content requests were measured in 5 seconds intervals. The distributions of content requests per content refers to measures in one sample day.

6.2.3.1 Datasets To accomplish variations of the access pattern, we have performed experiments with the IRM request model implemented in the simulator, and also on three public datasets suitable for caching experimentation: a dataset of user requests for Youtube videos (ZINK et al., 2008), a Web Proxy dataset from Digital Equipment Corporation (DEC) (COOPERATION, 1996), and web traffic traces from Boston University (CUNHA; BESTAVROS; CROVELLA, 1995). All datasets contain trace files with users' content requests and the respective timestamps. Figure 6.3 illustrates sample traces of each dataset. The figure shows the different content request rates over time and the distribution of content popularity in a sample day. We carried out experiments with different periods on the datasets and selected one-day traces to present the results.

6.2.3.2 Network Topology Regarding the network topology, we divided the evaluations into single-cache and multi-cache scenarios (Fig. 6.4). In the single-cache, we have tested the impact of different access patterns, cache sizes, and agent iteration times.

In the multi-cache scenario, we present an analysis regarding the impact of different node positions on the network. We aim to explore the variance of suitable policies for the individual caches. To this end, we first carried out experiments with a tree topology composed of one intermediate caching node and three edges caching nodes. Then, we expand the study on different intermediate positions with nine caching nodes arranged in a 3x3 grid topology. We placed the producer and consumers at opposite ends of the same grid's diagonal. The consumer issues content interests that transverse the caching nodes up to match searched contents in respective caches or the content producer. When a cache node does not have a requested content, it broadcasts the incoming content interest to neighbor nodes. This setting allows us to simulate dense cache connections while exploring the different in-network cache positions. The different positions allow each cache to have unique traffic views and thereby possible variations of suitable policies. All caches have similar sizes.

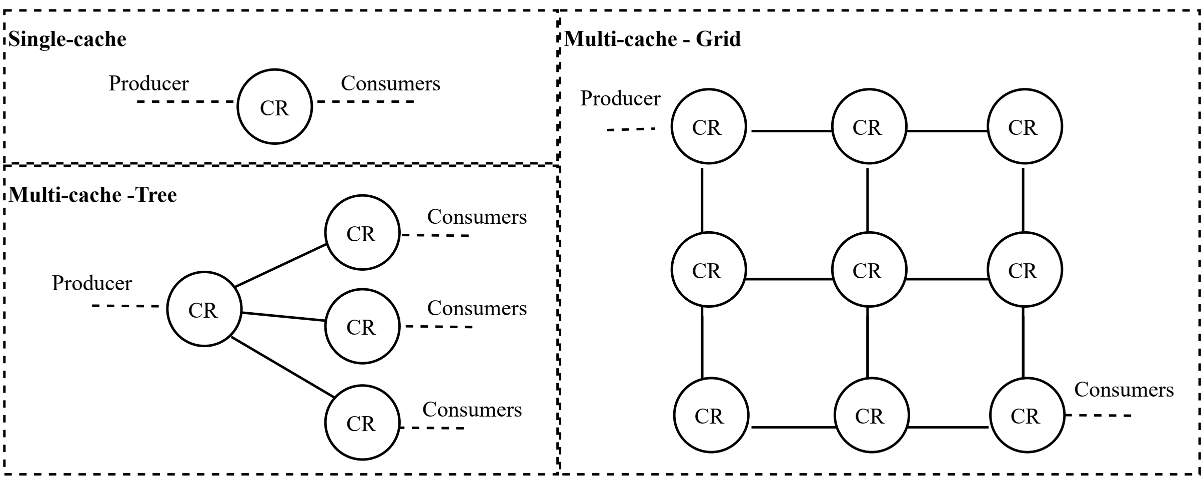
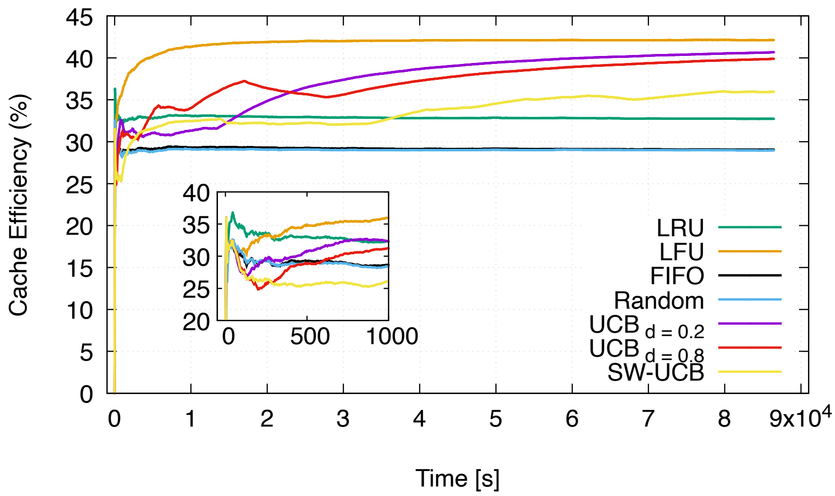


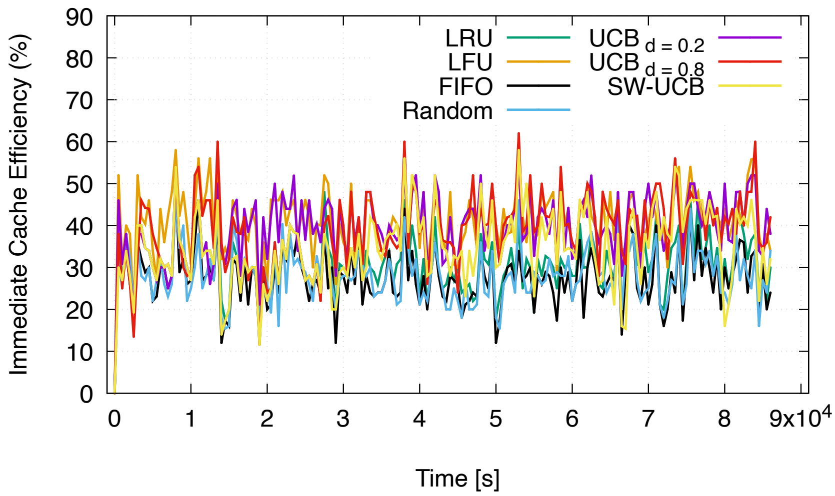
Figure 6.4 Cache network topologies for the experiments. The network devices implement the NDN protocol stack, and all the NDN routers have cache capacity.

6.3 CACHING META-POLICY RESULTS AND ANALYSIS

This subsection presents our findings in applying the meta-policy strategy under different network scenario variations. In general, the policies presented different behaviors for each scenario, and most bandit configurations performed close to the best-fixed policy for all scenarios. We first show the meta-strategy application in the single-cache IRM scenario of section 6.2.1 with the three variations of bandit algorithms.



(a) Total cache efficiency over time.



(b) Immediate efficiency measures at each interval. The graph shows one every 100 measures to improve visualization.

Figure 6.5 Cache efficiency for the single-cache IRM scenario. Simulation parameters: relative cache size: 5%, iteration time: 5s, c = 0.2, warm-up period: 200 s; 20.000 distinct contents, Zipf(a)=0.8, request rate: 10 requests/s.

Figures 6.5 and 6.6 depict the results of letting bandits agents learning the best policy online. The figures shows: the total cache efficiency over time, i.e., at each iteration, we

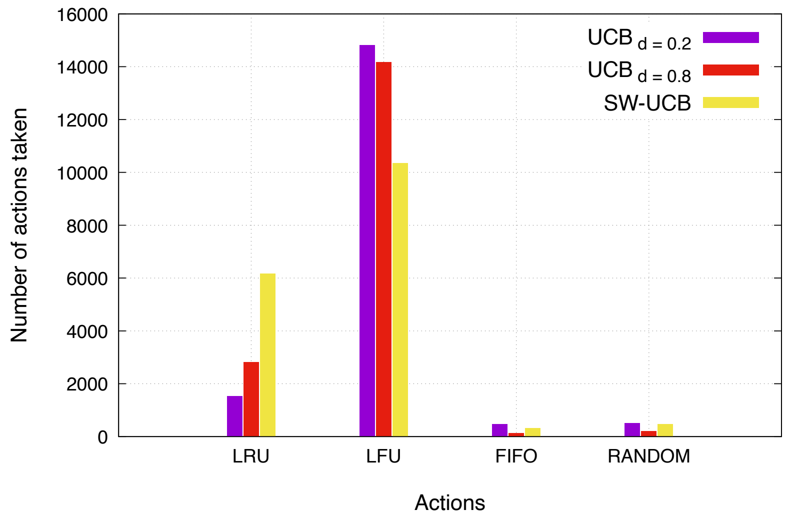


Figure 6.6 Distribution of actions taken by OLPF agent.

measured the total content requests and total cache hit from the simulation's start to the corresponding iteration (Fig. 6.5 (a)), the immediate measures of cache efficiency at each 5 seconds intervals (Fig. 6.5 (b)), and the distribution of policies choices by bandit algorithms (Fig. 6.6).

The agents explored all policies in the set according to its learning algorithm. As we have described, LFU is the best policy in the set for the IRM model, followed by LRU. Figure 6.5 (a) shows that both UCB$_{d=0.2}$ and UCB$_{d=0.8}$ performed closer to LFU in the long run. Even applying discount factors, UCB$_d$ is influenced by the initial distribution values. However, in this scenario, both lightweight and aggressive discount factors could minimize the influence of lower LFU initial values. A stationary strategy would not be able to have that effect. Although SW-UCB considers more recent distribution values and is not impacted by the initial values, the bandit had struggled to adapt by showing a slower learning curve. One possible reason is that the policies may obtain immediate measures with nearby values that may overlap in some iterations, as shown in part (b) of Figure 6.5. Hence the importance of learning with many iterations of the agent to estimate more confident policy values.

It is worth mentioning that the distributions perceived by the bandit in the learning process are not exactly the same as those obtained when a single policy is executed throughout the entire cache execution. Inside the bandit, the policies obtain different reward values due to the different cache states at each iteration that starts with a new policy. Even with fragmented executions, the policies manage to maintain their distribution characteristics. However, we notice that in such a static setting, frequency-based policies may perform optimally when executed alone but may not perform the same way inside the bandit. The reason is the possible loss of high-frequency contents during other policies' execution. In similar cases, the bandits can only approximate their performance.

6.3.1 Impact of Content Request Patterns

Figures 6.7, 6.8, and 6.9 depict the results of applying the caching meta-policy strategy in scenarios with different content request patterns. The figures show the cache efficiency for one sample day of Youtube, DEC, and Boston traces, respectively. We maintained the same simulation parameters used in the IRM scenario described above while varying the input requests. Unlike the IRM model, the real web traces present degrees of temporal locality between content requests. Temporal locality describes correlation properties in content requests, in which recently accessed contents are likely to be reaccessed shortly. Policies such as LRU tend to present better performances when the temporal correlation is the prevailing characteristic of the content request pattern.

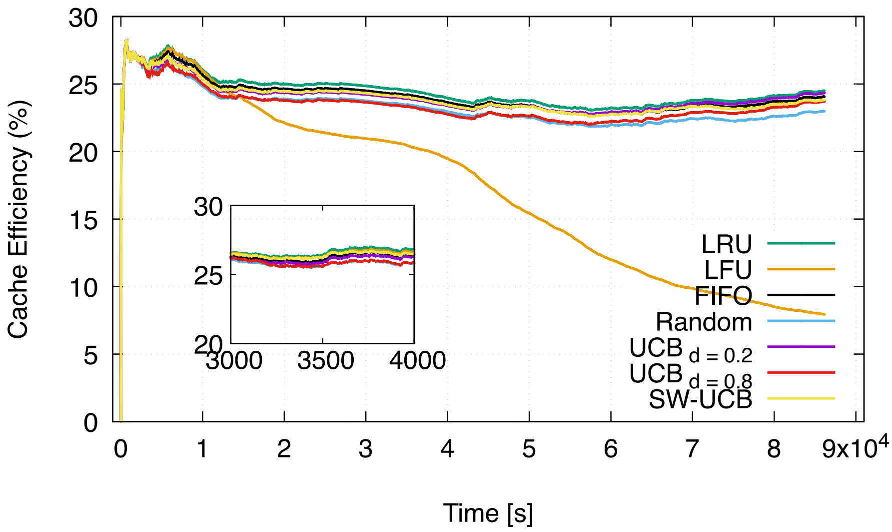


Figure 6.7 Youtube trace. Cache efficiency for single-cache scenario. Simulation parameters: relative cache size: 5%, iteration time: 5s, $c = 0.2$, warm-up period: 200 s.

For Youtube trace, LRU stands as the best policy in the set, and the performance of all bandits approached LRU. The other policies performed well and approximated to LRU, except for LFU, which had degraded performance as the simulation progressed. Therefore, in more dynamic scenarios such as with content popularities changes, the bandits can perform as well as the best-fixed policy in the long run. In such a scenario, even stationary bandits would perform well.

The DEC trace results presented interesting behavior with the variation of the best policy during the simulation time. After an initial period in which all policies appeared to perform almost evenly, LFU started to perform better and became the best option. Still, LFU gradually lost performance due to content popularity profile changes, and LRU remained stable as the best choice. Since LRU uses the least recent approach to evict content, it is more appropriate for dynamic scenarios. However, the cache size is also a factor that impacts cache efficiency. The increase in cache size could mitigate LFU performance degradation in such cases. Regarding the bandit's behaviors, $\mathrm{UCB}_{d=0.8}$ demonstrated better adaptability. The bandit approached LFU and then adapted to LRU

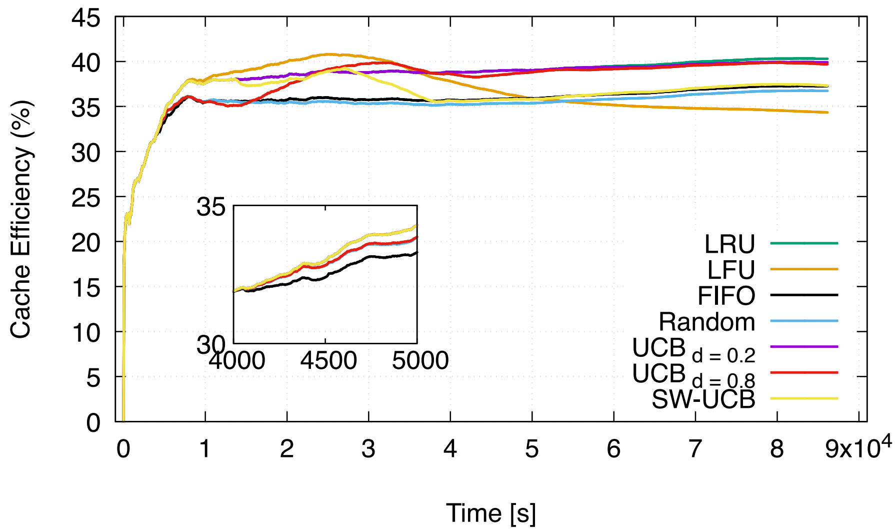


Figure 6.8 DEC trace. Cache efficiency for single-cache scenario. Simulation parameters: relative cache size: 5%, iteration time: 5s, $c = 0.2$, warm-up period: 200 s.

when LRU became the best option. UCB$_{d=0.2}$ got stuck in LRU from the beginning, probably due to its higher initial distribution values when LFU went through its natural learning curve. The trace results reaffirmed the better adaptability of the meta-policy strategy to more dynamic scenarios since most bandits performed as well as the best-fixed policy in the long run.

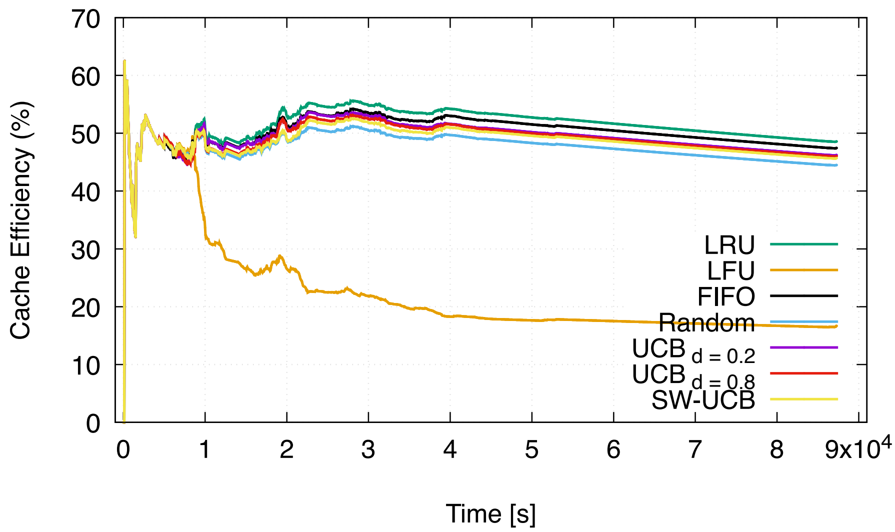
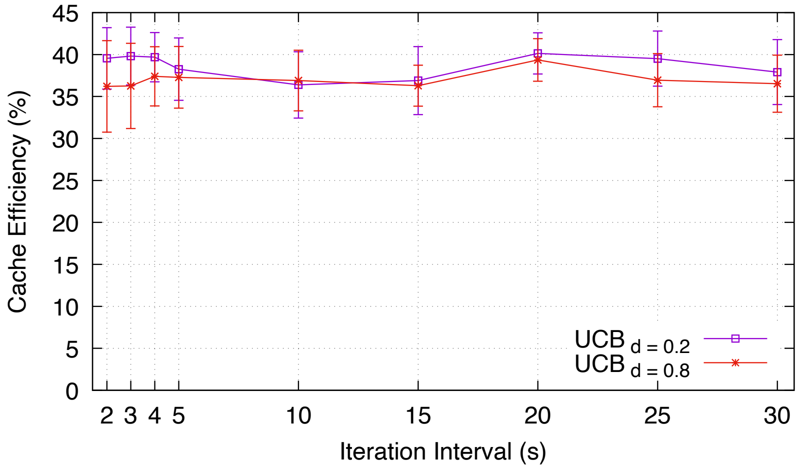


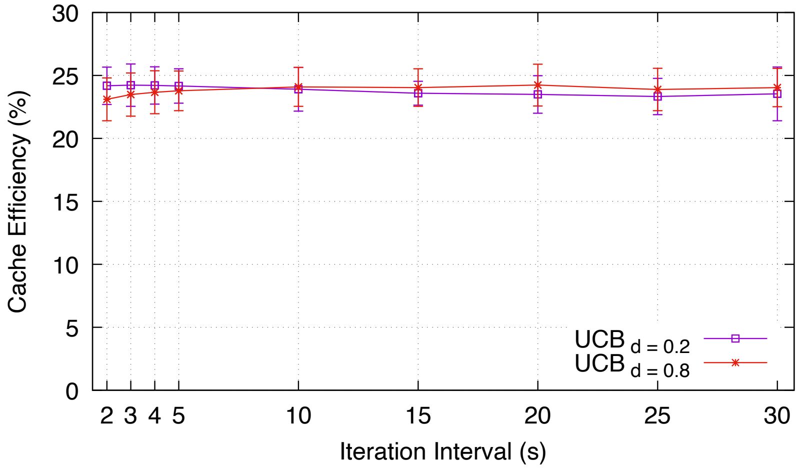
Figure 6.9 Boston trace. Cache efficiency for single-cache scenario. Simulation parameters: relative cache size: 5%, iteration time: 5s, $c = 0.2$, warm-up period: 200 s.

For the Boston trace, the policies behaved very similarly to the Youtube trace; however, the agents could not approach the LRU performance in the same way. The reason was the absence of request patterns in approximately half of the trace time. In the trace, requests are practically extinguished in the second half of the day. Thus, the performance measurements obtained by the agents in the absence of requests were not able to measure the value of each policy. Still, the learning process in the first half of the time allowed the agent to perform close to the best policies in the set.

6.3.2 Impact of Agent Iteration Time Interval



(a) IRM model.



(b) Youtube trace.

Figure 6.10 Average cache efficiencies for different agent iteration intervals in a single-cache scenario. Average over 10 runs. Common simulation parameters: duration of each run: 1 day, relative cache size: 5%, $c = 0.2$, warm-up period: 200 s; For IRM model: 20.000 distinct contents, Zipf(a)=0.8, request rate: 10 requests/s.; The Youtube trace samples are similar to the sample in Table 6.1.

The iteration interval determines the time an agent will intervene in the cache to evaluate the running policy and perform policy changes. As such, the interval sets the time a policy has to run before being evaluated. The shorter the interval, the more interactions the agent will make in a time horizon, and thereby the agent will have more confidence in the estimated policies' values. However, small intervals may not adequately reflect the value of the evaluated policies, as the policies need time to show their ability to assist content requests. In contrast, long intervals lead to slow convergence in agent learning.

We assessed the impact of different time intervals on cache performance when applying our meta-policy strategy. Figure 6.10 depicts the results of experiments with the IRM model and Youtube traces. There were variations for each interval as the distributions obtained are distinct, but the average performance for each interval size remained nearby to each other. The variation was more significant for the IRM experiments, while the cache performance remained almost constant for the Youtube experiments.

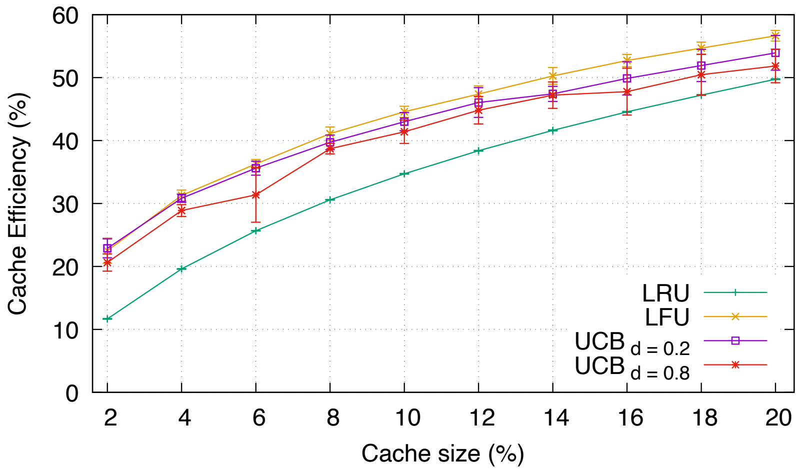
In general, the cache maintained the performance regardless of the time interval used by the agent. The experiments used the static time configuration, but the cache can adopt dynamic approaches to increase or reduce the interval according to variations in the network in which the cache operates.

6.3.3 Impact of Cache Size

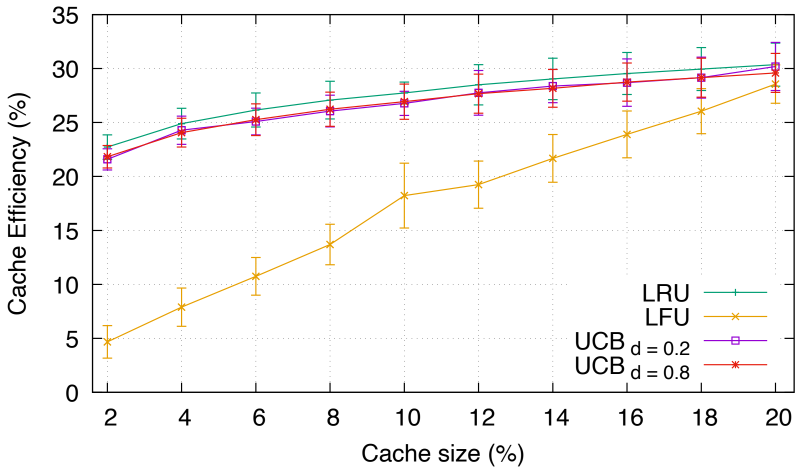
The cache size is one of the context factors correlated to the policies' performances and can influence the policy choice. We carried out experiments with cache size variations in scenarios with the IRM model and Youtube traces to analyze the impact on our proposed method. Figure 6.11 depicts the results. In general, the direct impact of cache size variations relied on the policies' performances. However, we notice a not negligible impact on the average performance of the agents.

Regarding the policies performances, the increase in cache size usually causes two effects: first, the cache performance naturally increases regardless of the policy since there is more space to store popular content, but the performance gain is not linear with increasing cache size; and second, the policies performances tend to converge for reasonably large cache size. The IRM and Youtube scenarios presented both effects but with very different granularities, as shown in the picture.

Regarding the caching meta-policy strategy, the agents choose between the four replacement policies as in the previous experiments. The figure shows the comparison with the two most representative policies. In the IRM experiments, the agents generally achieved better average performance for smaller cache sizes. There were more variations for larger cache sizes, as seen from the increase in the standard deviation. That means the UCB agents explored the policies more. This behavior could be associated with the slight convergence of policies' performance and the learning pattern of UCB algorithms. For the Youtube experiments, the agents performed very similarly for all cache sizes. The combination of both results reinforces that variations in the policy set's performance pattern can influence the agent's learning rate, not the cache size directly.



(a) IRM model.



(b) Youtube trace.

Figure 6.11 Average cache efficiencies for different cache sizes in a single-cache scenario. Average over 5 runs. Common simulation parameters: duration of each run: 1 day, relative cache size: 5%, iteration time: 5s, $c = 0.2$, warm-up period: 200 s; For IRM model: 1.000 distinct contents, Zipf(a)=0.8, request rate: 5 requests/s.; The Youtube trace samples are similar to the sample in Table 6.1.

6.3.4 Impact of Node Location in the Network

This section presents our experiment in the multi-cache topologies described in section 6.2.3.2. The experiments aimed to explore the variance of suitable policies for different cache node positions on the network, and thereby our strategy's effectiveness in learning accordingly.

Multiple cache levels naturally present variations in the traffic characteristics perceived by each cache. The reason is the filtering effect when a cache closer to the user hits a content request. The cache does not propagate that request to the rest of the network and propagates only the miss requests to upper-level caches. This behavior modifies the original characteristics of the traffic and directly impacts each cache's choice of policies. Therefore, a homogeneous policy configuration may not adequately address the individual cache needs.

Besides, a policy running in one cache can influence the efficiency of all neighbor caches. Therefore, an appropriate choice of policies would consider that interaction to ensure overall network efficiency. Yet, this section presents a simple model of independent and distributed bandits. We show that, even without collaboration, most bandits can learn and adapt the policies with all routers executing the meta-policy selfishly. We compare the results with the homogeneous policy setting, i.e., all routers with the same replacement policy.

We first show the results of the experiment in the tree topology. The topology has one intermediate router and three access routers. We aimed to evaluate the meta-policy adaptability in the intermediate node while all edge routers also run the meta-policy. We set up distinct traffic profiles for each access router: IRM traffic, parts of DEC trace, and Youtube trace. Thus, the intermediate node receives filtered and mixed traffic of all three edge nodes. Figure 6.12 depicts the results. The edge nodes maintained the cache behavior discussed before in the single cache experiments according to their respective traffics. Meanwhile, the intermediate cache node was also able to adapt its policy on-demand. The graphs show the efficiency of an agent in the intermediate node relative to the efficiency of all other edge nodes' efficiencies running the same agent type. Likewise, the single policies' efficiencies are relative to the same policy running in all edges.

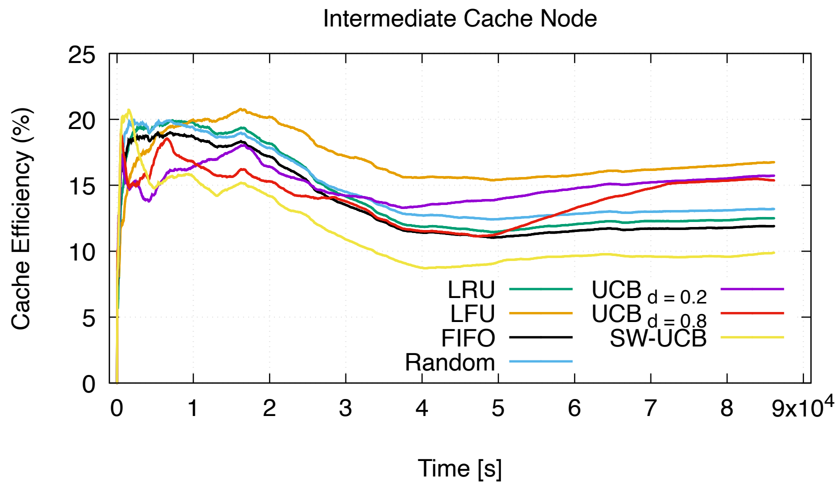
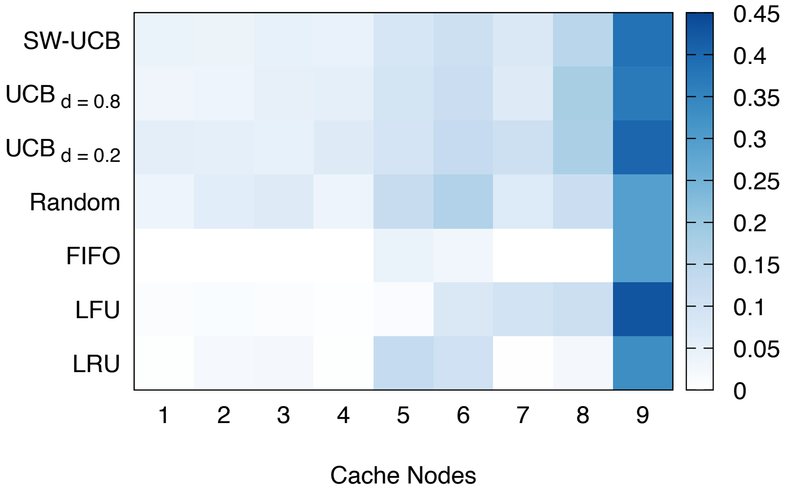
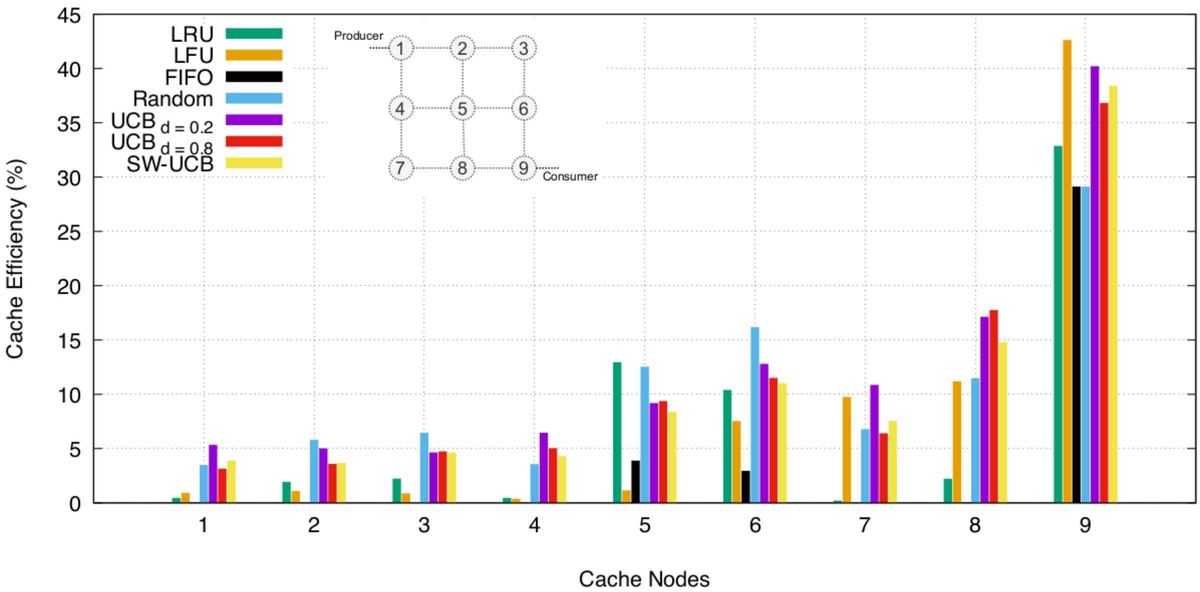


Figure 6.12 Cache efficiencies for different node positions in the tree topology. Simulation parameters: relative cache size: 2%, iteration time: 5s, c = 0.2, warm-up period: 200 s; For IRM model: 20.000 distinct contents, Zipf(a)=0.8, request rate: 2 requests/s; For Dec trace: ≈ 65.000 requests, ≈ 37.000 contents; For Youtube trace: ≈ 45.000 requests, ≈ 30.000 contents.

To further explore the adaptability in intermediate cache nodes, we performed experiments in the 3x3 grid topology presented in section 6.2.3.2. We placed the content consumers at one cache node to isolate the edge effect and evaluated the agent's ability to learn from the different intermediate node positions.



(a) Heatmap of individual cache efficiencies.



(b) Histogram of individual cache efficiencies.

Figure 6.13 Cache efficiencies for different node positions in the IRM 3x3 grid scenario. Simulation parameters: relative cache size: 5% for each cache node, iteration time: 5s, $c = 0.2$, warm-up period: 200s, distinct contents: 20.000, Zipf(a)=0.8, request rate: 10 requests/s.

Figure 6.13 illustrates the caching nodes' efficiencies for the experiment with IRM traffic. We kept the same simulation parameters as the single-cache IRM scenario presented earlier (see Fig. 6.5), e.g., the same consumer request rate and the number of distinct contents. Notice the variance in both cache efficiencies and policy behaviors according to the node position. In the present scenario with equal caches setting, the variations are mainly due to each node's particular traffic view.

The caches enhanced with the meta-policy were able to adapt their policies on-demand. The bandits presented variations in efficiency for each node position as the single policies due to the different traffic views in each position. Overall, $\text{UCB}_{d=0.2}$ stood out with the highest performance levels. As mentioned earlier, the bandits' performances are linked to the policy set. FIFO has performed poorly in the cache network scenario for most node positions, and the bandits managed to maintain good performance even using FIFO in their continuous learning process. Regarding the cache efficiencies compared with the single-cache IRM scenario, as expected, the overall cache network efficiency improves, i.e., the sum of all individual cache efficiencies, since we have increased the total system cache capacity. Naturally, the average cache efficiency reduces due to a combination of the hierarchical grid structure and the configuration parameters of the simulated scenario (e.g., content popularity, request rate, and cache size).

In caching networks, each cache could work with different policy sets and bandit algorithms. More realistic traffic presents variations in the temporal locality patterns perceived by each cache. The traffic received by caches closer to the users presents strong temporal localities. As cache levels filter requests, the temporal locality intensity becomes gradually weakening, and the traffic profile at upper-level caches becomes more random. Real caching networks would similarly benefit from the online policy adaptation.

6.4 CHAPTER SUMMARY

The chapter presented evaluation experiments of the caching meta-policy strategy. We have implemented the strategy in a NDN simulator and evaluated in conjunction with algorithms for non-stationary stochastic bandits. We carry out experiments with different traffic patterns, cache size, agent iteration time, and cache position in multi-cache network topologies.

The main evaluation outcomes are:

- The caching meta-policy strategy can learn the best caching policy for a variety of scenarios, with different traffic patterns and cache sizes.
- The combination of the adopted policy set with non-stationary UCB algorithms showed better learning for dynamic traffic patterns.
- Long periods with no traffic patterns compromise an effective estimation of policy efficiency, and thus make it difficult for the agent to learn.
- The size of interval time in which an agent interacts with the cache had low impact on the average cache performance.
- Variations in the cache size had indirect impact on the meta-policy due to the influence on the performance behavior of the policy set.

- The meta-policy can be employed in multiple hierarchical caches. We presented a simple model of independent and distributed caches, but the evaluation in multi-cache scenarios requires the modelling of collaborative caching systems.

Chapter

7

CONCLUSIONS

This section concludes the thesis document. We present concluding remarks about the work we carried out along the thesis research (Sec. 7.1); we pinpoint related subjects out of the scope of this thesis (Sec. 7.2); and, last, discuss a broad of future research directions (7.3).

7.1 RESEARCH CONCLUSION

In-network cache architectures, such as ICNs, have proven to be an efficient alternative to deal with the growing content consumption on networks. In caching networks, any device can potentially act as a caching node. In practice, real cache networks may employ different caching replacement policies by a node. The reason is that the policies may vary in efficiency according to several context factors, and a better understanding of the relationship between context features and cache policies becomes necessary. In addition, caching systems would benefit from models designed for choosing caching policies appropriately to cache contents on-demand and over time. The lack of suitable policies for all nodes and scenarios undermines the efficient use of available cache resources.

This thesis presented relevant contributions regarding the delimitation of context features that impacts cache performance, and a method to enhances the cache with the meta capacity of learning suitable policies on-demand. To the better of our knowledge, this work is the first attempt to collect context features' impact on caching replacement policies, and also to employ a caching meta-policy model to improve caching systems. Although the research scope of the thesis was on ICNs, the context findings and policy choice strategy can be generalized and extended to other caching systems, such as CDNs and Web proxy caches.

The results of the context delimitation studies encompassed evidence-based proof that efficient utilization of cache resources relies on deploying cache replacement policies according to the overall network context. The results reaffirm the absence of a single optimal strategy to meet the requirements of all network since the caching policies' performances vary according to different context characteristics. Moreover, the results included catalogs of features from the content, node, and network dimensions explored by the policies. The studies also embraced the human dimension as an incipient and potential factor to be incorporated in the cache system design.

The design of caching policies has been seen a paradigm shift with the application of online learning techniques in recent literature. In this work we explore the online learning potential from a different perspective. Instead of proposing new caching policies, we have introduced a meta-policy approach that models the replacement policy choosing problem as an online learning with bandit feedback problem. The meta-policy is a general model to adapt the cache to suitable policies according to the cache context. Such strategy opens up straightforward mappings to build self-driven intelligent networks. In this thesis scope, we have presented preliminary simulation-based evaluations of the meta-policy strategy in ICNs, and the results have shown its adaptability and effectiveness for different scenarios. Nevertheless, the evaluation can be extended to real world deployments with different policies sets and reinforcement learning algorithms.

7.2 OUT OF SCOPE SUBJECTS

The following topics were out of this thesis scope:

New cache replacement policy proposal. We did not intend to propose a new replacement policy. Instead, we propose a method for choosing between existing ones.

Cache replacement policy comparisons. We also did not aimed to explain individual behaviors and rationals of the cache replacement schemes proposed in the literature. Instead, we aimed to understand the context information intersecting the schemes and their possible effects.

7.3 FUTURE RESEARCH DIRECTIONS

In this section, we discuss different research directions for context-aware cache replacement schemes in ICNs.

Collaborative caching systems:

Collaborative caching systems are complex and may employ different caching strategies. One approach to tackle the correlation in cache's decisions is to model the problem as a combinatorial MAB (CMAB). In CMAB, a bandit plays a set of arms together and observes their individual rewards. This way, one action corresponds to a combination of different arms. The learning process, thus, aims to converge to the best combination. Such strategy has been shown effective for proactive cache content placement in mobile BSs (BLASCO; GÜNDÜZ, 2014). In that case, the contents are arms for a BS bandit player. The general problem is to choose the best combinations of contents to be cached over a fixed content set. Regarding the caching replacement policy choosing problem for multiple caches, a possible combinatorial model would have a centralized entity deciding over the policies' combinations for all caches together. Instead of accounting for the individual caches efficiencies separately, that centralized entity would account for the aggregated network efficiency. Such a model requires investigation of computationally efficient multi-task bandit approaches.

Meanwhile, a centralized bandit convergence would be impractical in feasible times for dynamic, distributed, and heterogeneous caching settings with intermittent connections. Besides the variance of caching nodes, heterogeneous devices could work with different policy sets and bandit algorithms. Moreover, the caching nodes may employ different bandit iteration times. Such cache networks are well suited for collaborative multi-agent MAB models. Convergent learning is still a challenge but feasible in combination with solutions designed for game-theoretical problems.

Context information management:

Dealing with contextual information requires well-defined procedures on acquiring, representing, reason, and distributing the information. Context information management is widely studied and applied in many sciences that rely on context-awareness (PERERA et al., 2013). Still, it is a challenge for complex systems such as dynamically distributed networks to efficiently perform online context management, especially when there is a need to represent a high number of dimensions and elements relevant to represent the domain. The integration between ICN and SDNs (KIM; CHUNG; MOON, 2015; CHARPINEL et al., 2016; YAO et al., 2016; KALGHOUM; GAMMAR; SAIDANE, 2018; LIU et al., 2018; SAADEH et al., 2019) can further benefit context management solutions because of the SDN paradigm's centralized control view. It is necessary to investigate what context information could be efficiently handled by central controlling.

The sets of context features identified within our proposed classification are enablers to a semantic representation of the context domain and can be extended or adapted according to different application requirements. However, towards an efficient real-world deployment, there is also the need to argue about the *quality* of context information. Quality can associate many aspects like reliability, precision, timeless, access right, significance, granularity, and completeness. Those aspects are translated into metrics defined by the science of Quality of Context (QoC) (BUCHHOLZ; KÜPPER; SCHIFFERS, 2003). The relevance of QoC metrics varies following the type of information. Hence, different QoC metrics should follow the different context subcategories in each context dimension.

Scalability of context suitability:

Exploring context information is essential to address a mismatch between caching policies and emerging networks. This exploration contributes to achieving more potentially precise and customized techniques. However, the more the use of contextual information, the more computationally expensive the caching scheme might become. The need to compute more context information may increase the complexity of the caching policy itself. Therefore, it is essential to investigate the *performance cost* of individual context information and the solution as a whole. The performance cost depends not only on managing the information but also on how the policy treats the information.

Machine learning techniques:

In addition to being used for context information inference (ZHAO et al., 2017; NAKAYAMA; ATA; OKA, 2015; LIU et al., 2018), machine learning techniques can investigate how to exploit better context information to optimize the eviction process. In one perspective, machine learning techniques could select which contextual information is most relevant and should shape the eviction process. The relevance of contextual information may vary depending on the network and objectives. This way, given a network with a set of available contextual information, it would help investigate how to choose what should be used by the eviction scheme to increase network performance.

In another perspective, the techniques can direct the learning of the best kind of policy based on what context information is available. Reinforcement learning techniques have been successfully applied for caching schemes (SUNG et al., 2016; SADEGHI; SHEIKHOLESLAMI; GIANNAKIS, 2017). However, in those works, the context state is represented solely by the cached contents in an instant of time. It would be relevant to extend the concept of context to represent the state with more available information that would impact the learning policy process. Depending on the number of context information used, there may be a large space of possible states, which will require considerable computational effort to represent the possible variations. When most of the states are rarely revisited, the chosen technique must deal with some sort of generalization. Furthermore, model-free techniques are best indicated when there is no previous knowledge dataset to help the decision process.

Dynamic and adaptive instantiation of cache policies:

Along with SDN and ICN, Network Function Virtualization (NFV) techniques are strong candidates for realizing and fostering next-generation networks (Zhang et al., 2018; SAADEH et al., 2019). Through the network function virtualization concept, in-network caching strategies can quickly execute as Virtual Network Function (VNF) along with some management structure. This combination paves the way for efficient deployment of adaptive caching policies according to the context's dynamic changes. To realize a plug-and-play vision of virtual function would be interesting to have a rich repository of heterogeneous caching functions and multi-attribute functions exploring different combinations of context information.

Human aspects:

In recent years, the community has witnessed a growing number of researches focused on solutions that exploit the human-user context to solve problems in different areas (Shafigh et al., 2019; Zaidi et al., 2019; ZENG et al., 2019). Due to mobile computing expansion, networking-related studies also tend to consider human aspects such as interactions, social ties, and personality to propose human-awareness solutions. This movement from device-to-device to people-to-people communication paradigm aims to look at network configurations taking into account the user's perspective, integrating human perception approaches with QoS metrics, and further, with the mapping of user behavioral profiles. Network contexts are more likely to cope with group-based rather than individual user profiles. Different user profiles, such as personality profiles, may reflect distinct patterns of how users in each profile interact with the network, and consequently, each profile may produce different impacts on the network resource consumption. Therefore, the network can adapt according to the predominant user profiles to improve the distribution/consumption of resources and user QoE at the same time.

In ICN research, human factors present great potentials to improve the communication service delivery, in particular through adaptive caching solutions (Ribeiro; Sampaio; Ziviani, 2018). One approach is to explore potential correlations between user characteristics and cache policies and adopt mechanisms for dynamically adapt the most suitable caching strategies to the predominant user behavior. A key challenging consists of finding out the human aspects that most positively impact the network efficiency and how they could be operatively explored in ICN architectures. That requires a multidisciplinary view with the integration of psychology research to support lower granularity levels of user information.

Privacy:

In-network cache aggregates benefits to ICN architectures by reducing bandwidth consumption and the latency to deliver contents over the network, but it also introduces architectural vulnerabilities regarding cache privacy (ACS et al., 2013). For example, in side-channel timing attacks, a malicious user can deduce what content was accessed recently by another user on the same network by merely measuring content delivery times with standard content requests. Acs et al. (2013) discussed techniques for mitigating privacy caching attacks in which contents marked as private could have different treatments by the cache management mechanism. One countermeasure presented to inhibit the timing attack consists of the insertion of artificial delay times in the content delivery process, so the malicious user cannot differentiate which content was retrieved from the cache or directly from the producer.

Recent efforts from the NDN research community have tried to address many of the current privacy concerns (COMPAGNO et al., 2020; DOGRULUK et al., 2020), but more work lies ahead concerning the context information processed by caching strategies. The use of context information to allow the dynamic adoption of the most appropriate cache policy may require the processing of sensitive data of related users stored in communication devices. One major concern resides in guaranteeing the anonymity of data processed, particularly involving users for privacy-preserving cache management.

Similarly, there is a concern about the privacy of cache management strategies adopted on the network routers. Fan et al. (2020) recently presented a method capable of detecting the placement policy configured in the routers. As described in the malicious attempt to discover the previously accessed content in the network, the method does not require any privileged access and can infer a placement policy through ordinary content requests. Knowing the strategies used for content management can enhance the inference mechanisms of accessed content.

BIBLIOGRAPHY

ABDULLAHI, I.; ARIF, S.; HASSAN, S. Survey on caching approaches in information centric networking. *Journal of Network and Computer Applications*, Elsevier, v. 56, p. 48–59, 2015.

ABIDI, A.; GAMMAR, S. Towards new caching strategy for information-centric networking based on data proximity control. In: IEEE. *Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), 2015 IEEE International Conference on*. [S.l.], 2015. p. 540–547.

ABOWD, G. D.; DEY, A. K.; BROWN, P. J.; DAVIES, N.; SMITH, M.; STEGGLES, P. Towards a better understanding of context and context-awareness. In: SPRINGER. *International symposium on handheld and ubiquitous computing*. [S.l.], 1999. p. 304–307.

ABRAMS, M.; STANDRIDGE, C.; ABDULLA, G.; FOX, E.; WILLIAMS, S. Removal policies in network caches for world-wide web documents. In: . [S.l.: s.n.], 1996. v. 26, p. 293–305.

ACS, G.; CONTI, M.; GASTI, P.; GHALI, C.; TSUDIK, G. Cache privacy in named-data networking. In: IEEE. *2013 IEEE 33rd International Conference on Distributed Computing Systems*. [S.l.], 2013. p. 41–51.

ADRICHEM, N. L. V.; KUIPERS, F. A. Globally accessible names in named data networking. In: IEEE. *2013 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. [S.l.], 2013. p. 345–350.

AFANASYEV, A.; MOISEENKO, I.; ZHANG, L. et al. ndnsim: Ndn simulator for ns-3. *University of California, Los Angeles, Tech. Rep*, v. 4, 2012.

AHLGREN, B.; DANNEWITZ, C.; IMBRENDA, C.; KUTSCHER, D.; OHLMAN, B. A survey of information-centric networking. *IEEE Communications Magazine*, IEEE, v. 50, n. 7, p. 26–36, 2012.

AL-TURJMAN, F. Cognitive caching for the future sensors in fog networking. *Pervasive and Mobile Computing*, Elsevier, v. 42, p. 317–334, 2017.

AL-TURJMAN, F. M.; AL-FAGIH, A. E.; HASSANEIN, H. S. A value-based cache replacement approach for information-centric networks. In: *LCN Workshops*. [S.l.: s.n.], 2013. p. 874–881.

ALEGRE, U.; AUGUSTO, J. C.; CLARK, T. Engineering context-aware systems and applications: A survey. *Journal of Systems and Software*, Elsevier, v. 117, p. 55–83, 2016.

AMADEO, M.; CAMPOLO, C.; MOLINARO, A. Information-centric networking for connected vehicles: a survey and future perspectives. *IEEE Communications Magazine*, IEEE, v. 54, n. 2, p. 98–104, 2016.

AMADEO, M.; CAMPOLO, C.; MOLINARO, A.; RUGGERI, G. Content-centric wireless networking: A survey. *Computer Networks*, Elsevier, v. 72, p. 1–13, 2014.

AN, Y.; LUO, X. An in-network caching scheme based on energy efficiency for content-centric networks. *IEEE Access*, IEEE, v. 6, p. 20184–20194, 2018.

AOKI, M.; SHIGEYASU, T. Effective content management technique based on cooperation cache among neighboring routers in content-centric networking. In: IEEE. *2017 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA)*. [S.l.], 2017. p. 335–340.

ARALDO, A.; ROSSI, D.; MARTIGNON, F. Cost-aware caching: Caching more (costly items) for less (isps operational expenditures). *IEEE Transactions on Parallel and Distributed Systems*, IEEE, v. 27, n. 5, p. 1316–1330, 2015.

ARAUJO, F. R. C.; SOUSA, A. M. de; SAMPAIO, L. N. Armazenamento oportunista em redes de dados nomeados sem fio como suporte à mobilidade de produtores. In: *XXXVI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC) 2018*. Campos do Jordão, SP: [s.n.], 2018.

ARAúJO, F. R. C.; SOUSA, A. M. de; SAMPAIO, L. N. Scan-mob: An opportunistic caching strategy to support producer mobility in named data wireless networking. *Computer Networks*, v. 156, p. 62 – 74, 2019. ISSN 1389-1286. Disponível em: ⟨http://www.sciencedirect.com/science/article/pii/S1389128618311599⟩.

ARLITT, M.; CHERKASOVA, L.; DILLEY, J.; FRIEDRICH, R.; JIN, T. Evaluating content management techniques for web proxy caches. *ACM SIGMETRICS Performance Evaluation Review*, ACM, v. 27, n. 4, p. 3–11, 2000.

ARSHAD, S.; AZAM, M. A.; REHMANI, M. H.; LOO, J. Recent advances in information-centric networking-based internet of things (icn-iot). *IEEE Internet of Things Journal*, IEEE, v. 6, n. 2, p. 2128–2158, 2018.

ARSHADI, L.; JAHANGIR, A. H. Benford's law behavior of internet traffic. *Journal of Network and Computer Applications*, Elsevier, v. 40, p. 194–205, 2014.

BADOV, M.; SEETHARAM, A.; KUROSE, J.; FIROIU, V.; NANDA, S. Congestion-aware caching and search in information-centric networks. In: ACM. *Proceedings of the 1st ACM Conference on Information-Centric Networking*. [S.l.], 2014. p. 37–46.

BALAMASH, A.; KRUNZ, M. An overview of web caching replacement algorithms. *IEEE Communications Surveys & Tutorials*, IEEE, v. 6, n. 2, p. 44–56, 2004.

BARI, M. F.; CHOWDHURY, S. R.; AHMED, R.; BOUTABA, R.; MATHIEU, B. A survey of naming and routing in information-centric networks. *IEEE Communications Magazine*, IEEE, v. 50, n. 12, p. 44–53, 2012.

BAUGH, J. P.; GUO, J. A per-face popularity scheme to increase cache robustness in information-centric networks. *Procedia computer science*, Elsevier, v. 134, p. 267–274, 2018.

BECK, H.; BIERBAUMER, B.; DAO-TRAN, M.; EITER, T.; HELLWAGNER, H.; SCHEKOTIHIN, K. Stream reasoning-based control of caching strategies in ccn routers. In: IEEE. *2017 IEEE International Conference on Communications (ICC)*. [S.l.], 2017. p. 1–6.

BERGER, D. S.; GLAND, P.; SINGLA, S.; CIUCU, F. Exact analysis of ttl cache networks. *Performance Evaluation*, Elsevier, v. 79, p. 2–23, 2014.

BERNARDINI, C.; SILVERSTON, T.; FESTOR, O. Socially-aware caching strategy for content centric networking. In: IEEE. *Networking Conference, 2014 IFIP*. [S.l.], 2014. p. 1–9.

BETTINI, C.; BRDICZKA, O.; HENRICKSEN, K.; INDULSKA, J.; NICKLAS, D.; RANGANATHAN, A.; RIBONI, D. A survey of context modelling and reasoning techniques. *Pervasive and Mobile Computing*, Elsevier, v. 6, n. 2, p. 161–180, 2010.

BILAL, M.; KANG, S.-G. Time aware least recent used (tlru) cache management policy in icn. In: IEEE. *Advanced Communication Technology (ICACT), 2014 16th International Conference on*. [S.l.], 2014. p. 528–532.

BILAL, M.; KANG, S.-G. A cache management scheme for efficient content eviction and replication in cache networks. *IEEE Access*, IEEE, v. 5, p. 1692–1701, 2017.

BITAGHSIR, S. A.; DADLANI, A.; BORHANI, M.; KHONSARI, A. Multi-armed bandit learning for cache content placement in vehicular social networks. *IEEE Communications Letters*, IEEE, v. 23, n. 12, p. 2321–2324, 2019.

BLASCO, P.; GÜNDÜZ, D. Learning-based optimization of cache content in a small cell base station. In: IEEE. *2014 IEEE International Conference on Communications (ICC)*. [S.l.], 2014. p. 1897–1903.

BUCHHOLZ, T.; KÜPPER, A.; SCHIFFERS, M. Quality of context: What it is and why we need it. In: *Workshop of the HP OpenView University Association*. [S.l.: s.n.], 2003.

CAARLS, W.; HARGREAVES, E.; MENASCHÉ, D. S. Q-caching: an integrated reinforcement-learning approach for caching and routing in information-centric networks. *arXiv preprint arXiv:1512.08469*, 2015.

CHAI, W. K.; HE, D.; PSARAS, I.; PAVLOU, G. Cache "less for more" in information-centric networks. In: SPRINGER. *International Conference on Research in Networking*. [S.l.], 2012. p. 27–40.

CHANDRASEKARAN, G.; WANG, N.; HASSANPOUR, M.; XU, M.; TAFAZOLLI, R. Mobility as a service (maas): A d2d-based information centric network architecture for edge-controlled content distribution. *IEEE Access*, IEEE, v. 6, p. 2110–2129, 2018.

CHANDRASEKARAN, G.; WANG, N.; TAFAZOLLI, R. Caching on the move: Towards d2d-based information centric networking for mobile content distribution. In: IEEE. *Local Computer Networks (LCN), 2015 IEEE 40th Conference on*. [S.l.], 2015. p. 312–320.

CHAO, F.; HUANG, T.; JIANG, L.; CHEN, J.-y.; LIU, Y.-j. Fast convergence caching replacement algorithm based on dynamic classification for content-centric networks. *The Journal of China Universities of Posts and Telecommunications*, Elsevier, v. 20, n. 5, p. 45–50, 2013.

CHARPINEL, S.; SANTOS, C. A. S.; VIEIRA, A. B.; VILLACA, R.; MARTINELLO, M. Sdccn: A novel software defined content-centric networking approach. In: IEEE. *2016 IEEE 30Th international conference on advanced information networking and applications (AINA)*. [S.l.], 2016. p. 87–94.

CHE, H.; TUNG, Y.; WANG, Z. Hierarchical web caching systems: Modeling, design and experimental results. *IEEE Journal on Selected Areas in Communications*, IEEE, v. 20, n. 7, p. 1305–1314, 2002.

CHEN, B.; LIU, L.; ZHANG, Z.; YANG, W.; MA, H. Brr-cvr: A collaborative caching strategy for information-centric wireless sensor networks. In: IEEE. *Mobile Ad-Hoc and Sensor Networks (MSN), 2016 12th International Conference on*. [S.l.], 2016. p. 31–38.

CHEN, M.; MAU, D. O.; ZHANG, Y.; TALEB, T.; LEUNG, V. C. Vendnet: Vehicular named data network. *Vehicular Communications*, Elsevier, v. 1, n. 4, p. 208–213, 2014.

CHEN, T.; DU, Y.; DONG, B.; CHEN, Y.; ZHANG, C. Multi-objective learning approach to proactive caching in wireless networks. *IEEE Communications Letters*, IEEE, v. 23, n. 9, p. 1538–1541, 2019.

CHEN, T.; LI, H.; AN, J.; WANG, Y.; SUN, J.; WANG, H. The improved multi-attribute caching: An efficient cache strategy in ccn. In: IEEE. *2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN)*. [S.l.], 2017. p. 1007–1011.

CHEN, X.; FAN, Q.; YIN, H. Caching in information-centric networking: From a content delivery path perspective. In: IEEE. *Innovations in Information Technology (IIT), 2013 9th International Conference on*. [S.l.], 2013. p. 48–53.

CHEN, X.; SHANG, C.; WONG, B.; LI, W.; OH, S. Efficient multicast algorithms in opportunistic mobile social networks using community and social features. *Computer Networks*, Elsevier, v. 111, p. 71–81, 2016.

CHOOTONG, S.; THAENTHONG, J. Cache replacement mechanism with content popularity for vehicular content-centric networks (vccn). In: IEEE. *2017 14th International Joint Conference on Computer Science and Software Engineering (JCSSE)*. [S.l.], 2017. p. 1–6.

CHORLEY, M. J.; COLOMBO, G. B.; ALLEN, S. M.; WHITAKER, R. M. Visiting patterns and personality of foursquare users. In: IEEE. *2013 International Conference on Cloud and Green Computing*. [S.l.], 2013. p. 271–276.

COMPAGNO, A.; CONTI, M.; LOSIOUK, E.; TSUDIK, G.; VALLE, S. A proactive cache privacy attack on ndn. In: IEEE. *NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium*. [S.l.], 2020. p. 1–7.

CONTI, M.; BOLDRINI, C.; KANHERE, S. S.; MINGOZZI, E.; PAGANI, E.; RUIZ, P. M.; YOUNIS, M. From manet to people-centric networking: Milestones and open research challenges. *Computer Communications*, Elsevier, v. 71, p. 1–21, 2015.

COOPERATION, D. E. *Digital's web proxy traces*. September, 1996. Disponível em: ⟨ftp://ftp.digital.com/pub/DEC/traces/proxy/webtraces.html⟩.

COSTA, R. L.; VIANA, A. C.; ZIVIANI, A.; SAMPAIO, L. N. Humanos no ciclo de comunica ção: facilitadores das redes de próxima gera ção. *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)-Minicursos*, 2018.

CUNHA, C. R.; BESTAVROS, A.; CROVELLA, M. E. *Characteristics of WWW client-based traces*. [S.l.], 1995.

DAI, C.; ZHU, K.; WANG, R.; CHEN, B. Contextual multi-armed bandit for cache-aware decoupled multiple association in udns: a deep learning approach. *IEEE Transactions on Cognitive Communications and Networking*, IEEE, v. 5, n. 4, p. 1046–1059, 2019.

DAI, H.; LIU, B.; YUAN, H.; CROWLEY, P.; LU, J. Analysis of tandem pit and cs with non-zero download delay. In: IEEE. *INFOCOM 2017-IEEE Conference on Computer Communications, IEEE*. [S.l.], 2017. p. 1–9.

de Sousa, A. M.; Araújo, F. R. C.; Sampaio, L. N. A link-stability-based interest-forwarding strategy for vehicular named data networks. *IEEE Internet Computing*, v. 22, n. 3, p. 16–26, May 2018. ISSN 1941-0131.

DEY, A. K. Understanding and using context. *Personal and ubiquitous computing*, Springer-Verlag, v. 5, n. 1, p. 4–7, 2001.

DHIAB, I.; BAROUNI, Y.; KHALFALLAH, S.; SLAMA, J. B. H. Pseudo-lru replacement policy in named data networking using fat tree datacenter network architecture. In: IEEE. *2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA)*. [S.l.], 2017. p. 682–687.

DIN, I. U.; HASSAN, S.; ALMOGREN, A.; AYUB, F.; GUIZANI, M. Puc: Packet update caching for energy efficient iot-based information-centric networking. *Future Generation Computer Systems*, Elsevier, 2019.

DIN, I. U.; HASSAN, S.; KHAN, M. K.; GUIZANI, M.; GHAZALI, O.; HABBAL, A. Caching in information-centric networking: Strategies, challenges, and future research directions. *IEEE Communications Surveys & Tutorials*, IEEE, v. 20, n. 2, p. 1443–1474, 2017.

DOGRULUK, E.; GAMA, Ó.; COSTA, A. D.; MACEDO, J. Public key certificate privacy in vondn: Voice over named data networks. *IEEE Access*, IEEE, v. 8, p. 145803–145823, 2020.

DONG, L.; WANG, G. Enhanced in-network capabilities of information-centric networks for emerging iot applications. In: IEEE. *2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. [S.l.], 2016. p. 573–577.

DRON, W.; LEUNG, A.; UDDIN, M.; WANG, S.; ABDELZAHER, T.; GOVINDAN, R.; HANCOCK, J. Information-maximizing caching in ad hoc networks with named data networking. In: IEEE. *2013 IEEE 2nd Network Science Workshop (NSW)*. [S.l.], 2013. p. 90–93.

DRäXLER, M.; KARL, H. Efficiency of on-path and off-path caching strategies in information centric networks. In: IEEE. *2012 IEEE International Conference on Green Computing and Communications*. [S.l.], 2012. p. 581–587.

DUAN, J.; WANG, X.; WANG, S.; XU, S. Profit-based caching for information-centric network. In: IEEE. *Dependable, Autonomic and Secure Computing (DASC), 2013 IEEE 11th International Conference on*. [S.l.], 2013. p. 481–486.

ELBAMBY, M. S.; BENNIS, M.; SAAD, W.; LATVA-AHO, M. Content-aware user clustering and caching in wireless small cell networks. In: IEEE. *2014 11th International Symposium on Wireless Communications Systems (ISWCS)*. [S.l.], 2014. p. 945–949.

ENGELENBURG, S. van; JANSSEN, M.; KLIEVINK, B. Designing context-aware systems: A method for understanding and analysing context in practice. *Journal of logical and algebraic methods in programming*, Elsevier, v. 103, p. 79–104, 2019.

EUM, S.; NAKAUCHI, K.; MURATA, M.; SHOJI, Y.; NISHINAGA, N. Catt: potential based routing with content caching for icn. In: *Proceedings of the second edition of the ICN workshop on Information-centric networking*. [S.l.: s.n.], 2012. p. 49–54.

EUM, S.; SHOJI, Y.; MURATA, M.; NISHINAGA, N. Design and implementation of icn-enabled ieee 802.11 access points as nano data centers. *Journal of Network and Computer Applications*, Elsevier, v. 50, p. 159–167, 2015.

FAN, C.; SHANNIGRAHI, S.; PAPADOPOULOS, C.; PARTRIDGE, C. Discovering in-network caching policies in ndn networks from a measurement perspective. In: *Proceedings of the 7th ACM Conference on Information-Centric Networking*. [S.l.: s.n.], 2020. p. 106–116.

FANG, C.; YU, F. R.; HUANG, T.; LIU, J.; LIU, Y. A survey of energy-efficient caching in information-centric networking. *IEEE Communications Magazine*, IEEE, v. 52, n. 11, p. 122–129, 2014.

FANG, C.; YU, F. R.; HUANG, T.; LIU, J.; LIU, Y. A survey of green information-centric networking: Research issues and challenges. *IEEE Communications Surveys & Tutorials*, IEEE, v. 17, n. 3, p. 1455–1472, 2015.

FRICKER, C.; ROBERT, P.; ROBERTS, J. A versatile and accurate approximation for lru cache performance. In: IEEE. *2012 24th International Teletraffic Congress (ITC 24)*. [S.l.], 2012. p. 1–8.

FRICKER, C.; ROBERT, P.; ROBERTS, J.; SBIHI, N. Impact of traffic mix on caching performance in a content-centric network. In: IEEE. *2012 Proceedings IEEE INFOCOM Workshops*. [S.l.], 2012. p. 310–315.

GALLO, M.; KAUFFMANN, B.; MUSCARIELLO, L.; SIMONIAN, A.; TANGUY, C. Performance evaluation of the random replacement policy for networks of caches. In: *Proceedings of the 12th ACM SIGMETRICS/PERFORMANCE Joint International Conference on Measurement and Modeling of Computer Systems*. New York, NY, USA: ACM, 2012. (SIGMETRICS '12), p. 395–396. ISBN 978-1-4503-1097-0. Disponível em: ⟨http://doi-acm-org.ez10.periodicos.capes.gov.br/10.1145/2254756.2254810⟩.

GALLO, M.; KAUFFMANN, B.; MUSCARIELLO, L.; SIMONIAN, A.; TANGUY, C. Performance evaluation of the random replacement policy for networks of caches. *Performance Evaluation*, Elsevier, v. 72, p. 16–36, 2014.

GARCÍA, G.; BEBEN, A.; RAMÓN, F. J.; MAESO, A.; PSARAS, I.; PAVLOU, G.; WANG, N.; ŚLIWIŃSKI, J.; SPIROU, S.; SOURSOS, S.; HADJIOANNOU, E. Comet: Content mediator architecture for content-aware networks. In: IEEE. *2011 Future Network & Mobile Summit*. [S.l.], 2011. p. 1–8.

GARETTO, M.; LEONARDI, E.; MARTINA, V. A unified approach to the performance analysis of caching systems. *ACM Transactions on Modeling and Performance Evaluation of Computing Systems*, ACM, v. 1, n. 3, p. 12, 2016.

GARIVIER, A.; MOULINES, E. On upper-confidence bound policies for switching bandit problems. In: SPRINGER. *International Conference on Algorithmic Learning Theory*. [S.l.], 2011. p. 174–188.

GAWŁOWICZ, P.; ZUBOW, A. Ns-3 meets openai gym: The playground for machine learning in networking research. In: *Proceedings of the 22nd International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*. [S.l.: s.n.], 2019. p. 113–120.

GE, C.; WANG, N.; SKILLMAN, S.; FOSTER, G.; CAO, Y. Qoe-driven dash video caching and adaptation at 5g mobile edge. In: ACM. *Proceedings of the 3rd ACM Conference on Information-Centric Networking*. [S.l.], 2016. p. 237–242.

GHAHFAROKHI, B. S.; MOGHIM, N.; EFTEKHARI, S. Reducing channel zapping time in live tv broadcasting over content centric networks. *Multimedia Tools and Applications*, Springer, v. 76, n. 22, p. 23239–23271, 2017.

GOMAA, H.; MESSIER, G. G.; WILLIAMSON, C.; DAVIES, R. Estimating instantaneous cache hit ratio using markov chain analysis. *IEEE/ACM transactions on Networking*, IEEE, v. 21, n. 5, p. 1472–1483, 2013.

GÜR, G. Energy-aware cache management at the wireless network edge for information-centric operation. *Journal of Network and Computer Applications*, Elsevier, v. 57, p. 33–42, 2015.

GÜR, G.; PORAMBAGE, P.; LIYANAGE, M. Convergence of icn and mec for 5g: Opportunities and challenges. *IEEE Communications Standards Magazine*, IEEE, v. 4, n. 4, p. 64–71, 2020.

HAHM, O.; BACCELLI, E.; SCHMIDT, T.; W¨AHLISCH, M.; ADJIH, C. A named data network approach to energy efficiency in iot. In: *IEEE GLOBECOM Workshop on Named Data Networking for Challenged Communication Environments*. [S.l.: s.n.], 2016.

HAN, B.; WANG, X.; CHEN, X.; KWON, T. T.; CHOI, Y.; MAU, D. O.; CHEN, M. Ppp: prefix-based popularity prediction for effective caching in content-centric networking. In: ACM. *Proceedings of The Ninth International Conference on Future Internet Technologies*. [S.l.], 2014. p. 7.

HE, S.; WANG, T.; WANG, S. Mobility-driven user-centric ap clustering in mobile edge computing based ultra dense networks. *Digital Communications and Networks*, Elsevier, 2019.

HOU, R.; ZHANG, L.; WU, T.; MAO, T.; LUO, J. Bloom-filter-based request node collaboration caching for named data networking. *Cluster Computing*, Springer, v. 22, n. 3, p. 6681–6692, 2019.

HU, X.; GONG, J.; CHENG, G.; FAN, C. Enhancing in-network caching by coupling cache placement, replacement and location. In: IEEE. *Communications (ICC), 2015 IEEE International Conference on*. [S.l.], 2015. p. 5672–5678.

HUANG, L.; GUAN, Y.; ZHANG, X.; GUO, Z. On-path collaborative in-network caching for information-centric networks. In: IEEE. *2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. [S.l.], 2017. p. 175–180.

HUANG, S.; ZENG, Z.; OTA, K.; DONG, M.; WANG, T.; XIONG, N. An intelligent collaboration trust interconnections system for mobile information control in ubiquitous 5g networks. *IEEE Transactions on Network Science and Engineering*, IEEE, 2020.

HUANG, W.; SONG, T.; YANG, Y.; ZHANG, Y. Cluster-based selective cooperative caching strategy in vehicular named data networking. In: IEEE. *2018 1st IEEE International Conference on Hot Information-Centric Networking (HotICN)*. [S.l.], 2018. p. 7–12.

IOANNIDIS, S.; YEH, E. Adaptive caching networks with optimality guarantees. In: ACM. *ACM SIGMETRICS Performance Evaluation Review*. [S.l.], 2016. v. 44, n. 1, p. 113–124.

IOANNIDIS, S.; YEH, E. Adaptive caching networks with optimality guarantees. *IEEE/ACM Transactions on Networking*, IEEE, v. 26, n. 2, p. 737–750, 2018.

IOANNOU, A.; WEBER, S. A survey of caching policies and forwarding mechanisms in information-centric networking. *IEEE Communications Surveys & Tutorials*, IEEE, v. 18, n. 4, p. 2847–2886, 2016.

JEDARI, B.; PREMSANKAR, G.; ILLAHI, G.; FRANCESCO, M. D.; MEHRABI, A.; YLÄ-JÄÄSKI, A. Video caching, analytics, and delivery at the wireless edge: A survey and future directions. *IEEE Communications Surveys & Tutorials*, IEEE, v. 23, n. 1, p. 431–471, 2020.

JEON, H.; LEE, B.; SONG, H. On-path caching in information-centric networking. In: IEEE. *2013 15th International Conference on Advanced Communications Technology (ICACT)*. [S.l.], 2013. p. 264–267.

JI, J.; ZHU, K.; NIYATO, D.; WANG, R. Probabilistic cache placement in uav-assisted networks with d2d connections: Performance analysis and trajectory optimization. *IEEE Transactions on Communications*, IEEE, v. 68, n. 10, p. 6331–6345, 2020.

JIA, R.; LIU, Z.; WANG, X.; GAN, X.; WANG, X.; XU, J. J. Modeling dynamic adaptive streaming over information-centric networking. *IEEE Access*, IEEE, v. 4, p. 8362–8374, 2016.

JIANG, S.; ZHANG, X. F. Lirs: An efficient low inter-reference recency set replacement to improve buffer cache performance. In: . [S.l.: s.n.], 2002. v. 30, p. 31–42.

JIN, H.; XU, D.; ZHAO, C.; LIANG, D. Information-centric mobile caching network frameworks and caching optimization: a survey. *EURASIP Journal on Wireless Communications and Networking*, Nature Publishing Group, v. 2017, n. 1, p. 33, 2017.

JIN, S.; BESTAVROS, A. Temporal locality in web request streams-sources, characteristics, and caching implications. In: CITESEER. *In Proceedings of SIGMETRICS*. [S.l.], 1999.

JOHNSON, T.; SHASHA, D. 2q: A low overhead high performance buffer management replacement algoritm. In: *Proceedings of the Twentieth International Conference on Very Large Databases, Santiago, Chile*. [S.l.: s.n.], 1994. p. 439–450.

KALGHOUM, A.; GAMMAR, S. M. Towards new information centric networking strategy based on software defined networking. In: IEEE. *Wireless Communications and Networking Conference (WCNC), 2017 IEEE*. [S.l.], 2017. p. 1–6.

KALGHOUM, A.; GAMMAR, S. M.; SAIDANE, L. A. Towards a novel cache replacement strategy for named data networking based on software defined networking. *Computers & Electrical Engineering*, Elsevier, v. 66, p. 98–113, 2018.

KALGHOUM, A.; SAIDANE, L. A. Fcr-ns: a novel caching and forwarding strategy for named data networking based on software defined networking. *Cluster Computing*, Springer, p. 1–14, 2019.

KANG, S.-J.; LEE, S.-W.; KO, Y.-B. A recent popularity based dynamic cache management for content centric networking. In: IEEE. *Ubiquitous and Future Networks (ICUFN), 2012 Fourth International Conference on*. [S.l.], 2012. p. 219–224.

KARAMI, A.; GUERRERO-ZAPATA, M. An anfis-based cache replacement method for mitigating cache pollution attacks in named data networking. *Computer Networks*, Elsevier, v. 80, p. 51–65, 2015.

KHAN, F. H.; KHAN, Z. Popularity-aware content caching for distributed wireless helper nodes. *Arabian Journal for Science and Engineering*, Springer, v. 42, n. 8, p. 3375–3389, 2017.

KHAN, J.; WESTPHAL, C.; GARCIA-LUNA-ACEVES, J.; GHAMRI-DOUDANE, Y. Nice: Network-oriented information-centric centrality for efficiency in cache management. In: ACM. *ICN '18: Proceedings of the 5th ACM Conference on Information-Centric Networking*. [S.l.], 2018. p. 31–42.

KHELIFI, H.; LUO, S.; NOUR, B.; MOUNGLA, H.; FAHEEM, Y.; HUSSAIN, R.; KSENTINI, A. Named data networking in vehicular ad hoc networks: State-of-the-art and challenges. *IEEE Communications Surveys & Tutorials*, IEEE, 2020.

KIM, W.-S.; CHUNG, S.-H.; MOON, J.-W. Improved content management for information-centric networking in sdn-based wireless mesh network. *Computer Networks*, Elsevier, v. 92, p. 316–329, 2015.

KITCHENHAM, B.; CHARTERS, S. *Guidelines for performing systematic literature reviews in software engineering*. [S.l.], 2007.

KLAINE, P. V.; IMRAN, M. A.; ONIRETI, O.; SOUZA, R. D. A Survey of Machine Learning Techniques Applied to Self-Organizing Cellular Networks. *IEEE Communications Surveys & Tutorials*, IEEE, v. 19, n. 4, p. 2392–2431, 2017.

KOPONEN, T.; CHAWLA, M.; CHUN, B.-G.; ERMOLINSKIY, A.; KIM, K. H.; SHENKER, S.; STOICA, I. A data-oriented (and beyond) network architecture. In: *Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications*. [S.l.: s.n.], 2007. p. 181–192.

LAL, K. N.; KUMAR, A. A cache content replacement scheme for information centric network. *Procedia Computer Science*, Elsevier, v. 89, p. 73–81, 2016.

LAL, K. N.; KUMAR, A. A popularity based content eviction scheme via betweenness-centrality caching approach for content-centric networking (ccn). *Wireless Networks*, Springer, v. 25, n. 2, p. 585–596, 2019.

LAOUTARIS, N.; SYNTILA, S.; STAVRAKAKIS, I. Meta algorithms for hierarchical web caches. In: IEEE. *IEEE International Conference on Performance, Computing, and Communications, 2004*. [S.l.], 2004. p. 445–452.

LEE, D.; CHOI, J.; KIM, J.-H.; NOH, S. H.; MIN, S. L.; CHO, Y.; KIM, C. S. Lrfu: A spectrum of policies that subsumes the least recently used and least frequently used policies. *IEEE transactions on Computers*, IEEE, n. 12, p. 1352–1361, 2001.

LEE, J.; LIM, K.; YOO, C. Cache replacement strategies for scalable video streaming in ccn. In: IEEE. *2013 19th Asia-Pacific Conference on Communications (APCC)*. [S.l.], 2013. p. 184–189.

LEE, J. W.; HONG, C. S. Fera: A caching scheme in ccn using file-extension and regression analysis. In: IEEE. *Network Operations and Management Symposium (APNOMS), 2017 19th Asia-Pacific*. [S.l.], 2017. p. 303–306.

LEHMANN, M. B.; BARCELLOS, M. P.; MAUTHE, A. Providing producer mobility support in NDN through proactive data replication. In: *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*. [S.l.]: IEEE, 2016. p. 383–391.

LI, B.; MA, M.; HU, S. Perceptive forwarding strategy in content-centric network. In: IEEE. *Globecom Workshops (GC Wkshps), 2015 IEEE*. [S.l.], 2015. p. 1–6.

LI, H.; NAKAZATO, H.; DETTI, A.; MELAZZI, N. B. Popularity proportional cache size allocation policy for video delivery on ccn. In: IEEE. *Networks and Communications (EuCNC), 2015 European Conference on*. [S.l.], 2015. p. 434–438.

LI, J.; LIU, B.; WU, H. Energy-efficient in-network caching for content-centric networking. *IEEE Communications Letters*, IEEE, v. 17, n. 4, p. 797–800, 2013.

LI, J.; LUO, H.; ZHANG, S.; LI, H.; YAN, F. Design and implementation of efficient control for incoming inter-domain traffic with information-centric networking. *Journal of Network and Computer Applications*, Elsevier, v. 133, p. 109–125, 2019.

LI, J.; SHAKKOTTAI, S.; LUI, J. C.; SUBRAMANIAN, V. Accurate learning or fast mixing? dynamic adaptability of caching algorithms. *IEEE Journal on Selected Areas in Communications*, IEEE, v. 36, n. 6, p. 1314–1330, 2018.

LI, T.; LIU, A.; XIONG, N.; ZHANG, S.; WANG, T. A trustworthiness-based vehicular recruitment scheme for information collections in distributed networked systems. *Information Sciences*, v. 545, 08 2020.

LI, T.; ZHAO, M.; WONG, K. Machine learning based code dissemination by selection of reliability mobile vehicles in 5g networks. *Computer Communications*, v. 152, 02 2020.

LI, Y.; YU, M.; LI, R. A cache replacement strategy based on hierarchical popularity in ndn. In: IEEE. *2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN)*. [S.l.], 2018. p. 159–161.

LI, Z.; SIMON, G.; GRAVEY, A. Caching policies for in-network caching. In: IEEE. *2012 21st International conference on computer communications and networks (ICCCN)*. [S.l.], 2012. p. 1–7.

LIANG, Y.; ZHOU, X.; GUO, B.; YU, Z. Understanding the regularity and variability of human mobility from geo-trajectory. In: IEEE. *2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*. [S.l.], 2012. v. 1, p. 409–414.

LIU, H.; AZHANDEH, K.; FOY, X. de; GAZDA, R. A comparative study of name resolution and routing mechanisms in information-centric networks. *Digital Communications and Networks*, Elsevier, v. 5, n. 2, p. 69–75, 2019.

LIU, W.; LI, X.; HUANG, D. A survey on context awareness. In: IEEE. *2011 International Conference on Computer Science and Service System (CSSS)*. [S.l.], 2011. p. 144–147.

LIU, W.-X.; ZHANG, J.; LIANG, Z.-W.; PENG, L.-X.; CAI, J. Content popularity prediction and caching for icn: A deep learning approach with sdn. *IEEE access*, IEEE, v. 6, p. 5075–5089, 2018.

LIU, Y.; ZHI, T.; XI, H.; QUAN, W.; ZHANG, H. A novel cache replacement scheme against cache pollution attack in content-centric networks. In: IEEE. *2019 IEEE/CIC International Conference on Communications in China (ICCC)*. [S.l.], 2019. p. 207–212.

LIU, Y.; ZHU, D.; MA, W. A novel cooperative caching scheme for content centric mobile ad hoc networks. In: IEEE. *Computers and Communication (ISCC), 2016 IEEE Symposium on*. [S.l.], 2016. p. 824–829.

LIU, Z.; DONG, M.; GU, B.; ZHANG, C.; JI, Y.; TANAKA, Y. Fast-start video delivery in future internet architectures with intra-domain caching. *Mobile Networks and Applications*, Springer, v. 22, n. 1, p. 98–112, 2017.

LLORCA, J.; TULINO, A. M.; VARVELLO, M.; ESTEBAN, J. O.; PERINO, D. Energy efficient dynamic content distribution. *IEEE Journal on Selected Areas in Communications*, v. 33, n. 12, p. 2826–2836, 2015.

MADUREIRA, A. L. R.; ARAÚJO, F. R. C.; ARAÚJO, G. B.; SAMPAIO, L. N. Ndn fabric: Where the software-defined networking meets the content-centric model. *IEEE Transactions on Network and Service Management*, IEEE, 2020.

MANGILI, M.; MARTIGNON, F.; CAPONE, A. A comparative study of content-centric and content-distribution networks: Performance and bounds. In: IEEE. *2013 IEEE Global Communications Conference (GLOBECOM)*. [S.l.], 2013. p. 1403–1409.

MAO, Y.; YOU, C.; ZHANG, J.; HUANG, K.; LETAIEF, K. B. A survey on mobile edge computing: The communication perspective. *IEEE Communications Surveys & Tutorials*, IEEE, v. 19, n. 4, p. 2322–2358, 2017.

MASTORAKIS, S.; AFANASYEV, A.; MOISEENKO, I.; ZHANG, L. *ndnSIM 2: An updated NDN simulator for NS-3*. [S.l.], 2016.

MASTORAKIS, S.; AFANASYEV, A.; ZHANG, L. On the evolution of ndnSIM: an open-source simulator for NDN experimentation. *ACM Computer Communication Review*, jul. 2017.

MEDDEB, M.; DHRAIEF, A.; BELGHITH, A.; MONTEIL, T.; DRIRA, K. How to cache in icn-based iot environments? In: IEEE. *2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA)*. [S.l.], 2017. p. 1117–1124.

MEDDEB, M.; DHRAIEF, A.; BELGHITH, A.; MONTEIL, T.; DRIRA, K.; MATHKOUR, H. Least fresh first cache replacement policy for ndn-based iot networks. *Pervasive and Mobile Computing*, Elsevier, v. 52, p. 60–70, 2019.

MEGIDDO, N.; MODHA, D. S. Arc: A self-tuning, low overhead replacement cache. In: *Fast*. [S.l.: s.n.], 2003. v. 3, n. 2003, p. 115–130.

MEGIDDO, N.; MODHA, D. S. Outperforming lru with an adaptive replacement cache algorithm. *Computer*, IEEE, v. 37, n. 4, p. 58–65, 2004.

MELAZZI, N. B.; BIANCHI, G.; CAPONI, A.; DETTI, A. A general, tractable and accurate model for a cascade of lru caches. *IEEE Communications Letters*, IEEE, v. 18, n. 5, p. 877–880, 2014.

MICK, T.; TOURANI, R.; MISRA, S. Muncc: Multi-hop neighborhood collaborative caching in information centric networks. In: ACM. *Proceedings of the 3rd ACM Conference on Information-Centric Networking*. [S.l.], 2016. p. 93–101.

MING, Z.; XU, M.; WANG, D. Age-based cooperative caching in information-centric networks. In: IEEE. *2012 Proceedings IEEE INFOCOM Workshops*. [S.l.], 2012. p. 268–273.

MOON, C.; HAN, S.; WOO, H.; KIM, D. Named data networking for infrastructure wireless networks. In: IEEE. *2016 IEEE International Conference on Consumer Electronics (ICCE)*. [S.l.], 2016. p. 343–344.

MÜLLER, S.; ATAN, O.; SCHAAR, M. van der; KLEIN, A. Smart caching in wireless small cell networks via contextual multi-armed bandits. In: IEEE. *2016 IEEE International Conference on Communications (ICC)*. [S.l.], 2016. p. 1–7.

NAKAYAMA, H.; ATA, S.; OKA, I. Caching algorithm for content-oriented networks using prediction of popularity of contents. In: IEEE. *Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on*. [S.l.], 2015. p. 1171–1176.

NAZ, S.; RAIS, R. N. B.; QAYYUM, A. Multi-attribute caching: Towards efficient cache management in content-centric networks. In: IEEE. *2016 13th IEEE Annual Consumer Communications & Networking Conference (CCNC)*. [S.l.], 2016. p. 630–633.

NDIKUMANA, A.; TRAN, N. H.; HO, T. M.; NIYATO, D.; HAN, Z.; HONG, C. S. Joint incentive mechanism for paid content caching and price based cache replacement policy in named data networking. *IEEE Access*, IEEE, v. 6, p. 33702–33717, 2018.

NEVES, M.; RODRIGUES, M.; AZEVÊDO, E.; SADOK, D.; CALLADO, A.; MOREIRA, J.; SOUZA, V. Selecting the most suited cache strategy for specific streaming media workloads. In: IEEE. *Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on*. [S.l.], 2013. p. 792–795.

NEWBERRY, E.; ZHANG, B. On the power of in-network caching in the hadoop distributed file system. In: *Proceedings of the 6th ACM Conference on Information-Centric Networking*. [S.l.: s.n.], 2019. p. 89–99.

NOUR, B.; LI, F.; KHELIFI, H.; MOUNGLA, H.; KSENTINI, A. Coexistence of icn and ip networks: An nfv as a service approach. In: IEEE. *2019 IEEE Global Communications Conference (GLOBECOM)*. [S.l.], 2019. p. 1–6.

OLIVEIRA, E. M. R.; VIANA, A. C.; SARRAUTE, C.; BREA, J.; ALVAREZ-HAMELIN, I. On the regularity of human mobility. *Pervasive and Mobile Computing*, Elsevier, v. 33, p. 73–90, 2016.

O'NEIL, E. J.; O'NEIL, P. E.; WEIKUM, G. The lru-k page replacement algorithm for database disk buffering. *Acm Sigmod Record*, ACM, v. 22, n. 2, p. 297–306, 1993.

ONG, M. D.; CHEN, M.; TALEB, T.; WANG, X.; LEUNG, V. Fgpc: Fine-grained popularity-based caching design for content centric networking. In: ACM. *Proceedings of the 17th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems*. [S.l.], 2014. p. 295–302.

OSTROVSKAYA, S.; SURNIN, O.; HUSSAIN, R.; BOUK, S. H.; LEE, J.; MEHRAN, N.; AHMED, S. H.; BENSLIMANE, A. Towards multi-metric cache replacement policies in vehicular named data networks. In: IEEE. *2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. [S.l.], 2018. p. 1–7.

PACIFICI, V.; DÁN, G. Content-peering dynamics of autonomous caches in a content-centric network. In: IEEE. *2013 Proceedings IEEE INFOCOM*. [S.l.], 2013. p. 1079–1087.

PAL, A.; KANT, K. Nacid: A neighborhood aware caching and interest dissemination in content centric networks. In: IEEE. *2017 26th International Conference on Computer Communication and Networks (ICCCN)*. [S.l.], 2017. p. 1–9.

PAN, T.; HUANG, T.; LI, C. Modeling ccn packet forwarding engine. In: IEEE. *GLOBECOM 2017-2017 IEEE Global Communications Conference*. [S.l.], 2017. p. 1–6.

PAN, T.; LIN, X.; ZHANG, J.; LI, H.; LV, J.; HUANG, T.; LIU, B.; ZHANG, B. Nb-cache: non-blocking in-network caching for high-speed content routers. In: *Proceedings of the International Symposium on Quality of Service*. [S.l.: s.n.], 2019. p. 1–10.

PANDA, P.; PATIL, G.; RAVEENDRAN, B. A survey on replacement strategies in cache memory for embedded systems. In: IEEE. *2016 IEEE Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER)*. [S.l.], 2016. p. 12–17.

Panigrahi, B.; Shailendra, S.; Rath, H. K.; Simha, A. Universal caching model and markov-based cache analysis for information centric networks. In: *2014 IEEE International Conference on Advanced Networks and Telecommuncations Systems (ANTS)*. [S.l.: s.n.], 2014. p. 1–6. ISSN 2153-1676.

PASCHOS, G. S.; DESTOUNIS, A.; VIGNERI, L.; IOSIFIDIS, G. Learning to cache with no regrets. In: IEEE. *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. [S.l.], 2019. p. 235–243.

PENTIKOUSIS, K.; OHLMAN, B.; CORUJO, D.; BOGGIA, G.; TYSON, G.; DAVIES, E. B.; MOLINARO, A.; EUM, S. *Information-Centric Networking: Baseline Scenarios*. RFC Editor, 2015. RFC 7476. (Request for Comments, 7476). Disponível em: ⟨https://rfc-editor.org/rfc/rfc7476.txt⟩.

PERERA, C.; ZASLAVSKY, A.; CHRISTEN, P.; GEORGAKOPOULOS, D. Context aware computing for the internet of things: A survey. *IEEE communications surveys & tutorials*, IEEE, v. 16, n. 1, p. 414–454, 2013.

PETERSEN, K.; FELDT, R.; MUJTABA, S.; MATTSSON, M. Systematic mapping studies in software engineering. In: *Ease*. [S.l.: s.n.], 2008. v. 8, p. 68–77.

PIETRONERO, L.; TOSATTI, E.; TOSATTI, V.; VESPIGNANI, A. Explaining the uneven distribution of numbers in nature: the laws of benford and zipf. *Physica A: Statistical Mechanics and its Applications*, Elsevier, v. 293, n. 1-2, p. 297–304, 2001.

PIRES, S. S.; RIBEIRO, A. V.; SOUSA, A. M. de; FREITAS, A. E.; SAMPAIO, L. N. On evaluating the influence of user's music listening habits on cache replacement policies. In: IEEE. *2018 IEEE Symposium on Computers and Communications (ISCC)*. [S.l.], 2018. p. 00930–00933.

PODLIPNIG, S.; BÖSZÖRMENYI, L. A survey of web cache replacement strategies. *ACM Computing Surveys (CSUR)*, ACM, v. 35, n. 4, p. 374–398, 2003.

PRAKASH, R. S.; MOHARIR, S. Poster: Caching static and transient data. In: ACM. *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*. [S.l.], 2018. p. 678–680.

PSARAS, I.; ASCIGIL, O.; RENE, S.; PAVLOU, G.; AFANASYEV, A.; ZHANG, L. Mobile data repositories at the edge. In: *{USENIX} Workshop on Hot Topics in Edge Computing (HotEdge 18)*. [S.l.: s.n.], 2018.

PSARAS, I.; CHAI, W. K.; PAVLOU, G. Probabilistic in-network caching for information-centric networks. In: *Proceedings of the second edition of the ICN workshop on Information-centric networking*. [S.l.: s.n.], 2012. p. 55–60.

QIAN, H.; MUQING, W.; DONGYANG, W.; SONG, G. Lifetime-based greedy caching approach for content-centric networking. In: IEEE. *Telecommunications (ICT), 2014 21st International Conference on*. [S.l.], 2014. p. 426–430.

QU, D.; WANG, X.; HUANG, M.; LI, K.; DAS, S. K.; WU, S. A cache-aware social-based qos routing scheme in information centric networks. *Journal of Network and Computer Applications*, Elsevier, v. 121, p. 20–32, 2018.

QUEVEDO, J.; CORUJO, D.; AGUIAR, R. A case for icn usage in iot environments. In: IEEE. *2014 IEEE Global Communications Conference*. [S.l.], 2014. p. 2770–2775.

RAHMAN, A.; TROSSEN, D.; KUTSCHER, D.; RAVINDRAN, R. *Deployment Considerations for Information-Centric Networking (ICN)*. RFC Editor, 2020. RFC 8763. (Request for Comments, 8763). Disponível em: ⟨https://rfc-editor.org/rfc/rfc8763.txt⟩.

RAMASWAMY, S.; RASTOGI, R.; SHIM, K. Efficient algorithms for mining outliers from large data sets. In: ACM. *ACM Sigmod Record*. [S.l.], 2000. v. 29, n. 2, p. 427–438.

RAN, J.; LV, N.; ZHANG, D.; MA, Y. yuan; XIE, Z. yong. On performance of cache policies in named data networking. In: ATLANTIS PRESS. *2013 International Conference on Advanced Computer Science and Electronics Information (ICACSEI 2013)*. [S.l.], 2013.

RAO, A.; SCHELEN, O.; LINDGREN, A. Performance implications for iot over information centric networks. In: ACM. *Proceedings of the Eleventh ACM Workshop on Challenged Networks*. [S.l.], 2016. p. 57–62.

RATH, H. K.; PANIGRAHI, B.; SIMHA, A. On cooperative on-path and off-path caching policy for information centric networks (icn). In: IEEE. *Advanced Information Networking and Applications (AINA), 2016 IEEE 30th International Conference on*. [S.l.], 2016. p. 842–849.

RAVI, A.; RAMANATHAN, P.; SIVALINGAM, K. M. Integrated network coding and caching in information-centric networks. In: IEEE. *Advanced Networks and Telecommuncations Systems (ANTS), 2014 IEEE International Conference on*. [S.l.], 2014. p. 1–6.

RAVINDRAN, R.; SUTHAR, P.; TROSSEN, D.; WANG, C.; WHITE, G. *Enabling icn in 3gpp's 5g nextgen core architecture*. 2021.

RAYCHAUDHURI, D.; NAGARAJA, K.; VENKATARAMANI, A. Mobilityfirst: a robust and trustworthy mobility-centric architecture for the future internet. *ACM SIGMOBILE Mobile Computing and Communications Review*, ACM New York, NY, USA, v. 16, n. 3, p. 2–13, 2012.

REN, J.; QI, W.; WESTPHAL, C.; WANG, J.; LU, K.; LIU, S.; WANG, S. Magic: A distributed max-gain in-network caching strategy in information-centric networks. In: IEEE. *2014 IEEE conference on computer communications workshops (INFOCOM WKSHPS)*. [S.l.], 2014. p. 470–475.

REZAZAD, M.; TAY, Y. Ccndns: A strategy for spreading content and decoupling ndn caches. In: IEEE. *IFIP Networking Conference (IFIP Networking), 2015*. [S.l.], 2015. p. 1–9.

RHAIEM, O. B.; FOURATI, L. C.; AJIB, W. New hierarchical parent-child caching strategy (h-cs) for ccn-based video streaming. In: IEEE. *Advanced Information Networking and Applications (AINA), 2016 IEEE 30th International Conference on*. [S.l.], 2016. p. 109–115.

Ribeiro, A. V.; Sampaio, L. N.; Ziviani, A. Affinity-based user clustering for efficient edge caching in content-centric cellular networks. In: *2018 IEEE Symposium on Computers and Communications (ISCC)*. [S.l.: s.n.], 2018. p. 00474–00479. ISSN 1530-1346.

RILEY, G. F.; HENDERSON, T. R. The ns-3 network simulator. In: *Modeling and tools for network simulation*. [S.l.]: Springer, 2010. p. 15–34.

ROCHA, A. A.; DEHGHAN, M.; SALONIDIS, T.; HE, T.; TOWSLEY, D. Dsca: A data stream caching algorithm. In: ACM. *Proceedings of the 1st Workshop on Content Caching and Delivery in Wireless Networks*. [S.l.], 2016. p. 8.

ROSÁRIO, D.; SERUFFO, M.; CERQUEIRA, E.; BOTH, C.; BRAUN, T.; GERLA, M. Trends in human-centric multimedia networking scenarios. In: IEEE. *2016 Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*. [S.l.], 2016. p. 1–5.

ROSENSWEIG, E. J.; MENASCHÉ, D. S.; KUROSE, J. On the steady-state of cache networks. In: *INFOCOM*. [S.l.: s.n.], 2013. p. 863–871.

ROSSI, D.; ROSSINI, G. Caching performance of content centric networks under multi-path routing (and more). *Relatório técnico, Telecom ParisTech*, p. 1–6, 2011.

SAADEH, H.; ALMOBAIDEEN, W.; SABRI, K. E.; SAADEH, M. Hybrid sdn-icn architecture design for the internet of things. In: IEEE. *2019 Sixth International Conference on Software Defined Systems (SDS)*. [S.l.], 2019. p. 96–101.

SADEGHI, A.; SHEIKHOLESLAMI, F.; GIANNAKIS, G. B. Optimal and scalable caching for 5g using reinforcement learning of space-time popularities. *IEEE Journal of Selected Topics in Signal Processing*, IEEE, v. 12, n. 1, p. 180–190, 2017.

SAFAVAT, S.; NAVEEN, N. N. S.; RAWAT, D. B. Recent advances in mobile edge computing and content caching. *Digital Communications and Networks*, Elsevier, 2019.

SALTARIN, J.; BRAUN, T.; BOURTSOULATZE, E.; THOMOS, N. Popnetcod: A popularity-based caching policy for network coding enabled named data networking. In: IEEE. *2018 IFIP Networking Conference (IFIP Networking) and Workshops*. [S.l.], 2018. p. 271–279.

SANTOS, D.; SILVA, R.; CORUJO, D.; AGUIAR, R. L.; PARREIRA, B. Follow the user: A framework for dynamically placing content using 5g-enablers. *IEEE Access*, IEEE, v. 9, p. 14688–14709, 2021.

SANTOS, F. Neves dos; ERTL, B.; BARAKAT, C.; SPYROPOULOS, T.; TURLETTI, T. Cedo: Content-centric dissemination algorithm for delay-tolerant networks. In: ACM. *Proceedings of the 16th ACM international conference on Modeling, analysis & simulation of wireless and mobile systems*. [S.l.], 2013. p. 377–386.

SAXENA, D.; RAYCHOUDHURY, V.; SURI, N.; BECKER, C.; CAO, J. Named data networking: a survey. *Computer Science Review*, Elsevier, v. 19, p. 15–55, 2016.

SERTBAŞ, N.; AYTAÇ, S.; ERMIŞ, O.; ALAG¨OZ, F.; G¨UR, G. Attribute based content security and caching in information centric iot. In: ACM. *Proceedings of the 13th International Conference on Availability, Reliability and Security*. [S.l.], 2018. p. 34.

Shafigh, A. S.; Glisic, S.; Hossain, E.; Lorenzo, B.; DaSilva, L. A. User-centric distributed spectrum sharing in dynamic network architectures. *IEEE/ACM Transactions on Networking*, v. 27, n. 1, p. 15–28, Feb 2019.

SHAILENDRA, S.; SENGOTTUVELAN, S.; RATH, H. K.; PANIGRAHI, B.; SIMHA, A. Performance evaluation of caching policies in ndn-an icn architecture. In: IEEE. *2016 IEEE Region 10 Conference (TENCON)*. [S.l.], 2016. p. 1117–1121.

SHARIAT, A.; TIZGHADAM, A.; LEON-GARCIA, A. An icn-based publish-subscribe platform to deliver uav service in smart cities. In: IEEE. *2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. [S.l.], 2016. p. 698–703.

SHUKLA, S.; ABOUZEID, A. A. Optimal device-aware caching. *IEEE Transactions on Mobile Computing*, IEEE, v. 16, n. 7, p. 1994–2007, 2017.

SILVA, V. B. C. da; CAMPISTA, M. E. M.; COSTA, L. H. M. K. Trac: A trajectory-aware content distribution strategy for vehicular networks. *Vehicular Communications*, v. 5, p. 18 – 34, 2016. ISSN 2214-2096. Disponível em: ⟨http://www.sciencedirect.com/science/article/pii/S221420961630033X⟩.

SINKY, H.; KHALFI, B.; HAMDAOUI, B.; RAYES, A. Responsive content-centric delivery in large urban communication networks: A linknyc use-case. *IEEE Transactions on Wireless Communications*, IEEE, v. 17, n. 3, p. 1688–1699, 2018.

SUN, X.; WANG, Z. An optimized cache replacement algorithm for information-centric networks. In: IEEE. *Smart City/SocialCom/SustainCom (SmartCity), 2015 IEEE International Conference on*. [S.l.], 2015. p. 683–688.

SUN, Y.; FAYAZ, S. K.; GUO, Y.; SEKAR, V.; JIN, Y.; KAAFAR, M. A.; UHLIG, S. Trace-driven analysis of icn caching algorithms on video-on-demand workloads. In: ACM. *Proceedings of the 10th ACM International on Conference on emerging Networking Experiments and Technologies*. [S.l.], 2014. p. 363–376.

SUNG, J.; KIM, K.; KIM, J.; RHEE, J.-K. K. Efficient content replacement in wireless content delivery network with cooperative caching. In: IEEE. *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*. [S.l.], 2016. p. 547–552.

SURESHJANI, M. H.; MOGHIM, N. Cache replacement policy discard of fast retrievable content in named data networking of things. In: ACM. *Proceedings of the international conference on smart cities and internet of things*. [S.l.], 2018. p. 6.

SUTTON, R. S.; BARTO, A. G. Reinforcement learning: An introduction. In: ______. [S.l.]: MIT press, 2018. p. 30 – 36.

TANG, Y.; GUO, K.; MA, J.; SHEN, Y.; CHI, T. A smart caching mechanism for mobile multimedia in information centric networking with edge computing. *Future Generation Computer Systems*, Elsevier, v. 91, p. 590–600, 2019.

TARNOI, S.; KUMWILAISAK, W.; SUPPAKITPAISARN, V.; FUKUDA, K.; JI, Y. Adaptive probabilistic caching technique for caching networks with dynamic content popularity. *Computer Communications*, Elsevier, v. 139, p. 1–15, 2019.

TARNOI, S.; SUKSOMBOON, K.; KUMWILAISAK, W.; JI, Y. Performance of probabilistic caching and cache replacement policies for content-centric networks. In: IEEE. *39th Annual IEEE Conference on Local Computer Networks*. [S.l.], 2014. p. 99–106.

TARNOI, S.; SUPPAKITPAISARN, V.; KUMWILAISAK, W.; JI, Y. Performance analysis of probabilistic caching scheme using markov chains. In: IEEE. *2015 IEEE 40th Conference on Local Computer Networks (LCN)*. [S.l.], 2015. p. 46–54.

THOMAS, Y.; XYLOMENOS, G. Towards improving the efficiency of icn packet-caches. In: IEEE. *Heterogeneous Networking for Quality, Reliability, Security and Robustness (QShine), 2014 10th International Conference on*. [S.l.], 2014. p. 186–187.

TYSON, G.; SASTRY, N.; RIMAC, I.; CUEVAS, R.; MAUTHE, A. A survey of mobility in information-centric networks: Challenges and research directions. In: ACM. *Proceedings of the 1st ACM workshop on Emerging Name-Oriented Mobile Networking Design-Architecture, Algorithms, and Applications*. [S.l.], 2012. p. 1–6.

ULLAH, R.; REHMAN, M. A. U.; NAEEM, M. A.; KIM, B.-S.; MASTORAKIS, S. Icn with edge for 5g: Exploiting in-network caching in icn-based edge computing for 5g networks. *Future Generation Computer Systems*, Elsevier, v. 111, p. 159–174, 2020.

VIEIRA, V.; TEDESCO, P.; SALGADO, A. C. Designing context-sensitive systems: An integrated approach. *Expert Systems with Applications*, Elsevier, v. 38, n. 2, p. 1119–1138, 2011.

VURAL, S.; WANG, N.; NAVARATNAM, P.; TAFAZOLLI, R. Caching transient data in internet content routers. *IEEE/ACM Transactions on Networking*, IEEE, v. 25, n. 2, p. 1048–1061, 2017.

WANG, G.-q.; HUANG, T.; JIANG, L.; XIE, R.-c.; LIU, Y.-j. In-network caching for energy efficiency in content-centric networking. *The Journal of China Universities of Posts and Telecommunications*, Elsevier, v. 21, n. 4, p. 25–31, 2014.

WANG, H.; CHEN, Z.; XIE, F.; HAN, F. A data structure for content cache management in content-centric networking. In: IEEE. *Networking and Distributed Computing (ICNDC), 2012 Third International Conference on*. [S.l.], 2012. p. 11–15.

WANG, J. A survey of web caching schemes for the internet. *ACM SIGCOMM Computer Communication Review*, ACM New York, NY, USA, v. 29, n. 5, p. 36–46, 1999.

WANG, J. M.; BENSAOU, B. Improving content-centric networks performance with progressive, diversity-load driven caching. In: IEEE. *Communications in China (ICCC), 2012 1st IEEE International Conference on*. [S.l.], 2012. p. 85–90.

WANG, J. M.; BENSAOU, B. Progressive caching in ccn. In: IEEE. *Global Communications Conference (GLOBECOM), 2012 IEEE*. [S.l.], 2012. p. 2727–2732.

WANG, L.; BAYHAN, S.; KANGASHARJU, J. Optimal chunking and partial caching in information-centric networks. *Computer Communications*, Elsevier, v. 61, p. 48–57, 2015.

WANG, S.; BI, J.; WU, J. On performance of cache policy in information-centric networking. In: IEEE. *Computer Communications and Networks (ICCCN), 2012 21st International Conference on*. [S.l.], 2012. p. 1–7.

WANG, S.; BI, J.; WU, J.; LI, Z.; ZHANG, W.; YANG, X. Could in-network caching benefit information-centric networking? In: ACM. *Proceedings of the 7th Asian Internet Engineering Conference*. [S.l.], 2011. p. 112–115.

WANG, W.; SUN, Y.; GUO, Y.; KAAFAR, D.; JIN, J.; LI, J.; LI, Z. Crcache: Exploiting the correlation between content popularity and network topology information for icn caching. In: IEEE. *2014 IEEE international conference on communications (ICC)*. [S.l.], 2014. p. 3191–3196.

WEI, T.; CHANG, L.; YU, B.; PAN, J. Mpcs: A mobility/popularity-based caching strategy for information-centric networks. In: IEEE. *Global Communications Conference (GLOBECOM), 2014 IEEE*. [S.l.], 2014. p. 4629–4634.

WILLIAMSON, C. On filter effects in web caching hierarchies. *ACM Transactions on Internet Technology (TOIT)*, ACM New York, NY, USA, v. 2, n. 1, p. 47–77, 2002.

WOOD, S.; MATHEWSON, J.; JOY, J.; STEHR, M.-O.; KIM, M.; GEHANI, A.; GERLA, M.; SADJADPOUR, H.; GARCIA-LUNA-ACEVES, J. Iceman: A system for efficient, robust and secure situational awareness at the network edge. In: IEEE. *Military Communications Conference, MILCOM 2013-2013 IEEE*. [S.l.], 2013. p. 1512–1517.

WU, T.-Y.; LEE, W.-T.; DUAN, C.-Y.; WU, Y.-W. Data lifetime enhancement for improving qos in ndn. *Procedia Computer Science*, Elsevier, v. 32, p. 69–76, 2014.

XIN, Y.; LI, Y.; WANG, W.; LI, W.; CHEN, X. Content aware multi-path forwarding strategy in information centric networking. In: IEEE. *Computers and Communication (ISCC), 2016 IEEE Symposium on*. [S.l.], 2016. p. 816–823.

XING, L.; ZHANG, Z.; LIN, H.; GAO, F. Content centric network with label aided user modeling and cellular partition. *IEEE Access*, IEEE, v. 5, p. 12576–12583, 2017.

XYLOMENOS, G.; VERVERIDIS, C. N.; SIRIS, V. A.; FOTIOU, N.; TSILOPOULOS, C.; VASILAKOS, X.; KATSAROS, K. V.; POLYZOS, G. C. A survey of information-centric networking research. *IEEE Communications Surveys & Tutorials*, IEEE, v. 16, n. 2, p. 1024–1049, 2013.

YANG, J.-Y.; CHOI, H.-K. Ppndn: Popularity-based caching for privacy preserving in named data networking. In: IEEE. *2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS)*. [S.l.], 2018. p. 39–44.

YANUAR, M. R.; MANAF, A. Performance evaluation of progressive caching policy on ndn. In: IEEE. *Advanced Informatics, Concepts, Theory, and Applications (ICAICTA), 2017 International Conference on*. [S.l.], 2017. p. 1–6.

YAO, H.; QIU, C.; FANG, C.; CHEN, X.; YU, F. R. A novel framework of data-driven networking. *IEEE Access*, IEEE, v. 4, p. 9066–9072, 2016.

YAO, L.; CHEN, A.; DENG, J.; WANG, J.; WU, G. A cooperative caching scheme based on mobility prediction in vehicular content centric networks. *IEEE Transactions on Vehicular Technology*, IEEE, v. 67, n. 6, p. 5435–5444, 2018.

YAO, L.; WANG, Y.; XIA, Q.; XU, R. Popularity prediction caching using hidden markov model for vehicular content centric networks. In: IEEE. *2019 20th IEEE International Conference on Mobile Data Management (MDM)*. [S.l.], 2019. p. 533–538.

YEH, E.; HO, T.; CUI, Y.; BURD, M.; LIU, R.; LEONG, D. Vip: Joint traffic engineering and caching in named data networks. In: IEEE. *Computing, Networking and Communications (ICNC), 2015 International Conference on*. [S.l.], 2015. p. 695–699.

YOKOTA, K.; SUGIYAMA, K.; KURIHARA, J.; TAGAMI, A. Rtt-based caching policies to improve user-centric performance in ccn. In: IEEE. *2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA)*. [S.l.], 2016. p. 124–131.

Zaidi, S.; Ben Smida, O.; Affes, S.; Vilaipornsawai, U.; Zhang, L.; Zhu, P. User-centric base-station wireless access virtualization for future 5g networks. *IEEE Transactions on Communications*, v. 67, n. 7, p. 5190–5202, July 2019.

ZENG, Y.; XIE, J.; JIANG, H.; HUANG, G.; YI, S.; XIONG, N.; LI, J. Smart caching based on user behavior for mobile edge computing. *Information Sciences*, Elsevier, v. 503, p. 444–468, 2019.

ZHANG, C.; REN, P.; DU, Q. A contextual multi-armed bandit approach to caching in wireless small cell network. In: IEEE. *2017 9th International Conference on Wireless Communications and Signal Processing (WCSP)*. [S.l.], 2017. p. 1–6.

ZHANG, G.; LI, Y.; LIN, T. Caching in information centric networking: A survey. *Computer Networks*, Elsevier, v. 57, n. 16, p. 3128–3141, 2013.

ZHANG, G.; LIU, J.; CHANG, X.; CHEN, Z. Combining popularity and locality to enhance in-network caching performance and mitigate pollution attacks in content-centric networking. *IEEE Access*, IEEE, v. 5, p. 19012–19022, 2017.

ZHANG, H.; XIE, R.; ZHU, S.; HUANG, T.; LIU, Y. Dena: An intelligent content discovery system used in named data networking. *IEEE Access*, IEEE, v. 4, p. 9093–9107, 2016.

ZHANG, L.; AFANASYEV, A.; BURKE, J.; JACOBSON, V.; CLAFFY, K. C.; CROWLEY, P.; PAPADOPOULOS, C.; WANG, L.; ZHANG, B. Named data networking. *ACM SIGCOMM Computer Communication Review*, ACM, v. 44, n. 3, p. 66–73, 2014.

ZHANG, L.; ESTRIN, D.; BURKE, J.; JACOBSON, V.; THORTON, J. D.; SMETTERS, D. K.; ZHANG, B.; TSUDIK, G.; CLAFFY, k.; KRIOUKOV, D.; MASSEY, D.; PAPADOPOULOS, C.; ABDELZAHER, T.; WANG, L.; CROWLEY, P.; YEH, E.

Named Data Networking (NDN) Project. https://named-data.net/techreport/TR001ndn-proj.pdf, 2010.

ZHANG, M.; LUO, H.; ZHANG, H. A survey of caching mechanisms in information-centric networking. *IEEE Communications Surveys & Tutorials*, IEEE, v. 17, n. 3, p. 1473–1499, 2015.

Zhang, Q.; Wang, X.; Huang, M.; Li, K.; Das, S. K. Software defined networking meets information centric networking: A survey. *IEEE Access*, v. 6, p. 39547–39563, 2018.

ZHANG, T.; WANG, Z.; LIU, Y.; XU, W.; NALLANATHAN, A. Caching placement and resource allocation for cache-enabling uav noma networks. *IEEE Transactions on Vehicular Technology*, IEEE, v. 69, n. 11, p. 12897–12911, 2020.

ZHANG, Y.; TAN, X.; LI, W. In-network cache size allocation for video streaming on named data networking. In: ACM. *Proceedings of the 2017 VI International Conference on Network, Communication and Computing.* [S.l.], 2017. p. 18–23.

ZHANG, Y.; TAN, X.; LI, W. Ppc: popularity prediction caching in icn. *IEEE Communications Letters*, IEEE, v. 22, n. 1, p. 5–8, 2018.

ZHANG, Z.; LUNG, C.-H.; LAMBADARIS, I.; ST-HILAIRE, M. When 5g meets icn: An icn-based caching approach for mobile video in 5g networks. *Computer Communications*, Elsevier, v. 118, p. 81–92, 2018.

ZHANG, Z.; MA, H.; XUE, Y.; LIU, L. Fair video caching for named data networking. In: IEEE. *2017 IEEE International Conference on Communications (ICC).* [S.l.], 2017. p. 1–6.

ZHAO, W.; QIN, Y.; GAO, D.; FOH, C. H.; CHAO, H.-C. An efficient cache strategy in information centric networking vehicle-to-vehicle scenario. *IEEE Access*, IEEE, v. 5, p. 12657–12667, 2017.

ZHOU, X.; YE, Z. Popularity and age based cache scheme for content-centric network. In: IEEE. *2017 3rd International Conference on Information Management (ICIM).* [S.l.], 2017. p. 322–326.

ZHOU, Y.; CUI, L.; JIANG, Y.; XU, M. Modeling and optimizing the cache deployment with filter effect in multi-cache system. In: IEEE. *2013 IEEE Symposium on Computers and Communications (ISCC).* [S.l.], 2013. p. 000561–000566.

ZHOU, Y.; PHILBIN, J.; LI, K. The multi-queue replacement algorithm for second level buffer caches. In: *USENIX Annual Technical Conference, General Track.* [S.l.: s.n.], 2001. p. 91–104.

ZHOU, Y.; YU, F. R.; CHEN, J.; KUO, Y. Resource allocation for information-centric virtualized heterogeneous networks with in-network caching and mobile edge computing. *IEEE Transactions on Vehicular Technology*, IEEE, v. 66, n. 12, p. 11339–11351, 2017.

ZINK, M.; SUH, K.; GU, Y.; KUROSE, J. Watch global, cache local: Youtube network traffic at a campus network: measurements and implications. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. *Multimedia Computing and Networking 2008*. [S.l.], 2008. v. 6818, p. 681805.