Mobility-based Proactive Caching Models for Addressing Niche Mobile Demand and Scalable ICN Name Resolution Designs

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in
Computer Science

by
Xenofon Vasilakos
xvas@aueb.gr, http://pages.cs.aueb.gr/~xvas/

Advising Committee:

Advisor: Associate Professor Vasilios A. Siris
Associate Professor George Xylomenos
Professor George C. Polyzos

Athens, Greece
March 2017

PH.D. DISSERTATION

Athens University of Economics and Business,
School of Information Sciences and Technology,
Department of Informatics, Mobile Multimedia Laboratory
Evelpidon 47A, 113 62, Athens, Greece

Mobility-based Proactive Caching Models for Addressing Niche Mobile Demand and Scalable ICN Name Resolution Designs

by
Xenofon Vasilakos

Date of the Ph.D. Defence: March 1st, 2017.

Examination Committee

- Associate Professor **Vasilios A. Siris**, School of Information Sciences and Technology, Department of Informatics, Athens University of Economics and Business (Advisor)
- Professor **George C. Polyzos**, School of Information Sciences and Technology, Department of Informatics, Athens University of Economics and Business (Member of the Advising Committee)
- Professor **Christos Douligeris**, School of Information and Communication Technologies, Department of Informatics, University of Piraeus
- Associate Professor **Stathes P. Hadjiefthymiades**, School of Science, Department of Informatics and Telecommunications, National and Kapodistrian University of Athens
- Associate Professor **Vana Kalogeraki**, School of Information Sciences and Technology, Department of Informatics, Athens University of Economics and Business
- Assistant Professor **Ioannis F. Marias**, School of Information Sciences and Technology, Department of Informatics, Athens University of Economics and Business
- Assistant Professor **Stavros Toumpis**, School of Information Sciences and Technology, Department of Informatics, Athens University of Economics and Business

DEDICATION

To my family and friends, for all their love, kind patience and support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

As a doctoral thesis is the pinnacle of many years of hard working and learning, I feel it is appropriate to begin by thanking my teachers. My thesis advisor, Associate Professor V. Siris, taught me to ask the right questions and to focus on the essence of things. I would like to thank him for his tireless and patient guidance, and for our stimulating discussions which helped me to keep my motivation high throughout my Ph.D. studies. I am also grateful to the other two members of my Ph.D. supervisory committee, Associate Professor G. Xylomenos and Professor G. Polyzos, for their guidance and encouragement. They have all been instrumental in obtaining financial support for conducting my research and our joint work witnesses their influence on my academic advancement, which has been invaluable for my professional development.

A special thanks goes to Dr. K. Katsaros, with whom we worked closely together. His guidance with respect to conducting a valid and solid system evaluation has been substantial and influential for future research efforts. Also, I am thankful to M. Pomonis for our fruitful collaboration and his overall assistance.

At a personal level, I had the privilege of working alongside great colleagues and friends at the Mobile Multimedia Laboratory. Our collaborations or discussions on various topics spanning from research to virtually everything, have affected my way of thinking and helped me to open my mind. I sincerely hope that our paths will cross again in the future. These people are Drs. N. Fotiou, V. Douros, and C. Stais, who were the first people that I met in the lab, Dr. C. Tsilopoulos, who I already had the pleasure to know and with whom we started our Ph.D. adventure together, and later on Y. Thomas, Dr. M. Karaliopoulos and L. Chatzieleftheriou. Also, I would like to extend my thanks to V. Giannaki, to Assistant Professor G. Papaioannou, to Drs. C. Ververidis and P. Fragoudis, and to all the administrative and technical staff of AUEB for their kind assistance. It was a great pleasure to have met all of them.

Last and certainly not least, this dissertation –just as any other milestone in my life– would have not been possible without the continuous support of my parents Orsalia and Stelios, and my grandparents Konstantinos, Kyriaki and Kyriakos. Also, I would like to thank all of my close relatives and friends. They have shown remarkable patience, kindness, understanding and endurance to all those days and long nights of work that led to this thesis, regardless of holidays and weekends.

VITA

2007	B.Sc. in Informatics, Athens University of Economics and Business, Greece
2009	M.Sc. in Parallel and Distributed Computer Systems, Vrije Univerity, Amsterdam, Netherlands
2017	Ph.D in Informatics, Athens University of Economics and Business, Greece

Full CV available at: `http://pages.cs.aueb.gr/~xvas/pdfs/CVdetailedXV.pdf`.

PUBLICATIONS

Journal Papers

X. Vasilakos, V. A. Siris, and G. C. Polyzos, **"Addressing Niche Demand based on joint Mobility Prediction and Content Popularity caching,"** Computer Networks, 2016, Vol. 110, pp. 306–323.

G. Xylomenos, C. Ververidis, V. A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. Katsaros, and G. C. Polyzos, **"A survey of Information-Centric Networking research,"** Communications Surveys & Tutorials, IEEE, 2014 Vol. 16(2), pp. 1024–1049.

G. Xylomenos, X. Vasilakos, C. Tsilopoulos, V. A. Siris, and G. C. Polyzos, **"Caching and Mobility Support in a Publish-Subscribe Internet Architecture,"** Communications Magazine, IEEE, 2012, Vol. 50(7), pp. 52–58.

G. Lucarelli, X. Vasilakos, and I. Androutsopoulos, **"Named Entity Recognition in Greek Texts with an Ensemble of SVMs and Active Learning,"** International Journal on Artificial Intelligence Tools, 2007, Vol. 16(6), pp. 1015–1045.

Refereed conference and workshop papers

V. A. Siris, X. Vasilakos, and D. Dimopoulos, **"Exploiting Mobility Prediction for Mobility & Popularity Caching and DASH adaptation,"** in Proceedings of the IEEE 17^{th} International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2016, pp. 1–8.

X. Vasilakos, and V. A. Siris, **"Adapting data popularity in mobility-based proactive caching decisions for heterogeneous wireless networks,"** in Proceedings of the 10^{th} International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness (QShine), 2014, pp. 191–192.

V. A. Siris, X. Vasilakos, and G. Polyzos, **"Efficient Proactive Caching for supporting Seamless Mobility,"** in Proceedings of the 15^{th} IEEE International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2014, pp. 1–6.

X. Vasilakos, K. Katsaros, and G. Xylomenos. **"Cloud Computing for Global Name-Resolution in Information-Centric Networks,"** in Proceedings of the IEEE Network Cloud Computing and Applications Symposium, 2012, pp. 88–94.

X. Vasilakos, V. A. Siris, G. C. Polyzos, and M. Pomonis, **"Proactive Selective Neighbor Caching for Enhancing Mobility Support in Information-Centric Networks,"** in Proceedings of the ACM ICN'12 workshop, in conjunction with SIGCOMM'12, 2012, pp. 61–66.

X. Vasilakos, V. A. Siris, and G. C. Polyzos, **"Towards Exploiting User-Centric Information for Proactive Caching in Mobile Networks,"** 28th World Wireless Research Forum (WWRF28), Athens, Greece, 2012.

K. Katsaros, N. Fotiou, X. Vasilakos, C. Ververidis, C. Tsilopoulos, G. Xylomenos, and G. C. Polyzos, **"On the Inter-domain Name Resolution for Information-Centric Networks"**, IFIP Networking, Prague, Czech Republic, 2012, pp. 13–26.

V. Giannaki, X. Vasilakos, G. Xylomenos, and G. C. Polyzos, **"Supporting Mobility in a Publish-Subscribe Internetwork Architecture,"** Mobile Multimedia Laboratory, IEEE ISCC, Corfu, Greece, 2011, pp. 1030–1032.

X. Vasilakos, V. A. Siris, and G. C. Polyzos, **"A Selective Neighbor Caching Approach for Supporting Mobility in Publish/Subscribe Networks,"** 5th ERCIM Workshop on eMobility (in conjunction with WWIC 2011), 2011, p. 63.

D. Trossen, X. Vasilakos, P. Flegkas, V. Sourlas, and G.Parisis, **"Mobility Work Re-Visited Not Considered Harmful"**, 3rd IEEE International Workshop on Mobile Computing and Networking Technologies (WMCNT 2011), 2011, pp. 1–8.

X. Vasilakos, J. Sacha, and G. Pierre, **"Decentralized As-Soon-As-Possible Grid Scheduling: a Feasibility Study,"** 2nd IEEE Workshop on Grid and P2P Systems and Applications (GridPeer), IEEE ICCCN'10, Zürich, Switzerland, 2010, pp. 1–6.

Technical Reports

V. A. Siris, X. Vasilakos, and G. C. Polyzos, **"Efficient Proactive Caching for Supporting Seamless Mobility,"** in arXiv:1404.4754[cs.NI], 2014.

N. Fotiou, K. V. Katsaros, X. Vasilakos, C. Tsilopoulos, C. N. Ververidis, G. Xylomenos, and G. C. Polyzos, **"H-Pastry: An Adaptive Multi-level Overlay Inter-Network,"** Athens Univ. of Econ. and Business, Greece, 2011. Available online at: `http://mm.aueb.gr/technical reports/2011-MMLAB-TR-003.pdf`.

V. Giannaki, X. Vasilakos, G. Xylomenos, and G. C. Polyzos, **"Supporting Mobility in a Publish Subscribe Internetwork Architecture,"** Athens Univ. of Econ. and Business, Greece, 2011. Available online at: `http://mm.aueb.gr/technicalreports/2011-MMLAB-TR-001.pdf`.

ABSTRACT OF THE DISSERTATION

Mobility-based Proactive Caching Models for Addressing Niche Mobile Demand and Scalable ICN Name Resolution Designs

Since its original conception, the Internet has undergone a series of radical technological breakthroughs and usage expansions. From the perspective of its users, it is a global framework for sharing *information* among millions of fixed and mobile users. However, its legacy architecture poses serious functionality limitations or inefficiencies to *information sharing & retrieval*. At the same time, the patched mobility support over the legacy architecture and even the intrinsic mobility support of Future Internet architectures that adapt the Information-Centric Networking (ICN) paradigm are not enough to sufficiently address the Quality-of-Service (QoS) requirements and the desired user-perceived Quality-of-Experience (QoE) levels in contemporary mobile application scenarios. These problems get further aggravated by the unprecedented and continuously increasing numbers of the different *Information Objects (IOs)* and mobile devices in the Internet, as well as the corresponding high name resolution traffic and mobile traffic volumes.

The main contribution of this thesis lies in the design of efficient *proactive caching* models for *enhancing* seamless mobility that run at the application layer of any underlying network architecture or within the network layer of ICN architectures. Our main model regards an *Efficient Mobility-based Caching (EMC)* scheme for addressing *niche* mobile demand along with popularity-based and legacy caching model extensions. Opposite to other proactive solutions, which focus exclusively on popular content, our proposed distributed design targets less popular or personalised content requests by prefetching and caching the desired information locally in small cells based on aggregated user *mobility prediction* information and a local cache congestion pricing scheme. Such niche demand, particularly for video content, represents a significant 20–40% of demand in the Internet and follows a growing trend. Due to its novel design, EMC can directly address niche demand and get easily extended to make a joint use of content popularity information with the novelty of dynamically *balancing* the influence of mobility prediction and content popularity information on local cache actions. Based on thorough performance evaluation simulations for exploring different demand levels, video catalogues and mobility scenarios, including human walking and automobile mobility, we show that the gains from mobility prediction can be high and able to adapt well to temporal locality due to the *localised* and *short timescale* character of the exploited mobility prediction measurements, exceeding cache gains from popularity-only caching. Moreover, the performance of our model can be further improved by adapting cache replacements at the cost of an added computational overhead.

Additionally, this thesis makes a contribution towards a feasible and *globally scalable ICN Name Resolution System (ICN-NRS)* to facilitate *information sharing & retrieval* in ICN. The contribution lies in certain design and performance aspects related to the structure of the underlying inter-domain topology in the Internet and specifically to *a)* the *design* of the *hierarchical overlay routing* mechanism that underlies the Distributed Hash Table-based Name Resolution System (DHT-NRS) and *b)* a thorough *performance quantification* study of DHT-NRS and other cutting-edge ICN NRSs with respect to load distribution and routing policy compliance across the Internet topology.

ΠΕΡΙΛΗΨΗ ΤΗΣ ΔΙΑΤΡΙΒΗΣ
(ABSTRACT IN GREEK)

Μοντέλα Προληπτικής Ενταμίευσης βάσει της Κινητικότητας των χρηστών για την Αντιμετώπιση της ζήτησης περιεχομένου Εξειδικευμένου ενδιαφέροντος και κλιμακώσιμα μοντέλα Επίλυσης Ονομάτων Περιεχομένου για Πληροφοριοκεντρικές αρχιτεκτονικές Δικτύων.

Από την εποχή της αρχικής του σύλληψης, το Διαδίκτυο έχει υποστεί μια σειρά από ριζοσπαστικές αλλαγές λόγω της τεχνολογικής προόδου και της διεύρυνσης των εφαρμογών χρήσης του Διαδικτύου. Από τη σκοπιά των χρηστών του, αποτελεί ένα παγκόσμιο μέσο για την ανταλλαγή αντικειμένων Πληροφορίας[1] ανάμεσα σε εκατομμύρια σταθερούς και κινητούς χρήστες. Ωστόσο, η κληροδοτημένη από τα προηγούμενα χρόνια αρχιτεκτονική του Διαδικτύου, θέτει σοβαρούς περιορισμούς ή καθιστά ανεπαρκή τη διάθεση και την ανάκτηση της επιθυμητής Πληροφορίας. Πλέον αυτού, τόσο οι επεκτάσεις που εφαρμόστηκαν (εν είδει «μπαλώματος») στην υπάρχουσα αρχιτεκτονική με στόχο την υποστήριξη της κινητικότητας των χρηστών, όσο και η εγγενώς προσφερόμενη υποστήριξη της κινητικότητας στην περίπτωση των αρχιτεκτονικών Πληροφοριοκεντρικής δικτύωσης (Information-Centric Networking (ICN)), *δεν* επαρκούν για να αντιμετωπίσουν τις σύγχρονες ανάγκες και απαιτήσεις σε όρους *ποιότητας-της-υπηρεσίας* (Quality-of-Service (QoS)) και επιθυμητού επιπέδου *ποιότητας-της-εμπειρίας* (Quality-of-Experience (QoE)) των χρηστών. Τα προβλήματα επιδεινώνονται από το καινοφανές και συνεχώς αυξανόμενο πλήθος των διαφορετικών αντικειμένων της Πληροφορίας και των κινητών συσκευών στο Διαδίκτυο, το οποίο συνεπάγεται υψηλά επίπεδα όγκου κίνησης δεδομένων, τόσο για τα αιτήματα επίλυσης ονομάτων περιεχομένου, όσο και για τη μετάδοση των ίδιων των αντικειμένων της Πληροφορίας.

Η κύρια συνεισφορά της παρούσας διατριβής έγκειται στο σχεδιασμό μοντέλων *αποδοτικής προληπτικής ενταμίευσης* (efficient proactive caching) για την ενίσχυση της απρόσκοπτης κινητικότητας των χρηστών. Τα εν λόγω μοντέλα είναι εκτελέσιμα σε επίπεδο εφαρμογής οποιασδήποτε υποκείμενης αρχιτεκτονικής δικτύωσης. Πλέον αυτού, δύνανται να εκτελούνται και σε επίπεδο δικτύου στην περίπτωση των αρχιτεκτονικών Πληροφοριοκεντρικής δικτύωσης. Το βασικό μοντέλο που προτείνουμε, είναι το αποκεντρωμένο μοντέλο *αποδοτικής προληπτικής ενταμίευσης βάσει της κινητικότητας* (Effiecient Mobility-based Caching (EMC)). Πιο συγκεκριμένα, το EMC έχει σχεδιαστεί για να προβαίνει σε αυτόνομες αποφάσεις για τη προληπτική λήψη και ενταμίευση των δεδομένων τοπικά εντός μικρών κυψελών ασύρματης μετάδοσης (small cells), κάνοντας χρήση *α)* της πληροφορίας πρόβλεψης με βάση τη *συνολική κινητικότητα* (aggregated mobility) των χρηστών και *β)* ενός *σχήματος δυναμικής τιμολόγησης* με βάση τη *συμφόρηση της μνήμης* (dynamic cache congestion pricing scheme) του τοπικού ενταμιευτή. Το EMC έχει ως στόχο του τη βελτίωση της εξυπηρέτησης της ζήτησης για περιεχόμενο *εξειδικευμένου* ενδιαφέροντος (Niche content demand). Σε αντίθεση με άλλες λύσεις προληπτικής ενταμίευσης, οι οποίες επικεντρώνονται αποκλειστικά στα δημοφιλή αντικείμενα της Πληροφορίας, η ζήτηση για εξειδικευμένο περιεχόμενο αναφέρεται σε λιγότερο δημοφιλές περιεχόμενο ή περιεχόμενο εξατομικευμένου ενδιαφέροντος, το οποίο –ιδίως στη περίπτωση περιεχομένου τύπου *βίντεο*– αντιπροσωπεύει

[1] Η έννοια της Πληροφορίας (Information) αναφέρεται σε οτιδήποτε μπορεί να προσφερθεί ή να ζητήθεί από τους χρήστες του Διαδικτύου, συμπεριλαμβανομένων των προσφερόμενων *υπηρεσιών* και *αντικειμένων περιεχομένου*.

ένα σημαντικό τμήμα επί του συνόλου των αιτημάτων στο Διαδίκτυο, της τάξεως του 20-40% και με αυξητική τάση. Τέλος, πλέον του EMC προτείνουμε μια σειρά από επεκτάσεις στο βασικό μοντέλο αποφάσεων, οι οποίες συνδυάζουν τη πρόβλεψη της κινητικότητας των χρηστών με τη *δημοφιλία* (popularity) του αιτούμενου περιεχομένου, με καινοτόμο στοιχείο τη *δυναμική εξισορρόπηση* της επιρροής μεταξύ της πρόβλεψης της κίνησης των χρηστών και της δημοφιλίας των αντικειμένων.

Βάσει ενδελεχών προσομοιώσεων για την αξιολόγηση της επίδοσης των μοντέλων που προτείνουμε, οι οποίες εξερευνούν διαφορετικά επίπεδα ζήτησης, συλλογές αντικειμένων τύπου βίντεο και διαφορετικά σενάρια κινητικότητας των χρηστών (π.χ., περίπατος ανθρώπων ή αυτοκίνηση), δείχνουμε ότι τα κέρδη από την πρόβλεψη της κινητικότητας μπορεί να είναι υψηλά. Παράλληλα, διαπιστώνουμε ότι τα κέρδη από τη κινητικότητα μπορεί να υπερβαίνουν εκείνα από την ενταμίευση με βάση μόνο τη δημοφιλία του περιεχομένου, υποδηλώνοντας έτσι ότι οι αποφάσεις του EMC είναι ικανές να προσαρμόζονται καλά στη *τοπικότητα της δημοφιλίας στο χρόνο* (temporal locality) λόγω της *τοπικότητας* και της *μικρής χρονικής κλίμακας* των μετρήσεων για τη πρόβλεψη της κινητικότητας των χρηστών. Επιπλέον, η απόδοση της λύσης μας μπορεί να βελτιωθεί περαιτέρω με την αντικατάσταση ήδη ενταμιευμένων δεδομένων με δεδομένα από μετέπειτα αιτήματα, υπό το κόστος μιας επιπρόσθετης υπολογιστικής επιβάρυνσης σε ό,τι αφορά τις αποφάσεις ενταμίευσης του μοντέλου.

Επιπροσθέτως, με την παρούσα διατριβή συνεισφέρουμε ως προς το σχεδιασμό ενός *εφικτού* και *κλιμακώσιμου Συστήματος Επίλυσης ICN Ονομάτων* (globally scalable ICN Name Resolution System (ICN-NRS)), ως την απαραίτητη υποδομή για την *ανταλλαγή & ανάκτηση* της Πληροφορίας σε όλο το εύρος του Διαδικτύου. Η ιδιαίτερη συνεισφορά μας έγκειται στις πτυχές του σχεδιασμού του μοντέλου επίλυσης ονομάτων που σχετίζονται με τη δομή της υποκείμενης τοπολογίας μεταξύ διαφορετικών δικτυακών τομέων (inter-domain topology) στο Διαδίκτυο, και πιο συγκεκριμένα *α)* στον σχεδιασμό του μηχανισμού ιεραρχικής δρομολόγησης πάνω από το επικαλυπτόμενο δίκτυο που υποστηρίζει το σύστημα επίλυσης ονομάτων Distributed Hash Table-based Name Resolution System (ICN-NRS) καθώς επίσης και *β)* στη διεξοδική μελέτη για την *ποσοτικοποίηση των επιδόσεων* του DHT-NRS αλλά και άλλων συστημάτων τεχνολογίας αιχμής από τη βιβλιογραφία πάνω σε συστήματα ICN-NRS.

Chapter 1

Introduction

This chapter introduces the reader to the fundamental concepts required for understanding the research work presented in this Ph.D. dissertation. These concepts go back to the original assumptions and corresponding design goals that shaped the native functionality and the underlying host-centric communication model of the current Internet architecture, as well as to the further added, extended and/or increased user needs that have formed an altered, more complex contemporary Internet usage model. In addition, this chapter discusses the intrinsic gap between the natively offered functionality and the current user needs for *information sharing* and seamless *mobility*, while also highlighting the inefficiencies and/or the ineffectiveness of the corresponding patched functionality solutions running on top of the legacy architecture. The former motivate the dual research focus of this dissertation for supporting *a) scalable content & service distribution* via designing and adapting an Internet-wide scalable *Name Resolution System* within the context of an *Information-Centric*[2] Future Internet architecture, and for *b) enhancing mobility support* with targeted solutions tailored for the special purposes of particular application scenarios.

In a nutshell, this introductory chapter is structured as follows: Section 1.1 features the main aspects which *motivate* our research. In Section 1.2 we express our *thesis* regarding the heart of the problems presented in Section 1.1 and proceed with stating the *original contributions* and the novelty of our work in Section 1.3. Finally, we close this chapter by outlining the structure of the current dissertation in Section 1.4 .

1.1 Motivation for the dissertation

We begin this section by arraying the original assumptions of the underlying Internet architecture and explaining why these assumptions no longer adhere to the reality, posing serious functionality limitations (Section 1.1.1). Then, we focus on the current situation with respect to two out of the most important new user needs, namely,

[2] Due to the special nature of the Information-Centric Networking architectural paradigm, the reader is provided with a dedicated and detailed chapter on page 18.

a) scalable named content & service sharing and retrieval (Section 1.1.2) and *b) mobility support* (Section 1.1.3).

1.1.1 The Internet legacy architecture and its inborn limitations

The current Internet architecture tracks its roots to concepts that began first to form in the early 1960's for its predecessor, the Advanced Research Projects Agency Network (ARPANET) [1], an early *packet switching* network which was the first to implement what was later to became the standard Internet protocol suite, commonly known as TCP/IP from its most important protocols, namely, the Transmission Control Protocol (TCP) and the Internet Protocol (IP). The prevailing requirement of that time was to interconnect a small number of trusted hosts and to enable the access to shareable expensive resources in supercomputers or mainframes, most commonly CPU cycles, large memory or storage capacity for running demanding tasks, as well as to various peripherals which were scarse and expensive at the time. As a result, the current Internet architecture has been founded upon *end-point communication* between terminal devices or hosts[3] lying in different networks that employ arbitrary architectures. In addition, all Internet terminals of the time of its original conception were wired stationary hosts, which along with end-point communication led to a host-centric model (a.k.a. end-point model) of communication based on the perception that the end-points were *immobile*.

The *host-centric* model was fairly simple: any host that wanted to communicate with another host merely needed to send data to the latter's IP address, and vice-versa. Furthermore, it matched conveniently with the already familiar and established public telephony model [2]. These features enabled host-centric Internet communication to continue to exist along with the significant addition of TCP in the first half of the 1970's [3] which extended the host-centric semantics by adding source and destination port numbers to corresponding source and destination IP addresses in data transport sessions between hosts.

But ever since its advent, the Internet has undergone a series of radical changes with respect to its usage model and device composition, which have triggered a series of new, more complex requirements. Two out of the most important of these requirements are the need for *a)* a *scalable distribution* of content and services (Section 1.1.2) and *b) mobility support* (Section 1.1.3), with other important requirements being communication security, user privacy, accountability, trust, etc. Moreover, the Internet has been experiencing an astounding growth in size and usage popularity due to major application and service innovations running on top of it, and due to important technological breakthroughs in the (inter-)network layer. The former have been the outcome of (i) the *"hourglass"*-shaped suite of the Internet protocols around a "narrow waist" of a few essential protocols on the network layer, and (ii) the significant advances in emerging

[3] For the rest of this dissertation, we use the terms "terminal devices", "hosts" or –for simplicity– "terminals" interchangeably to refer to end-point network nodes.

technologies such as wireless networks and portable devices. While the earlier enabled arbitrary network technologies to interconnect and interoperate easily on the link layer, and a multitude of applications to be developed and run easily on the application layer, the latter allowed for an easy and ubiquitous access to Internet applications, hence increasing its user masses.

Yet with size comes a concern regarding the ability of the architecture to scale and to efficiently utilise some of its non-infinite resources like bandwidth. Currently, the Internet facilitates more than a billion of miscellaneous devices (with a strong trend for mobile devices [4]), over a trillion [5, 6] of web pages with some reports [7] estimating this number far beyond that to even 10^{15}, and an enormous annual amount of transferred data in the order of Exabytes [4]. However, its legacy architecture can *not* directly address such complex needs due to two key aspects of its design:

1. **Strong identity and location coupling:** The identity of the communicating end-points is strongly coupled to their location address.

2. **End-to-end functionality:** The whole "intelligence" is pushed at the edge of the network on the basis of the so-called *"end-to-end argument"* [8], according to which the desired functionalities must be implemented only at the end points were the applications stand, rather than in the unreliable intermediary network nodes.

While the earlier point clearly cumbers seamless mobility support (see discussion in [9]), the latter prevents the integration of new distributed applications into the core of the network as a result of the distinctive role between middle and end-point nodes. This is a source of performance considerations, economic trade-offs, and empirically observed problems for users [10, 11] with respect to complicated settings, e.g. for firewalls or Network Address Translators (NATs), physical or logical infrastructure configurations, troubleshooting, etc.

1.1.2 Scalable Named Information sharing and retrieval

Today's Internet users are primarily interested in receiving (often vast) amounts of information, irrespective of the physical location of the hosts that can serve the desired information. This contrasts the prevailing need of past-generation users for direct end-host communication and (from a users' perspective) has eventually turned the Internet into a global indexing substrate for discovering and serving information after its *name*. Named pieces of information –which we refer to as *Information Objects (IOs)*[4]– have a very general context, embracing everything that can be offered via the Internet, from actual data-objects like files, to services and "things" within the context of the Internet of Things (IoT). And even when users still wish to directly access (or share) a specific resource, or to run a demanding application on top of a suitable hardware or software

[4] See detailed definition of an IO in Appendix A.1 on page 148.

infrastructure, the current trend is for *cloud* technologies which enable the use of multi-tenant resources that reside somewhere in the Internet "cloud" (see the discussion on page 7) behind the abstraction of a *named service*.

Shifting from a traditionally host-centric to an *information-centric* usage model happened through a series of major events presented in a chronological order in Figure 1.1, and via gradual technological and/or service evolution that eventually turned the Internet from a network of interconnected end-hosts to a substrate of interconnected information. On the one hand, milestones such as the advent of the World Wide Web (WWW) triggered information-centrism, whereas on the other hand the gradual evolution of, e.g. Content Delivery Networks (CDNs), has patched the existing network functionality in order to satisfy the arising usage requirements with respect to content discovery, scalable distribution and network performance. In what follows, we outline these milestones:

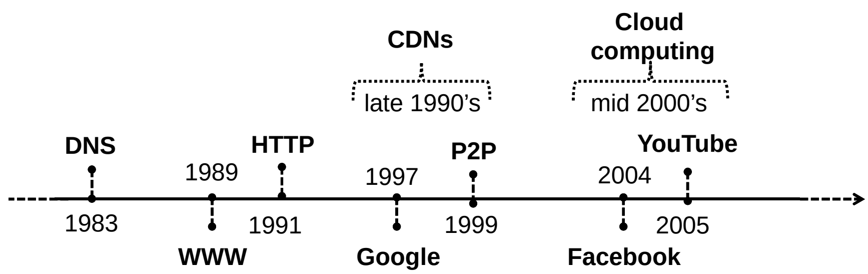


Figure 1.1: Chronicle towards an information-centric usage model in the Internet.

Domain Name System

The Domain Name System (DNS)[5] is a hierarchical, distributed, autonomous and reliable database which was standardised [12, 13] in 1983. Even though there was no intention to facilitate or promote information-centrism, DNS made the WWW (discussed next) possible by enabling the resolution of human readable Uniform Resource Locators (URLs) to actual IP addresses, i.e. it was the first step for using *names* for resources rather than raw addresses of host locations.

World Wide Web

The second –and most significant– milestone towards information-centrism, took place in 1989 with the invention of the World Wide Web (WWW), which introduced a special *information space* implemented on the application layer of the Internet architecture. The WWW is comprised of interlinked pieces of content in the form of documents that contain formatted text, multimedia content or even software components, and which are identified by a URL used to access these documents via special web browser applications. Alongside its communication protocol, HTTP (discussed next), WWW offers

[5]We return and discuss DNS again in Section 3.2.1 on page 40 with respect to its (in)adequacy as an name resolution service for information-centric networks.

an easy-to-use information-centric infrastructure on the application layer of the existing Internet architecture.

Hypertext Transfer Protocol

Arguably [14], the Hypertext Transfer Protocol (HTTP) already supports a fundamental set of information-centric primitives on an application level. It is a type of a request-response protocol over TCP, according to which a client, e.g. a web browser, submits a request to a server, e.g. a web server, which replies with a request-completion status and/or the requested content in the payload of its response, or performs a function on behalf of the client. HTTP offers an existing and convenient communication substrate beyond WWW usage such as in the case of Dynamic Adaptive Streaming over HTTP (DASH) (a.k.a. *MPEG-DASH*) [15], which employs adaptive bitrate streaming and has been used by YouTube, or for "web seeding" and tracking in BitTorrent [16].

Web search engines

The Web resembles a gigantic repository of IOs where only the number Web pages exceeds that of a trillion [5, 6]. Such a scale consequently raises a problem with respect to discovering the desired IOs. To address this issue, *web search engines* were designed to provide search services on the WWW. While initially focused only on textual searching, search engines can provide various types of searching, e.g based on an input image, as well as various search options for refined searches, e.g. within a certain geographical region. The first experimental search engine effort was the W3 Catalog [17] in 1993, with many to follow ever since. Being already developed since 1997, Google's search engine, based on the PageRank [18] search algorithm rose steadily to prominence during the 2000's and dominates web searching ever since.

Content Delivery Networks and Peer-to-Peer systems

Today's Internet works despite the paradox of an underlying network legacy architecture that remains host-centric and the fact that the current information owners or principals do *not* directly serve their content nor their services from their own network "premises". To make things work effectively and economic-/performance-wise efficiently, the current reality involves two major scalable alternatives to the traditional client-server model, namely, *a) Content Delivery Networks (CDNs)* and *b) Peer-to-Peer (P2P) content sharing* systems. On the one hand, P2P solutions are non-corporate and fully distributed in nature, while on the other hand, CDNs offer a centrally administered corporate service that is based on bringing content close to where it is consumed with the increasing cooperation of Internet Service Providers (ISPs) [19]. In what follows, we overview both approaches. Note that with time CDNs have prevailed over P2P solutions.

Content Delivery Networks

Content Delivery (or Distribution) Networks (CDNs)[6] are wide-scale, distributed networks of special servers deployed across multiple data centres within the ASs of ISPs, which offload work from origin servers by delivering content on their behalf. To do so, CDNs "hack" the natively supported host-centric communication with the cooperation of ISPs by intercepting user requests and consequently serving them from their own distributed infrastructure. CDNs started to form and evolve gradually during the late 1990's with infrastructure development, mirroring, caching and ISP multihoming, following a pre-formation period of server "farms", hierarchical caching and cache-proxies that were used to improve the web. Amongst many, some notable examples of vendors that offer commercial CDN services include Akamai (1998) [20], Limelight Networks (2001) [21] and Amazon (2008) [22], while there are also many other big companies which have their own CDN facilities like Netflix, Google, Microsoft, Apple, Facebook etc.

Peer-to-Peer content sharing systems

Peer-to-Peer (P2P) content sharing systems are composed of nodes which form an *overlay* network sitting on top of the Internet's actual inter-domain network substrate. As opposed to CDNs, P2P overlays do not employ a dedicated third-party infrastructure. Instead, P2P content sharing is built upon a distributed, application-level system architecture that partitions tasks or workload amongst *peers*, i.e. members that share the same roles, privileges, etc. Their introduction was significant for two main reasons: First, because they moved away from the traditional client-server model which matches the host-centric model of communication, and second, because they emphasized on both *indexing* and *distributing* information after its *name*. The most important P2P solutions proposed throughout the past two decades are:

- **Napster (1999):** Although the key concepts of P2P had previously been used in many application domains in the past, it was the major success of Napster [23] in 1999 which had a significant impact on the Internet usage by employing a centralised index for discovering content in the form of files distributed amongst peers who could then proceed with exchanging the files. Napster served as a precursor for later P2P overlay applications to come like eDonkey or Gnutella.

- **Edonkey (eMule) & Gnutella (circa 2000):** EDonkey, which later became known as eMule [24], improved on indexing by distributing this part of the architecture to special indexing servers. Likewise, Gnutella [25] employed distributed indexing, only by using Time To Live (TTL)-based flooding among special-purpose "servant" peers. These designs were more scalable and without any single point of failure, contrary to the centralised index of Napster that could not scale and posed a major vulnerability, as it made the Napster susceptible to attacks.

[6] We return and discuss the details of typical CDN functionality in Section 5.1.3 on page 75.

- **BitTorrent (2001):** Finally, things changed dramatically with the introduction of the BitTorrent [16] P2P content distribution system (circa 2001). BitTorrent enables a more efficient model for exchanging parts of data objects among multiple peers, and deals sufficiently with the problem of "free-riding" by peers that enjoy the advantage of downloading, yet without uploading content in return. At its hight of success, in 2008 BitTorrent was responsible for $\sim$34% of the entire Internet traffic [26], posing a serious challenge for ISPs who at the time saw their routing policies being implicitly violated, for reasons analysed in [27], along with a traffic explosion. The protocol is still widely in use to reduce the server and network impact of distributing large files but, notably, BitTorrent and all other P2P solution have a gradually declining share on global traffic [4] mainly in favour of video content.

Cloud computing

"Cloud computing" is an umbrella term used to refer collectively to the massively scalable computer capabilities of shared configurable processing resources, e.g. servers, storage, networks, applications or services, which are delivered to external customers from corresponding cloud providers via the Internet as a service on demand. Tracing its roots to the 1990's Virtual Private Network (VPN) services, cloud computing can be formally identified by five essential characteristics [28]: (i) unilateral, on-demand *provisioning* of computing capabilities, e.g. server time and/or network storage; (ii) *broad network access* via thin or thick client platforms, e.g. mobile phones; (iii) *resource pooling* and *multi-tenancy* of providers' physical and virtual computing resources which are dynamically (re)assigned after consumer demand; (iv) *rapid elasticity*, i.e. capabilities can be elastically provisioned and released to scale rapidly with consumer demand; and finally, (v) cloud providers monitor, control, report and automatically optimize *resource usage* with respect to the type of the offered service. Evidently, such a service model offers a series of advantages to customer organisations, particularly to the *smaller* ones with limited capital resources or to businesses with alternating high and low peak seasons, as opposed to investing in local CapEx and OpEx.

Social Networks

Social Networks (Social Networks)[7] are online platforms which reflect social relations among users who share a common background such as employment, news interests or real-life connections, thus forming a *personalized*, online application-layer network per user profile. Since their advent, they are having an increasing impact on ISP traffic [29] and on wireless mobility: Cisco [4] verifies that part of the observed increase in wireless traffic volumes comes as the joint result of the increasing popularity of Social Networks and particularly Facebook [30] (2004), in addition to traffic

[7] We return and discuss the implications of Social Networks with respect to video content demand within the context of mobility support in Section 5.1.4 on page 77.

from providers like YouTube, Netflix, etc., and the increased usage of technologically advanced smart phones and tablets with higher recording and playback resolution capabilities. Other important Social Networks include Twitter [31] (2006), LinkedIn (2002), Flickr (2004), Tumblr (2007), Instagram (2010), etc.

Large video-hosting & streaming services

Internet video demand lies in the heart of today's content-centrism of the Internet, representing one of the most demanding applications and most advancing types of traffic with respect to current and projected Internet usage [4]. Large video hosting websites allowed for this to happen. They are designed for uploading and publicly sharing personal, business, or royalty-free video content by individual users or businesses, most commonly via a website service or alternatively via a desktop application, an Application Programming Interface (API) and more recently straight via the individual users' mobile devices. The hosting service stores the content on its data servers, which it can then serve for viewing purposes via different types of embed codes or links. While the first-launched video hosting websites were initially able to serve only short clips due to technical reasons, nowadays video uploads may refer to any video length (even full-length movies) or quality as a result of an extensive and advanced CDN and ISP enhanced support [32, 33]. The world's leading and largest dedicated video sharing Internet service is YouTube [34] (2005). Nonetheless, it is not the first dedicated video hosting website to exist as it was preceded by "shareyourworld.com" (1997). Other important large video hosting websites include, but are not limited to, Metacafe (2003), Vimeo (2004) and DailyMotion (2005), while in the more recent years big Social Networks like Facebook have also become popular with respect to hosting user-generated video content.

Aside from user-generated videos in video-hosting and social networks there are also large Internet video streaming services. In 2007, Netflix [35] started to stream videos on demand (series or films) in the USA and as of 2016 it can stream videos almost worldwide. It is currently the largest Internet video streaming service with notable alternatives to Netflix being Amazon Prime, Cloudload, iTunes, Google Play, etc.

Finally, one must not pass by the role of the content-specific video hosting websites discussed in [36], which share a considerable part of traffic in the Internet. Demand from such video hosting websites is characterised by some special features, notably an extremely rapid decay of video popularity and only a partial popularity of specific parts within the same video.

1.1.3 Mobility support

As discussed in Section 1.1.1, mobility was not an issue at the time that the original Internet architecture was conceived. Back at that time, network communication regarded only wired, uniquely addressed, stationary terminals, which is a perception that has long ceased to hold due to the evolution and the ever-growing usage of wireless mobile technologies during the past two decades. Current trends show that wireless networking is undergoing an unparalleled growth to the point that mobile hosts will overtake fixed ones, not only in number but also in terms of traffic load: According to the annual mobile data traffic forecast by Cisco Systems for the period 2016-2020 [4], mobile data demand grew 4000x during the past ten years and by 74% in 2015 only, while demand levels per device are expected to grow to 4.4 GB per month by 2020, which is almost a 5x increase relative to 2015. Along the same lines, the 2014 review on 5G radio access by Ericsson [37] states that wireless communication systems will support more than 1000x today's traffic volume beyond 2020.

To address the need for supporting host mobility over the current TCP/IP architecture, the research community has contributed a variety of protocols which "patch" mobility support to the legacy architecture by hiding the *strong coupling* (see discussion on page 3) of a host's identity to its location from the application layer. The most important out of these protocols are the Mobile IP (MIP) protocol [38, 39] and its most recent hierarchically extended version, the Hierarchical MIPv6 (HMIP) [40], which was introduced to enhance support in micro-mobility scenarios. Though achieving their goal of supporting host mobility, the performance of these as well as other protocols is known to suffer [41] from slow handovers, with corresponding efforts like the Fast Handover for MIPv6 [42] scheme ending up in failure [43]. Nevertheless, recent developments have allowed audio/video streaming and other applications to appear to run seamlessly in mobile scenarios, i.e. in scenarios where mobile users perform a *horizontal* handover to another Access (or Attachment) Point (AP) or even a *vertical* handover to a different access technology such as from using Wi-Fi to cellular and vice-versa. Within this context, the most prominent examples are DASH [15] and Apple's HTTP Live Streaming [44]. Both approaches employ an adaptive bit-rate streaming technique that enables the use of alternative video segments (resp., streams) encoded at different bit-rates based on the current network conditions. As a result, the client side can seamlessly adapt to dynamic network conditions and avoid stalling periods due to bandwidth oscillations such as during or after handovers. In another example, the solution in the work of [45] uses bandwidth prediction and video quality scheduling, which can mitigate network-layer handover delay.

The aforementioned examples indicate a trend towards pushing the complexity of seamless mobility support on the application layer of specially crafted, *receiver-driven* mobile-friendly applications, with any remaining handover delay issues being due to the link layer (e.g., with respect to connectivity or the establishment of a new connection) rather than due to the underlying ID/location coupling on the legacy network layer. This form of support may be application-specific, hence not universally applicable, and perhaps proprietary, for instance in the case of audio/video collaborative applications; however, it demonstrates that seamless mobility can be achieved at the cost of reduced quality such as by lowering the bit-rate in the case of DASH. Therefore, *what remains to be addressed* is the need for improving the available *QoS* levels and the user-perceived *QoE* both *during* and *after* performing a mobile *handover*. This applies also to the case of mobility over Publish-Subscribe (Pub-Sub) architectures or overlays. The Pub-Sub paradigm is already receiver-driven and asynchronous in nature, which enables intrinsic mobility support (Section 2.2.4 on page 22). Moreover, it can support mobility (performance-wise) better than the patched mobile protocols over TCP/IP[8]. Still, mobile users can face issues, e.g. with respect to post-handover delay.

Along these lines, mobility solutions need to focus on user application requirements regarding QoS and other aspects such as service charges, given the current or predicted network conditions, i.e. as a function of both network capabilities and the expected mobile demand. The key element towards this direction lies in exploiting *users' context*. Nonetheless, most mobility support solutions *lack* important contextual information. Leveraging specific user-centric features [48, 49] such as age category, preferences (e.g., for Sci-Fi films), requests knowledge or prediction (e.g., about news subscriptions), mobility information (e.g., using individual mobility patterns or predicting a route via an external mechanism like a GPS navigator) and behaviour (e.g., by using mass transportation at specific hours and days) can reduce costs and improve the QoS levels offered to users via acting *proactively* before users move and/or issue a request. For instance, learning and adapting to user behaviour is critical for improving QoE or energy consumption [50]. In another example, *"cellular network providers and location-based services can benefit from knowledge of the inter-play between users and their locations and interests."* [51]. In addition, *grouping users* on the basis of common context, preferences or mobility can be expected to lead to more efficient decisions based on a more precise estimation of handover probabilities, future requests and the reuse of cached IOs.

Last, using *"fresh" mobility information* is particularly crucial for improving the service of mobile demand. While studies in the literature [52, 53] have shown that individual, personalised mobility patterns are highly predictable in scenarios that involve typical human routine in rural or suburban areas, more recent studies [54, 55] show that mobility solutions need to take into account the *latest* mobility information, preferably from *nearby* sources, in order to make accurate predictions in scenarios that involve densely populated (i.e., urban) areas. This is due to the more complex traffic networks and traffic jams that exist in urban areas, as well as due to the fact that many vehicles, such as taxis, follow random mobility patterns.

1.2 Thesis

The inefficiencies and inabilities of the current Internet architecture that we discuss in Section 1.1 with respect to information sharing & retrieval stem from a series of legacy design aspects and assumptions which make the (inter-)network layer agnostic to *what* and *where* it is being transferred. At the same time, the patched mobility support over the legacy architecture and even the intrinsic mobility support of Future

[8] The reader may refer to the performance evaluation results of MIP and HMIP in the works of [46, 47] over native Pub-Sub architectures and in the work of [41] over a Pub-Sub multicast overlay designed for motility support.

Internet architectures based on the Pub-Sub communication paradigm are not enough to sufficiently address the QoS requirements and the desired user-perceived QoE levels in contemporary mobile application scenarios. These problems get further aggravated by the unprecedented and continuously increasing numbers of the different IOs and mobile devices in the Internet, as well as by the corresponding high content resolution traffic and mobile traffic volumes.

In order to deal with the unprecedented growth of mobile applications, their corresponding demand volumes and their application-specific requirements which pose a serious challenge for providers, we call for cost-efficient enhancement solutions tailored for specific *mobility* scenarios involving applications with strict delay, reliability and other QoS or application-/user-specific requirements. Such *targeted* solutions should also offer economic benefits for both the mobile providers and their users, and improve the QoE. Furthermore, we opt for the Information-Centric Networking (ICN)[9] design paradigm for the Future Internet architecture and identify *global name resolution* as the greatest challenge within the context of all ICN initiatives. ICN departs away from the "end-to-end argument" and dismisses the obsolete assumption of stationary terminal hosts by serving requests after their *name* rather than resolving requests to host locations. As a result, the network layer can have a full knowledge on what is being transferred, its origins and destination, which allows the orchestration of more secure, targeted and efficient delivery structures such as via multicast, while at the same time the overall architecture is intrinsically *a) mobile-ready* and *b) information-centric*, matching the contemporary user needs.

Based on the above, we can outline our thesis in terms of the following three points:

(1) **We need to address *unique* mobile demand:** *Unique* demand refer to requests for IOs for which individual users have an exclusive interest within the boundaries of their currently hosting mobile network and, thus, they can not be addressed via "conventional" popularity-based caching techniques. Our thesis is that unique mobile demand is important and calls for a special treatment, particularly in application scenarios with strict delay and/or reliability requirements. Some representative examples include real-time or emergency notification services, home video surveillance, future applications within the context of IoT and online gaming, which are sensitive to end-to-end delay and/or require lossless transmission of the data published during mobile disconnection periods.

(2) **We need to address *niche* mobile demand:** In order to improve performance further than what can be achieved via conventional popularity-based caching, it is our thesis that we need to directly address *niche* mobile demand for the less popular or personalised content that lies in the (so-called) "long tail" of typical content popularity distributions. Niche content always represented a significant part of the

[9] We provide the reader with the necessary background on ICN in Chapter 2.

total demand and gains even more significance due to the arising popularity of mobile video, particularly in Social Networks. As with the case of unique demand, niche demand can not be adequately addressed via “conventional” popularity-based caching approaches exactly due to its less popular nature. At the same time, the distribution of popular content is already successfully addressed in practice by CDNs, which further justifies our thesis for focusing on the remaining part of mobile requests for which popularity information is known for (i) having a little value [14, 56] regarding the IOs lying in the long tail and for (ii) being not in a par with temporal locality conditions in certain mobility scenarios [54] due to user mobility.

(3) **Global ICN name resolution remains open:** Name resolution is a key function for the ICN paradigm. Our thesis is that the design of a feasible ICN Name Resolution System (NRS) in global adaptation scenarios remains open. Despite the body of work, the literature lacks a meticulous study for understanding the qualitative and quantitative properties of the proposed approaches under realistic network assumptions. Therefore, there is room for *exploring new architecture designs* as well as for revisiting and *further investigating* the performance aspects of *existing solutions*, in a pursuit towards a feasible global ICN NRS that *must* have the following properties:

   (a) be able to guarantee global resolution;

   (b) manage to balance the distribution of the resolution state across the domains of the inter-AS topology taking into account the size and the individual capabilities of the ASs;

   (c) imply scalable messaging burden;

   (d) be compliant to the established Border Gateway Protocol (BGP) routing policies.

1.3 Contributions

With this dissertation we make the following four original contributions. Note that contributions (1-3) can lie both within and *beyond* the scope of Information-Centric Network (ICN), being able to run as an application layer solution over any network architecture including the TCP/IP as well as to fully integrate with the network layer of ICN architectures. In contrast, contribution (4) lies exclusively within the research scope of ICN.

(1) **An *Efficient Proactive Caching* model for addressing *unique* mobile demand:** Our first contributions is an Efficient Proactive Caching (EPC) model which uses dedicated *cache* resources near to or collocated with mobile APs that are equipped with a local cache, to prefetch and cache data for individual mobile

user requests. The solution is tailored to delay-sensitive applications and addresses mobile requests that are *unique* to users, by exploiting *mobility* prediction and individual user requests information, and by applying a local cache congestion pricing scheme. Cache actions take place on a per-user request basis, following a simple, lightweight and fully decentralised process of autonomous local decisions at neighbouring caching points. The applied cache congestion pricing scheme allows efficient utilisation of the local cache-storage resources and can have advantages (mainly being lightweight and taking short time-scale online cache decisions that reflect temporal locality) over the centralised mathematical optimisation approximation approaches used in other proposals in the literature for reaching closer to an optimal allocation of the distributed cache resources.

(2) **An Efficient *Mobility-based Caching* solution for addressing *niche* content mobile demand:** Efficient Mobility-based Caching (EMC) is a proactive solution which addresses *niche* content mobile demand for personalised or less popular IO requests in wireless heterogeneous networking scenarios via proactive caching decisions at *small cells*, which exploit *aggregated* user *mobility* prediction information from requests for the *same* IO. The model uses the same cache congestion pricing scheme as EPC, hence it exhibits the same desirable features such as cache decisions being lightweight and fully decentralised at small cells, and it can be further extended to an Efficient Mobility-based Caching model with cache Replacements (EMC-R). EMC and particularly EMC-R can reach closer to an optimum allocation of the locally available cache resources due to exploiting mobility and content popularity information based on short time-scale measurements. In essence, short time-scale measurements combined with the distributed nature of EMC/EMC-R allows these protocols to capture and adapt to temporal locality more accurately than centralised proactive approaches designed for mobility support. The latter use long-term or predicted content popularity based on information from requests that span multiple domains and, hence, do *not* effectively capture locality in neither space or time.

(3) **A *joint* Mobility & Popularity-based Caching scheme for addressing both mobile *niche* and *popular* content requests:** Our third contribution is an Efficient Mobility and Popularity-based Caching model with cache Replacements (EMPC-R). This scheme advances further compared to EMC-R by adapting content *popularity* and *legacy*-caching model extensions to cache decisions. Alongside addressing niche mobile demand, EMPC-R aims to serve users who share the same popular content requests by extracting content popularity information from the requests of all the currently attached users to a small cell. Therefore, cache actions with EMPC-R can also benefit stationary (wireless or fixed) and temporarily immobile users such as students in a classroom during a class, for which the decision model lacks mobility information. In addition, EMPC-R can utilise any legacy-cached data that correspond to IOs for which there are currently no ac-

tive mobile requests, hence there is *no* available mobility information to utilise for cache decisions. Last, along with its inherent advantages from EMC-R, a major *novelty* of EMPC-R lies in its ability to dynamically *balance* the contribution of users' mobility and content popularity information on cache actions.

(4) ***Feasible global* Name Resolution in Information Centric Networking:** Our last contribution is on the design and evaluation study of a *feasible* Information-Centric Networking Name Resolution System (ICN NRS) that can both *a)* conform to the established policies and relations, and *b) scale* across the Internet topology. Towards this direction:

(a) We contribute, in part, to the design of the Hierarchical Pastry (H-Pastry) DHT architecture and to the meticulous performance evaluation of the Distributed Hash Table-based Name Resolution System (DHT-NRS) which uses H-Pastry as its overlay routing mechanism. Our exact contribution lies in *adapting multihoming* and *sibling* relations between ASs to the hierarchical inter-domain structure of H-Pastry. The conclusion drawn from this part of our research is that DHT-NRS/H-Pastry faces *issues* regarding *respecting* the established inter-domain *routing policies*. Moreover, due to the Canon paradigm, DHT-NRS/H-Pastry needs to assume that the business interests of ISPs with trust or collaborative relations (e.g., sibling domains) would allow them to reveal information about their internal organisation to each other. Consequently, its feasibility as an ICN NRS in global adaptation scenarios is under question.

(b) Triggered by (i) the former conclusion on the feasibility of DHT-NRS, (ii) the results of a *head-to-head* performance comparison between DHT-NRS and the Data-Oriented (and beyond) Network Architecture (DONA), and (iii) a series of similar conclusions drawn from the related literature on lookup-by-name schemes, we turn our interest to studying the performance aspects of approaches that adapt the route-by-name design paradigm. Specifically, we put our focus on DONA and the Content-Ubiquitous Resolution and deLivery Infrastructure for Next Generation services (CURLING). Our motivation lies in the intrinsic policy compliance features of these architectures which, however, *lack* a *rigorous evaluation* of their *scalability* properties in the literature. Our contributed study involves a thorough quantification of important performance aspects, particularly *load distribution* across the global Internet topology and the implied *processing & signalling overheads*. The outcome of this study justifies that *CURLING* can *qualify* as a possibly *feasible* ICN NRS in Internet-wide deployment scenarios.

Table 1.1: Summary of proactive caching model contributions (contributions 1-3), with references to unique model features, system running complexity and related publications.

	Unique features	Scenarios	Complexity of a cache decision (Neighbourhood of N cache points)	Publications

EPC	▪ Focuses on *unique* mobile demand:
	  ▪ Uses individual mobility prediction and individual mobile requests
		▪ Edge network mobility`
		▪ Subscriptions of unique interest
		▪ Home video surveillance content
		▪ Mobile video gaming
		▪ Encrypted content
			▪ Computational: (O(1)) for local cache decision
			▪ Messaging. Either multiple unicast or multicast
			  ▪ Multi-unicast: O(N); extra messages needed for hierarchical caching extension with asymptotically same complexity.
			  ▪ ICN Multicast: O(1)
				▪ 2011, 5th ERCIM Workshop, A Selective Neighbor Caching Approach for Supporting Mobility in Pub-Sub Networks
				▪ 2012, ACM ICN'12/SIGCOMM'12. Proactive Selective Neighbor Caching for Enhancing Mobility Support in ICNs
				▪ 2014, WoWMoM, Efficient Proactive Caching for supporting Seamless Mobility
				▪ 2016, Comp. Networks, Addressing Niche Demand based on joint Mobility Prediction and Content Popularity caching

EMC	▪ *Niche* demand: personalised & less popular ▪ Captures *temporal locality* due to its distributed nature and use of short timescale measurements ▪ Uses aggregated user mobility prediction	▪ Heterogeneous Wireless N/Ws ▪ Social networking ▪ Personalised video demand	▪ Computational: same as EPC ▪ Messaging: same as EPC; multicast option also via the macro cell	▪ 2016, WoWMoM, Exploiting mobility prediction for mobility & popularity caching and DASH adaptation ▪ 2016, Comp. Networks, Addressing Niche Demand based on joint Mobility Prediction and Content Popularity caching
EMC-R	▪ *Niche & popular* demand ▪ Closer to optimal cache sharing ▪ Added computational cost for replacements	▪ Same as EMC	▪ Computational: O($\log(n)$) were *n* is the number of objects already in the cache ▪ Floyd's algorithm for inserting n objects: O(n); significant improvement vs. Williams heap insertion ▪ Messaging: same as EMC	▪ 2016, WoWMoM, Exploiting mobility prediction for mobility & popularity caching and DASH adaptation ▪ 2016, Comp. Networks, Addressing Niche Demand based on joint Mobility Prediction and Content Popularity caching
EMPC-R	▪ Content popularity and legacy-caching extensions ▪ Balances mobility and popularity information	▪ Same as EMC/EMC-R ▪ Stationary or temporarily immobile users, e.g. during course classes or in working environments	▪ Computational: same as EMC-R ▪ Messaging: same as EMC and EMC-R	▪ 2014, QShine: "Adapting data popularity in mobility-based proactive caching decisions for heterogeneous wireless networks" ▪ 2016, WoWMoM, Exploiting mobility prediction for mobility & popularity caching and DASH adaptation ▪ 2016, Comp. Networks, Addressing Niche Demand based on joint Mobility Prediction and Content Popularity caching

Additional work

Apart from the aforementioned list of main contributions, our work on proactive caching additionally includes the following:

1. We study the impact of mechanisms such as mobile handover strategies and CDN-level caching on locally applied proactive caching schemes.

2. We provide an analytical discussion of the gain *performance metrics* with an emphasis on delay costs.

3. We explore the impact of various system parameters on performance. These parameters include (i) *decision-specific parameters* which, for instance, define how fast dynamic cache prices adapt to current demand levels; (ii) various combinations of *total* mobile cache *demand* and *supply* levels; (iii) *geospatial parameters* such as cache density or user density in space, small-cellular coverage range; (iv) various synthetic and real-trace user *mobility models*; (v) synthetic and real video content *request traces*; (vi) different *catalogue* size over local cache size ratios, etc.

4. We identify and motivate the *increasing importance* of *niche* mobile demand based on notable studies in the literature [57, 56, 58, 59, 60, 61]

5. We provide a thorough discussion on the particular role and impact of Social Networks on Internet video demand.

6. We investigate two main *application scenarios*. First, an edge-mobility scenario that involves delay-sensitive requests that are unique to users. In such a scenario users roam within the same edge network such as inside a building or a campus while continuously detaching and re-attaching to local APs (i.e., a micro-mobility scenario), or the users may move between more than one neighbouring edge networks such as inside an airport where multiple Local Area Networks (LANs) falling under different administrative authorities can coexist (i.e, a scenario with cooperating edge LANs). Note that in this first application scenario there is no fall-back connectivity assumed, i.e. users can experience disconnection periods. Second, a mobility scenario in Heterogeneous Wireless Networks (HWNs) where cost-concerned user wish to reduce the monetary charges induced by their macro-cellular provider by exploiting their ephemeral connectivity to small cells.

7. A two-level *hierarchical extension* of our basic (i.e., EPC) and consequent (i.e., EMC, EMC-R, EMC-R) proactive caching decision models, which allows the application of our scheme(s) beyond the edge of the mobile network and to also leverage the in-network cache resources via distributed cooperative proactive caching.

8. A detailed discussion and complexity analysis on how to *implement* the proposed proactive caching models over two candidate Future Internet ICN architectures,

namely Publish-Subscribe Internetworking (PSI) and NDN, as well as over the existing network architecture.

9. We have developed an EPC simulator for edge network mobility scenarios with the integrated simulation environment of OMNeT++-v4.0 and a custom Java simulator for EPC, EMC, EMC-R and EMPC-R in Heterogeneous Wireless Networking scenarios.

Finally, the body of work on ICN name resolution includes:

1. A discussion on the potential role of *cloud computing* in addressing the scalability concerns raised with respect to a global-level name resolution design, motivated by the ability of the cloud to offer large volumes of resources on demand and the design advantages with respect to routing efficiency of route-by-name schemes.

2. We have developed a simulator for DONA with the integrated simulation environment of OMNeT++-v4.0. Furthermore, we developed a custom Java simulator for DONA. The custom simulator was needed in order to overcome resource limitations with the OMNeT++ environment, which prevented the use of full-scale Internet topologies from CAIDA [62]. The latter simulation software is fully parallelised and can be used (with minor changes needed) for evaluating CURLING as well.

1.4 Dissertation outline

This dissertation is organised as follows. Chapter 2 discusses the fundamentals of Information-Centric Networking (ICN), which are needed for understanding several parts of this thesis work. Chapter 3 presents notable work in the literature related to our own conducted research presented in the following chapters. Chapter 4 proposes the Efficient Proactive Caching (EPC) distributed model for addressing unique mobile demand for users running applications with strict delay requirements in edge network mobility scenarios. Chapter 5 sets forth a mobility-based caching model for addressing niche mobile demand in Heterogeneous Wireless Networking scenarios, along with content popularity and legacy caching model extensions. Chapter 6 offers our study work towards a feasible, globally scalable name resolution in a Information-Centric Networking Future Internet architecture. Finally, Chapter 7 puts forward our future work plans.

Chapter 2

Information-Centric Networking fundamentals

To address the shortcomings and inabilities of the current Internet architecture that we discussed so far, architectures that adapt the design paradigm of Information-Centric Networking, commonly referred to as Information-Centric Networks (ICNs), have emerged for the Future Internet architecture. ICN architectures are designed to natively satisfy the new and increasing user requirements for host mobility, named content dissemination and named services (i.e., for Information-Centric communication) that emerged after the original conception of the Internet, rather than the traditional need for a pair-wise (i.e., host-centric) communication. By *naming information at the network layer*, ICN tries to address a series of limitations in the current Internet architecture by providing the right native functionality for deploying in-network *caching*, multicast mechanisms, and a connectionless/receiver-driven data transportation model to both stationary and *mobile* terminals.

In what follows, we first discuss in Section 2.1 how the ICN paradigm adapts the *Pub-Sub pattern* as its primitive messaging model and clarify how this differs from traditional Pub-Sub notification systems in the literature. Then, we identify a background of common key functionalities in Section 2.2. Finally, Section 2.3 closes this chapter with an analytical description of the most *influential* ICN architectural approaches.

2.1 Publish-Subscribe pattern adaptation

ICN adapts the general concept of the Publish-Subscribe (Pub-Sub) messaging pattern for its networking primitives, as by design this pattern decouples information from its location. Only in ICN, the use of the terms *a)* "*publisher*" and *b)* "*subscriber*", as well as the corresponding acts of *a)* "*publishing*" an IO and *b)* "*subscribing*" to an IO name are used within a different context compared to traditional Pub-Sub: end-point users or in-network elements take up the role of a publisher (resp., a subscriber) when they *announce* the *availability* of information in (resp., *request to consume* information from) the inter-network. Opposite to that, the more traditional/established Pub-Sub literature [63] implies that the act of "publishing" and "subscribing" refer to the actual submission of data and the actual receiving of data when or while the data get published, respectively. Additionally, there can be permanent subscriptions, which in ICN terms corresponds to the option of matching multiple publication announcements to a single subscription regardless of *when* (prior to or after) the matching publication(s) are announced.

2.2 Key functionalities

Despite their different design approaches, all ICN research initiatives aim to address a common set of key functionalities. In what follows we identify these key functionalities, namely: (i) information naming, (ii) name resolution & data routing, (iii) caching, (iv) mobility support, (v) security, (vi) accountability and, last, (vii) privacy.

2.2.1 Information Naming

Information naming lies at the heart of ICN with names being used with respect to all sorts of granularity, from, e.g., a whole video file, to file chunks or even data packets. Despite the different approaches to naming, all Future Internet ICN architectures in the literature adapt location-*independent* names to avoid the issues faced by the current architecture. The exact choice of the structure of names reflects the design goals set for most of the other key functions, from IO discovery and caching up to routing, forwarding, and even security, ranging from *a) hierarchical* or more *complex naming graphs* [64] to completely *b) "flat"* names, with each of these options having their own strengths and weaknesses. In brief, hierarchical names can be human-readable while flat names can be self-certifying, and there is no way to combine both of these properties in the same naming structure [65]. For instance, the architecture in NDN (Section 2.3.2) relies on an external trust mechanism for binding signed IOs to human-readable names [66], while the design in MobilityFirst (Section 2.3.3) relies on an external naming system that binds human-readable names to identifiers.

In what follows, we provide more insight on the debate between flat and hierarchical naming structures.

Hierarchical names

The greatest advantage of adapting hierarchical names in an architecture lies on its *scalability* properties with respect to name resolution and routing in the current architecture via name *aggregation*. Scalability is a critical aspect, in general, and the most difficult problem to address towards a global name resolution feasible in ICN, in particular. The number of named IO in the Internet is already enormous, with only the number of *unique web pages* indexed by Google being more than a trillion [6] since 2008 and billions [4] of devices (mobile phones, PDAs, tablets, sensors, home appliances etc.) are already (or will soon be) publishing content and services to the network. Therefore, it should be expected that any name resolution design will have to handle a number of unique information names in the order of 10^{13}, or even even 10^{16} as estimated by other studies [67]. The former exceeds by far the current situation with respect to resolved names by DNS or the routing state that BGP handles.

But even though hierarchical names can reduce the aforementioned number from 10^{13} to 10^{8} [68] and there are further aggregation techniques designed for ICN [69] which report that they can mitigate the problem, *scalability remains prevalent.* NDN for instance, uses hierarchical names for both resolution and routing, and assumes that extensive aggregation will take place in specially designed routers in order to compress the resulting routing tables, particularly the ones lying at the core of the network were the routing state is (inevitably) expected to accumulate. PSI in another example, adapts a naming structure were name hierarchies and more complex structures can occur based on the notion of a *scope* ID, paired with a (flat) ID for each IO. Scopes can fall under one or more other scopes and IOs may belong to more than one scopes, possibly with a different flat ID. These examples of architecture reveal that there may be a huge number of prefixes in ICN. Moreover, prefixes can have unknown popularities, unlike the current reality which allows DNS name resolution and BGP routing to scale via aggregating multiple records under common popular prefixes.

As a last remark, to make such an extensive aggregation work in practice in ICN, names *must* reflect the actual network topology[10]. But as we argued on page 3, any form of binding names to locations cumbers mobility support.

Flat names

In contrast to hierarchical naming, flat names *avoid* any location-to-identity binding. On the one hand, this evidently facilitates mobility. However on the other hand, flat names are not easy to aggregate, which implies either some huge routing and/or name resolution tables, or some scalable but complicated and costly solutions based on Distributed Hash Tables (DHTs) that have their own shortcomings (Section 3.2.2 on page 41 for more).

2.2.2 Name Resolution & Data Routing

Name resolution refers to the function of resolving (i.e. matching) a name to any of the available sources that can supply the corresponding IO. Data routing on the other involves the construction of a path for transferring the resolved piece of information

[10] For a detailed explanation on why names must reflect the actual network topology, the reader may refer to the example portrayed in Figure 3 of reference [70] and the corresponding discussion in Section IV.A.

from its source to the requesting host. Depending on the ICN approach, the former two functions can be *a) coupled* or *b) decoupled*, whereas a further, more refined categorisation that applies only to name resolution function can be made between *a) lookup-by-name* and *b) route-by-name* approaches.

The coupled vs. the decoupled approach

Architectures which use the coupled approach, e.g. NDN, route user requests for attaining an IO to a resolved information provider via some path, and use the *reverse* of this path to forward the requested IO to the interested host. On the other extreme, architectures adapting the decoupled approach, e.g. PSI, use the name resolution function is completely unrelated to the construction of a forwarding path from an information provider to a requesting user. Also, there can be ICN architectures like DONA that are able to support both a coupled or a decoupled name resolution and data routing. Regardless of which approach is adapted by an architecture, it can be designed to easily support the creation of multicast trees from an information provider to multiple requesting users with the use of Pub-Sub primitives.

The lookup-by-name vs. the route-by-name approach

The lookup-by-name and the route-by-name design alternatives have their own advantages and shortcomings, triggering part of the research presented in this dissertation in Chapter 6. In a nutshell, lookup-by-name NRSs are essentially large-scale distributed databases that maintain mappings between IO names and the location of their most suitable source, e.g. the nearest or the currently least congested one. Within this context, a series of DHT-based solutions (Section 3.2.3 on page 42) have been explored in the literature as possible lookup-by-name NRSs, with typical DHTs used in these approaches being Chord [71] and Pastry [5]. On the other, solutions that take the route-by-name approach route name resolution requests *hop-by-hop* to the location of the corresponding IO itself, rather than to a network element that can match the desired name to another network location. Within this context lies the design of both DONA [72] (Section 3.2.4 on page 44) and CURLING [73] (Section 3.2.4 on page 45).

2.2.3 Caching

Caching can be categorised into "on-path" and "off-path" caching [74]. As demoted by their names, on-path caching leverages cached IOs along the network path used during name resolution, while off-path caching leverages cached IOs in dedicated caches outside that path. Note that off-path caching in particular, depends on the adapted name resolution and routing approach. Architectures that adapt a coupled name resolution and data routing support off-path caching by the routing system used to forward the requests for information, while architectures adapting the decoupled approach treat off-path caches as any other provider or source of information.

2.2.4 Mobility support

ICNs have two key differences compared to standard IP networks due to their Pub-Sub communication model, with implications to mobility: First, they employ a *receiver-driven* model where receivers request IOs after their name in an *asynchronous* manner from its provider(s). For instance, in Pub-Sub terms a mobile requester issues a subscription request for a data object; the latter object may or may be not published yet, but if/when it does, the pending subscription will be resolved and the transportation of the published data will begin from a resolved source to the current network location of the mobile. Second, data transportation from publishers to receivers is performed in a *connectionless*, hence stateless, manner, contrary to the connection-oriented, hence statefull, end-to-end control model adapted with TCP, which involves location-dependent IP addresses and has to be either patched [75] in order to maintain connections, or connections must start all over from the beginning.

A further advantage of the ICN paradigm with respect to mobility support lies in its *multicast* nature, and its combined provision of anonymity and – as mentioned above– inherent asynchrony [41]. First, anonymity and asynchrony enable the quick adaptation to the continuous attachment and detachment of mobile devices or agents to the APs that reside at the edge of a mobile network. Note that APs are *not* necessarily wireless; they may well refer to any kind of physical or software mobile layer, including wireless environments such as the IEEE 802.11 or the Universal Mobile Telecommunications System (UMTS), or to a mobile software agent [76] that migrates from on physical host in the "cloud" to another. Second, the multicast nature of Pub-Sub can handle large populations of mobiles and their continuous trend for switching to new locations. Last, given the restricted capabilities of mobiles such as with respect to battery life or bandwidth, multicast helps systems to better utilize their resources, e.g. by performing fewer (re)transmissions, which implies energy savings.

The above features allow mobile receivers to resubmit their requests for content that they did not receive while they were at their previous AP or they could not receive while being in transit to their next AP, *without* requiring re-establishing a connection. Therefore, ICN network architectures can *intrinsically* support *subscriber* host *mobility* without the need of a cumbersome and costly add-on overlay solution like MIP. Furthermore, the architecture itself can fully integrate special protocols that orchestrate multicast [41], smart caching [77] or a combination of both [78] to improve the utilisation of the available network resources when supporting mobile demand. However, there are still challenges to face and a need for further enhancing *seamless* mobility support in ICN. Increased *delay* for receiving data can still be incurred after handovers against application requirements, while there can also be reliability issues with respect to to some types of content publications issued during mobile disconnection periods. In addition, while the discussion above reveals that subscribers' mobility is straight-forward, publishers' mobility is difficult to facilitate because the name resolution system (resp., the routing tables) in the coupled (resp., decoupled) name resolution and data routing approach must be updated both frequently and promptly. Last, another issue regards *information* & *software* mobility. Both of these forms of mobility should be expected to be far more common than today in ICN architectures and a Future Internet flourished with cloud services. Apart from the performance challenges which are similar to supporting publisher mobility, changing regions raises issues [79] with respect to regionalisation and context transfer [79], due to implications to locality, trust, security and the ability to reconcile different and sometimes conflicting dissemination policies (e.g., in scenarios where information provisioning moves to a different legislative domain). While caching and multicast might alleviate some of these issues, e.g. within a network region due to recognized demand in the population, information and software mobility remains a difficult challenge.

2.2.5 Security, Accountability and Privacy

Human-readable name, e.g. hierarchical names, require some trusted relationship with the name resolution system or with a third-party agent so as to verify the actual correspondence between the returned data and the requested IO name. Unlike that, flat names (see the discussion on page 20) can support self-certification as in the case of [72]. However, they are *not* human-readable, pushing the problem of trust to another function that maps human-readable names to flat IO names.

Regarding accountability & privacy, all ICN architectures are driven by the requesters' explicit interests. This has benefits with respect to the deployment of in-network accountability mechanisms based on self-certifying names such as with the Packet Level Authentication (PLA) technique [80] for encrypting and signing individual packets. But even without self-certifying names, many ICN architectures use indirection (e.g., the RandezVous (RV) function of the PSI architecture discussed on the next page) between IO requesters and IO sources. The point of indirection must be globally trusted, i.e. it must know the identities and current locations of all parties, thus facilitating accountability as well as user privacy, as information providers do not need to be aware of the identities of the requesters and vice-versa. In addition, techniques such as LIPSIN [81] enhance privacy by hiding the location and the identity of the two communicating parties even from the forwarding entities on the communication path.

2.3 Important ICN architectures

Following its promising design principals, the ICN paradigm has naturally dragged the attention of the research community, resulting in a plethora of initiatives for the Future Internet architecture. These initiatives aim to replace the inherent host-centric communication model in order to directly address the problems and limitations identified on page 3 in Section 1.1.1. In what follows we present a detailed overview of the ICN architectural approaches of PSI on the next page and NDN on page 27. Amongst many, we chose to focus on these designs because they are highly influential in the field and because they are part of the background of our conducted research on global name resolution and mobility support. In addition, we outline the important aspects of the architectures in DONA, MobilityFirst, i3 and ROFL on page 30. DONA is amongst the first and most influential ICN architectures with respect to all of its aspects, while the latter have significant approaches to mobility support. The reader may refer to [70] for an analytical discussion and a cross-comparison amongst these as well as other major ICN architectures.

2.3.1 Publish-Subscribe Internetworking

Publish-Subscribe Internetworking (PSI) [82] is a clean-slate approach to a Future Internet architecture that was developed during the PUblish-SUbscribe Internet Technology (PURSUIT) [83] and its predecessor Publish-Subscribe Internet Routing Paradigm (PSIRP) [84] EU research projects, while it is also the basis for the iP Over IcN the betTer IP (POINT) [85] project. PSI adapts an architectural design which consists out of three clearly discrete functions [86, 87, 88]: *a)* the *RandezVous (RV)* function, which is incarnated via a dedicated Rendezvous Network (RENE) of special Randezvous Nodes (RNs); *b)* the *Topology Management* function, which is implemented by a dedicated Topology Manager (TM); and *c)* the *Forwarding* function, which implemented by a network of special Forwarding Nodes (FNs).

Name resolution & data routing

Following the aforementioned model, the process of name resolution, routing and forwarding in PSI can be outlined as a series of steps, where the order of step (1) and step (2) can be reversed due to the asynchronous Pub-Sub nature of the architecture:

(1) Publication: A publisher submits a special Publish message to its local RN identified by the name of the advertised information. This RN is the publisher's default gateway to the RENE which will act as a mediator in matching this publication with prospective (or already pending) subscriptions referring to the same information name. The Publish message is routed based on the advertised name via a DHT-based routing overlay, either the one proposed in [89] or the one proposed in [90], to the RandezVous Point (RVP), i.e. the RN that is responsible for matching requests for the published name. Note that a RVP may actually reside outside the boundaries of the AS of its publisher.

(2) Subscription: When a subscriber issues a Subscribe message to its local RN that refers to the same name as the one advertised with the aforementioned Publish message, the Subscribe message is routed to the resolution point RN were the matching takes place.

(3) Rendezvous match: The RV function matches the subscription to the publication based on the information name and then it instructs the TM to construct a delivery path route composed of FNs from the publisher to the subscriber.

(4) Path construction & data publication: The TM sends this route to the publisher with a special START PUBLISH message. Finally, the publisher starts to send the data to the subscriber via the path of the FNs.

As follows from the above, name resolution and data routing are *decoupled* in PSI: name resolution is performed by the RENE, and data routing is orchestrated by the TMs while executed by the FNs. Name resolution can be time consuming, especially due to DHT routing which can follow stretched paths over the underlay bellow the RENE. On the contrary, data forwarding [81] takes place at line speeds and without any state at the FN as we explain in the next paragraph.

Naming information

Regarding *naming* in PSI, each piece of information is identified by a statistically unique pair of IDs: a Scope ID (SID) and a flat Rendevous ID (RID). A scope groups related information together and can be used to impose a certain dissemination strategy for the scope such as with respect to access rights. Scopes can fall under other scopes, hence they can form complex scope structures including hierarchies [64]. Moreover, IOs may belong to more than one scopes and (possibly) have different RIDs.

Topology management

The incarnation of *topology management* is based on dedicated TM nodes which jointly implement this function by executing a distributed routing protocol such as the well known Open Shortest Path First (OSPF) protocol, in order to discover the network topology. The actual *delivery paths* are constructed upon request by the rendezvous function as a series of links between FNs. These links are encoded into source routes with the help of a technique [81] which is based on Bloom filters [91]. In essence, each network node assigns a tag (a long bit string produced by a set of hash functions) to each of its outgoing links, and advertises these tags via the routing protocol. A path through the network is then encoded by OR-ing the tags of its constituent links and the resulting Bloom filter is included in each data packet. When a data packet arrives at a FN, the FN simply ANDs the tags of its outgoing links with the Bloom filter in the packet and forwards the packet over a link upon a corresponding tag match. Finally, *multicast* transmission is feasible in PSI via encoding the entire multicast tree into a single Bloom filter.

Based on the aforementioned, we note the minimal state maintained by the forwarders in PSI: They merely maintain their link tags. The separation of the routing and forwarding functions allows the TMs to construct paths using complex criteria (e.g., load balancing), without requiring signaling to the (stateless) FNs. However, this comes at a significant cost: the topology management and forwarding functions can be practically applied only with respect to intra-domain data routing due to the Bloom filters susceptibility to false positive matches, particularly when the filter (i.e., the delivery path) contains much information (i.e., many link tags). To address this problem with respect to inter-domain routing, label[11] switching at the inter-domain level can be one solution, based on either label stacking or label replacement.

Regarding the implementation of *transport layer protocols*, e.g. via a sliding window of pending packet requests, data packets that belong to the same information can be *individually requested* by the subscriber using the notion of *Algorithmic IDs*. The latter are packet names generated by a pre-agreed algorithm between the communicating parties. Individual packet requests are forwarded in a like manner to data packets, using *reverse Bloom filters* calculated by the TM. Notice that the RENEhas no role in this process.

Caching

With respect to *caching*, PSI can natively support both on-path and off-path caching, as well as CDN-like managed information replication [74]: *a) on-path* caching is supported by intercepting and caching the forwarded packets at the FNs of a transfer path. *b) off-path* caching [92] (including a *proactive* approach [93, 94] for enhancing mobility support) and managed information replication [95] can take place with dedicated caches which can operate as alternative publishers via the standard rendezvous function. But unlike other architectures such as Named Data Networking (NDN), which is explicitly based on on on-path caching on symmetric paths, on-path caching in PSI can be less effective due to the fact that different data transfer sessions corresponding to requests for the same information may use entirely different paths. Regarding

Mobility support

PSI can fully support mobility with its pure Pub-Sub primitives and can further facilitate or enhance mobility via multicast and caching [74]. In particular, PSI identifies four mobility scenarios based on *a) host movement*, i.e. either *local* or *global*, and *b) technology interchange*, i.e. *static* when it involves a single technology and *dynamic* when it involves vertical handovers:

1. *Local subscriber* mobility can be handled via both multicast and/or caching, i.e. by multicasting IOs to multiple possible locations [41, 77] for the mobile subscriber and receiving IOs from nearby caches [93, 77] after a handover.
2. *Global subscriber* mobility is handled by modifying the forwarding function of the architecture [96].
3. *Mobility and mobile service enhancement* can be achieved in PSI with proactive caching based on user context information [48], mobility information and content popularity [97]. On the on hand, individual mobility prediction can be used to guarantee a certain level of QoS with an emphasis on *reducing* handover-related *latencies* by proactively caching the information requested by an individual mobile

[11] Where a label is the Bloom filter that contains the encoded path of link tags.

subscriber to the network areas where the subscriber is expected to handoff [93, 94, 98]. Additionally –and as a bi-product)– this approach can enhance reliability with respect to data that are published during the mobile subscriber's disconnection period, i.e. while the mobile is in transit to the next AP. On the other hand, joint mobility-based and content popularity proactive caching [97, 55] focuses on jointly exploiting (i) aggregated mobility-prediction information with respect to a common information request by mobile requesters, and (ii) the popularity of the requested information, both based on short-time scale available user mobility and content popularity information. The latter approach is less user-centric compared to the individual proactive caching service offered with the former approach. However, it targets niche mobile demand, which is a significant part of the traffic (Section 5.1.2 on page 74), can facilitate delay/cost-sensitive mobile applications and users, and, in overall, offer a better QoE to users.

4. *Publisher mobility* is hard to support because the topology management function must to be continuously updated with the mobile publisher's new network location.

Security

PSI supports the PLA technique [80] for encrypting and signing individual packets. This technique assures data integrity and confidentiality as well as malicious publisher accountability. The use of flat names also allows self-certifying names for immutable data objects, using the object's hash as the rendezvous ID. In addition, the paths encoded into Bloom filters can use dynamic link identifiers, making it impossible for an attacker to craft new or reuse old Bloom filters to launch Denial-of-Service (DoS) attacks.

2.3.2 Named Data Networking

Named Data Networking (NDN) [99] (formerly known as Content-Centric Networking (CCN) [100]) is a fully-fledged ICN architecture [2, 101] that envisions a new Internet protocol stack which will adapt the exchange of *named data* as the thin waist of the the architecture, using various networking technologies (including IP itself) below the waist for connectivity. Unlike PSI, CCN users issue INTERESTS which the network routes towards potential publishers. Names are hierarchical and special Content Routers (CRs) look-up for a longest prefix match to decide the upstream nodes that will forward INTERESTS up until a matching DATA PACKET is found. Then, a special Pending Interests Table (PIT) maps incoming interfaces to INTEREST packets in order to forward DATA PACKETS to users following the reverse path. Note that the choice of hierarchical naming in NDN allows to aggregate name resolution and data routing information across similar names, which is significant with respect to the scalability of the architecture.

Naming Information

Hierarchical names are similar to URLs. However, they are different in two ways: (i) their first part is neither a DNS name or an IP address; (ii) names do not have to be human-readable. Instead, a name component can be anything, including a hash value. A name request is considered to match any piece of information whose name prefix matches the requested name, e.g. `/aueb.gr/ai/main.html` can be matched by an information named `/aueb.gr/ai/main.html/_v1/_s1`. In the latter example, the match could signify the first segment of the first version of the requested data. After receiving this data object, the subscriber can request for the next data segment or next version of the information. Objects segmentation is expected to be known by the subscribers' application and the prefix matching rule enables applications to discover what is either available or will become available by a future publication.

Name Resolution & Data Routing

As mentioned earlier, subscribers issue INTEREST messages to request IOs in the form of DATA messages. Both types of message carry the name of the requested/transferred information, which is used to forward the messages hop-by-hop via a networks of CRs, each of which maintains three data structures: (i) the Forwarding Information Base (FIB), (ii) the PIT and (iii) the Content Store (CS) . The FIB maps names to the output interface(s) that should be used to forward the INTERESTs towards some appropriate data source; the PIT keeps track of the incoming interface(s) from which the pending INTERESTs arrive; and, finally, the CS serves as a local cache for caching information data as they pass through the CR.

When an INTEREST arrives and the extracted name matches an object in the CS, then it immediately sends back the cached data through the incoming interface in a DATA message, while the INTEREST is discarded. Otherwise the CR performs a longest prefix match on its FIB to decide towards which direction the INTEREST should be forwarded. If the PIT already contains an entry for the exact name, then the router adds the incoming interface to this PIT entry and discards the INTEREST. Note that the former leads to the formation of a multicast tree for the particular IO. Finally, when an IO that matches the requested name is found at a publisher node or a CS, the INTEREST message is discarded and the information is returned in a DATA message. This message is forwarded back to subscriber(s) in a hop-by-hop manner, based on the state maintained in the PITs.

The former reveal that name resolution and data routing are *coupled* in NDN, thus routing is symmetric. In order to populate the FIBs, NDN can use distributed routing protocols like OSPF, in which CRs advertise name prefixes rather than IP address ranges. A CR may have multiple interfaces in its FIB for a prefix, for example, if it is multi-homed or if it is aware of multiple CDN servers hosting the information. In this case its strategy layer may choose to send the INTEREST either to all these interfaces or only to the interface that has exhibited the best performance so far.

Caching

With respect to caching, NDN natively supports on-path caching in the CSs of the CRs as described above. The CS can use a, e.g. Least Recently Used (LRU), replacement policy, but, practically/realistically it can not be used as a long-term storage. Therefore it is mostly useful for recovery from packet losses and for handling flash crowds of requests. Off-path caching is supported by delivering an Interest to any data source that may be hosting the requested IO, e.g., the strategy layer can direct the Interest to a CDN server rather than to the originating publisher. However, this is not transparent to the architecture because it requires populating the FIBs with pointers to such copies.

Mobility support

To support mobility, mobile subscribers in NDN can simply issue new Interests from their current location. The Interests will be suppressed by the PIT of the first common CR in both delivery routes, prior and post the handover. Nevertheless, the corresponding IOs will be also delivered to the mobiles' old location. When a publisher moves on the other hand, the FIBs pointing to it have to be updated, thus requiring advertising again the name prefixes for the information it is hosting. This represents a very high overhead in high-mobility solutions, which is why NDN utilizes the Listen First Broadcast Later (LFBL) protocol [102] in ad-hoc/opportunistic networks. In LFBL, Interests are flooded and when a potential source for the requested information receives an Interest, then it listens to the (wireless) channel in order to discover if another node has already sent a matching Data message. If not, it sends the Data message itself towards the subscriber.

Security

Last, regarding security NDN supports the association of human-readable hierarchical information names with the corresponding IOs in a verifiable way [103]. Each Data message contains a signature over the name and the information included in the message, plus information about the key used to produce the signature. This allows any node to verify the binding between the name of the packet and the accompanying information. In order to verify that the information comes from an authorized source though, the subscriber must trust the owner of the public key used for signing. The hierarchical structure of names simplifies building trust relationships as a name, e.g., `/aueb.gr/ai/main.html`, can be signed by the owner of the domain, e.g. in this case `/aueb.gr/ai` whose key may be certified by the owner of the `/aueb.gr` domain.

2.3.3 DONA, MobilityFirst, i3 and ROFL

Data-Oriented (and beyond) Network Architecture

The Data-Oriented (and beyond) Network Architecture (DONA)[12] [72] is a fully fledged ICN architecture that is based on a joint name resolution and data overlay routing scheme of special Resolution Handlers (RHs) sitting on top of the existing Internet substrate. Compared to the current architecture, DONA replaces the hierarchical URLs with *flat*, *self-certifying names*. These names are composed by *a)* the *cryptographic hash* of the public key of the *principal* (that can be considered as the owner of) the information, and *b)* a information-specific *label*, which allows users to verify that the received information corresponds to the actually requested name. In addition, while URLs refer to specific network locations via the DNS, flat names in DONA can be persistent even if the source of information gets relocated, a fact which also facilitates caching and replication at the network layer.

The architecture can naturally support *on-path caching* at the RHs which are traversed during the process of name resolution in two ways: first, if path-labels are used, in which case the intermediate RHs in the reverse-path to the subscriber can autonomously decide to cache the transited data; and second, in case a RH replaces the source IP address of an incoming resolution request with its own IP address prior to relaying the request to the next RH, thus causing any consecutive data in response to get cached there. No matter the way used, any consequent resolution requests encountered by the caching RH will be responded with data straight from its local cache. However to support *off-path caching* or replication, DONA requires additional registrations of the cached copies or replicas. A RH receiving multiple registration messages referring to the same name will only store and propagate upwards the hierarchy the pointers to the "best" (e.g., the nearest one) copy.

Finally, *mobility* is supported in DONA by updating mobiles' subscriptions with new FIND messages issued from their current new location, while publisher mobility requires additional registrations (either re-REGISTER or un-REGISTER) at the non-negligible cost of extra messaging all the way to the tier-1 RHs.

MobilityFirst

MobilityFirst [104] is another clean-slate Future Internet architecture with an *emphasis on mobility*. Each entity, i.e. either a data object, device or service, has a Globally Unique Identifier (GUID), which can be translated into *many* addresses at various points of the network in order to enable the *dynamic indirection* of messages after a mobile entity. GUIDs are flat 160-bit strings with no semantic structure, which can be translated to human-readable names via a dedicated global naming service [105]. They can be self-certifying IO digests to allow integrity verification, or hashes of public keys

[12] This reference to DONA leaves the details of the name resolution and routing processes for Section 3.2.4 as part of the related work on global name resolution in ICN.

to bind devices to principals, or they can simply be random with a sufficiently long size for collision avoidance. Note that every device must have GUIDs for itself and its data objects or services, which makes it possible for the architecture to support both name-based information delivery and host-to-host communication via device GUIDs. GUIDs get translated to network addresses with a special global NRS: (i) any publisher who wants to make some information available gets a GUID from the naming service and registers it with its network address in the global NRS; (ii) any subscriber that wants to receive some information, sends a special GET message with the desired GUID along with its own GUID for receiving the response to its local Content Router. This as well as the rest of the routers can only route based on network addresses, hence it gets from the global NRS a mapping between the destination GUID and one or more network addresses, or alternatively a (full or partial) source route. Evidently, the resulting name resolution and data routing process is a hybrid between traditional IP routing and name-based routing.

Internet Indirection Infrastructure

The Internet Indirection Infrastructure (i3) [106] is an overlay architecture designed to provide the types of functionality and/or communication which the native Internet architecture lacks. Specifically, i3 provides (i) multicast, (ii) anycast, (iii) service composition and (iv) mobility support, based on *a) infrastructure nodes* and *b) "triggers"*, the latter being a rendezvous primitive used to decouple publishers from subscribers. In order for two or more parties to communicate in i3, they need to refer to a *common ID* for the communication session that is both random and unique. Triggers are composed of a unique ID as well as the IP address and the listening port of the publisher node. In order to receive the data, a subscriber must submit a trigger that will be eventually handled by one of the infrastructure nodes. Likewise, a publisher inserts her data to the overlay using the same mechanism, thus the data reach to the same handling node which in turn does the data forwarding to the subscribers. The former rendezvous function is incarnated via an underlying routing framework that is based on the Chord [71] DHT overlay architecture.

Routing on Flat Labels

Routing on Flat Labels (ROFL) [107] is a routing infrastructure based *solely on host identifiers* and without any notion of location. ROFL adapts a DHT-like routing scheme but without relying on an underlying IP layer. In the intra-domain level, it uses a Chord-like [71] scheme, where each node establishes links to its predecessor and to its successor nodes in the circular ID space. In addition, the nodes cache AS-level routing information extracted from incoming packets. In the inter-domain level, however, ROFL adapts the Canon [108] hierarchical DHT scheme to allow the establishment of inter-domain paths that respect the established inter-domain routing policies. Being a

DHT-based solution, however, the Canon scheme comes at a significant cost: it implies highly stretched paths which require a dramatic increase of state size to improve routing.

Chapter 3

Literature review

This chapter presents the work in the literature that is related to *a) proactive caching* approaches for enhancing *Mobility Support* and *b) ICN Name Resolution System* designs. Specifically, Section 3.1 discusses the state-of-the-art on proactive caching approaches in correspondence to our own proactive models presented in Chapters 4 and 5. In addition, Section 3.2 discusses the necessary background for the part of this dissertation that is dedicated to scalable ICN NRS designs. We assume that the reader is expected to be familiar with the essentials of ICN that are provided in Chapter 2.

3.1 Proactive Caching approaches for enhancing Mobility Support

Caching is a well-known and widely used technique for improving the performance of future requests based on the *locality of reference* [109], with one of its uses applying to mobility support. For instance, in Pub-Sub event notification systems such as SIENA [110] or JEDI [111] that are designed to deliver publications to all subscribers, caching enhances reliability in dynamic mobility scenarios where users leave and join the network on a frequent basis by guaranteeing the delivery of all the (cached) IOs that were published during mobile disconnection periods to all the mobile subscribers.

Proactive caching, in particular, has more recently triggered much of attention for enhancing mobility within the scope of Pub-Sub architectures like ICNs, as well as beyond Pub-Sub (particularly) for wireless Wi-Fi and cellular networks. The core idea lies on *pre*-fetching and caching data locally along the lines of previously investigated solutions proposed from at least since the 1990's for supercomputer instruction caches [112] and as a performance enhancement [113, 114, 115, 116] for file and Web access systems for which it can improve latency up to 60% in contrast to only 26% (at best) with mere caching [116]. The idea has been also revisited during the first half of the 2000's for mitigating bottlenecks with respect to DNS records [117] and spatial queries [118], before it finally surfaced again for mobility support.

Depending on the aspects of the underlying network architecture such as wired or wireless, homogeneous or heterogeneous, Pub-Sub like in ICN or traditional host-centric communication over TCP/IP, and the mobility scenarios and the application requirements (mainly with respect to mobility frequency, mobility intermittency, sensitiveness to delay, sensitiveness to monetary aspects and QoS demand), proactive caching can be aimed on any combination out of following:

(i) download delay reduction;
(ii) reliability assurance;
(iii) QoS levels guarantee, e.g. certain bandwidth levels and/or low delay jittering and the implied improvement of user QoE qualitative aspects;
(iv) monetary cost reduction for both providers and users.

3.1.1 Publish-Subscribe networking environments

Within the context of Pub-Sub systems and, by extension, ICN architectures, *(i) proactive caching* represents one out of three caching-based paradigms for enhancing mobility support that are identified in [119] based on *when* and *where* the subscriptions and/or the corresponding IOs are prefetched and cached, with the other two being *(ii) reactive* approaches and *(iii) durable subscriptions*.

With proactive caching solutions [119, 120, 93], mobility is enhanced by prefetching and caching the IOs which match a mobile's subscriptions in next hop distance "neighbouring" proxies, before the mobile disconnects. This way, proactive approaches trade-off buffer space for reduced delay in forwarding IOs to mobile subscribers after the latter complete a handover. The aforementioned *proxies* are special in-network entities that are close to or collocated with mobile APs, and are responsible for handling the subscriptions of all the mobiles attached to some AP within their authority. The process of IO prefetching can be done either immediately to the next hop neighbours of some proxy which a mobile associates to; or by prefetching only the subscriptions of the mobile first and issuing them afterwards, right before or right after the mobile detaches from its current AP and certainly before the mobile attaches to its destination AP. Therefore, when the mobile associates to one of these proxies, it can quickly receive its desired IOs that were proactively transmitted during its disconnection.

The selection of the proxies to proactively cache the IOs can be based on prediction as suggested in [93, 120][13], or on the knowledge of all neighbouring proxies lying one hop ahead as in the work of [119]. The work of [120] does not propose a specific prediction algorithm, whereas the one of [93] uses information from past handovers and the one of [119] proposes the caching of all the IOs matching some mobile's subscriptions when the latter disconnects from its AP.

Regarding the other two paradigms, when a mobile disconnects from its current proxy in the case of reactive caching approaches [121, 122, 123, 124], the proxy keeps caching the IOs that match the subscriptions of the mobile. When it reconnects, the mobile informs the new proxy of the old proxy's identity, and the new proxy requests from the old proxy all IOs that have been cached during the disconnection period. This reactive procedure has the disadvantage of *increased delay* for the new proxy to start forwarding IOs to the mobile, since the proxy must receive all the IOs cached in the old proxy before it can start forwarding. This delay can be avoided with both durable subscriptions and proactive caching solutions. However, with durable subscriptions [125], proxies maintain a mobile's subscriptions and cache the IOs which match these subscriptions *independent* of whether the mobile is connected to the proxy or not. Without any additional mechanisms, in order to avoid loosing information, all proxies within a domain that a mobile can possibly connect to would need to maintain subscriptions and cache matching IOs. Note though, that from all three paradigms, the approach with durable subscriptions incurs significantly more memory costs.

3.1.2 Wi-Fi and Cellular Networks

Proactive caching for mobility support has been proposed for context caching for the purpose of fast handovers by Mishra et al. [126] and for vehicular Wi-Fi access by Deshpande et al. [52]. Moreover, Pack et al. [127] propose a selective neighbour caching approach for reducing the handover delay in Wireless LANs (WLANs). The motivation in the above work lies in that, even when the number of neighbouring APs is large, at most 3 or 4 are targets of the handoffs. Although a similar motivation can apply to ICN architectures, the application of such an idea for supporting mobility in ICN is different

[13] A more complete discussion on mobility prediction is provided in Section 5.5.2 on page 111.

due to the different nature of the problem: the objective in ICN is to forward to mobiles the IOs that match their subscriptions, whereas in [127] the objective is to proactively send a mobile's context to neighbouring APs in order to reduce association delay. In addition, the approaches of [128, 129] improve QoS support during handovers in cellular networks based on exploiting mobility information such as the probability of a mobile connecting to future APs, in order to undertake cache actions that reduce handover delays in Wi-Fi and cellular networks.

As mentioned earlier, the authors of [116] state that pre-fetching and caching can improve latency by up to 60%. On the down side however, if the pre-fetching policy fails to predict the user's next requests, then it wastes valuable network bandwidth and cache space. To address this problem, there are proposals in the literature that exploits users' access patterns and mobility in order to pre-fetch data for future content requests [52, 53, 130] . Whereas the work by Dandapat et al. [130] assumes that mobiles' routes are known, Deshpande et al. [52] as well as Nicholson et. al [53] use mobility prediction instead.

3.1.3 Popularity-based & Mobility-based approaches for Mobile Video

Many recent research efforts adapt the idea of proactive caching of *popular* content, and *video* in particular, to small cells and/or Wi-Fi hotspots (i) as a remedy for *backhaul bottlenecks* which hinder *offloading* [131] of macro-cellular traffic to such cells, and (ii) in order to exploit the higher speed and lower power consumption of short-range needs of short-range wireless transmissions. Besides these popularity-based solutions, there are proposals that focus on *mobility* prediction or combine mobility prediction to content *popularity* information for taking proactive actions. These solutions can be categorised in those that *a) push content to mobiles* and those which *b) cache* content *proactively* in small cells or Wi-Fi hotspots. Note that the vast majority of these solutions, which we discuss next, focus particularly on mobile *video* due to its increasing popularity and its requirements for QoS and high throughput.

Pushing content to mobiles

The work by Bastug et al. [132, 133] exploits context-awareness and Social Networks to predict the set of influential users to which they proactively cache strategic contents in order to be further disseminated to their contacts via Device-to-Device (D2D) communication. The authors also explore proactive caching in small cells during off-peak hours based on popularity, correlations among users, and files patterns. Note that the work of [132] employs perfect knowledge of popularity while in the work of [134] makes a quick reference on content popularity estimation, showing that a good estimation of content popularity is a (hard) issue on its own. The works of [58, 135, 136, 137] which we discuss later, analyse the properties of requests popularity on loca and global .

Likewise, the work in [138] by Gonçalves et al. uses D2D communication to exploit predictable demand with proactive data caching and in order to minimize user payments by trading proactive downloads. The solution yields mutual benefit for carriers via dynamic pricing for differentiating between off-peak and peak time prices.

Malandrino et al. [139] propose proactive “seeding” (i.e., pushing) of content to mobiles in order to minimize the peak load in cellular networks based on a content-spreading prediction over Social Networks approach. The same idea, only from an energy consumption prospective, is explored by Göngür et al. [140] who propose proactive content caching to devices in order to reduce energy consumption by increasing the total transmission time of a request and by downloading it during better channel conditions rather than at the time of use.

Last, *centrality* measures for content placement is used in [141], a game theoretical formulation of the data placement (caching) problem as a many-to-many matching game is given in [142], and proactive caching with perfect knowledge of content popularity is investigated in [132].

Caching in Small Cells and Hotspots

Golrezaei et al. [143, 144] suggest proactive caching of popular IOs to small-range Base Stations (BSs) called “helpers”, each of which is equipped with a large cache-storage and high wireless Wi-Fi capacities. Their solution is aimed against the capacity bottlenecks in the backhaul of the helpers for the purpose of *offloading* wireless traffic from a macro cell’s BS to the helpers. The solution focuses on video on-demand streaming scenarios from Internet-based servers were offloading can improve the area spectral efficiency of video transmission in HWNs by handling mobiles’ video traffic via short-range links to the nearest helpers. Note that the work of [143, 144] shows that optimum IO assignment is NP-hard. This outcome coincides with the one in [145], which refers essentially to an instance of the Data Placement Problem[14] [146] where users can be satisfied by multiple resources, i.e. in this case by any of the APs with overlapping coverage within the range of the user in dense WLAN scenarios.

Similarly to [143, 144], the work by Zhang et al. [54] proposes to exploit cache storages at wireless APs based on content popularity in order to bring content closer to mobiles to improve downloading performance. Going a step further than [143, 144], the proposed solution also employs a separate dedicated buffer for prefetching content in order to adapt better to temporal locality conditions due to the mobility of requesters by capturing short-term content access patterns. An interesting conclusion of this work is that it is insufficient to rely on the past history of a device in order to predict when and where to prefetch IOs, particularly in urban environments. Therefore, the authors propose a prediction model based on aggregated network-level statistics.

In the same direction as the above works, is that of “MobiCacher” by Guan et al. [147]. In order to overcome the lack of sufficient backhaul capacity to connect small

[14] We discuss the Data Placement Problem more on page 52, where we argue about its different nature and corresponding unsuitability of the proposed Data Placement Problem solutions compared to local, distributed caching in in mobility scenarios.

cells to the core network, the authors try to utilise the effective application layer semantics of both spatial and temporal locality so as to increase local cache hit ratios at small cells and, hence, to avoid incurring backhaul traffic. For that, they formulate a mobility-aware content caching optimization problem (again related to the Data Placement Problem, hence being NP-complete) with the objective to maximize the caching utility with bounded approximation ratio. The proposed solution is a polynomial-time heuristic which considers user mobility patterns. Note however, that unlike our own work and the work of [54], MobiCacher assumes that user content requests and mobility patterns are known a priori.

The work by Gomes et al. in [148, 149] proposes a model approach along with different content migration preemptive strategies for ICN caches that reside at the edge of Long-Term Evolution (LTE) mobile networks. The main idea is to optimize preemptive migration based on requests by nearby users, with the goal to deliver content faster to end users, and at the same time to efficiently use bandwidth and storage resources lying in the core of the network. The approach emphasizes on caching content at the edge that is popular to *local users*, i.e. it tries to capture locality in space and time as in the aforementioned works of [54, 98, 55].

Poularakis et al. [150], focus on the local caching of popular files at small cell BSs to reduce the traffic incurred when transferring the requested content from the core network to the users, and propose a caching model that considers the fact that an operator can serve the requests for the same IO that occur in short timescales via a single multicast transmission. The aim of this solution is to transmit the IO data only once rather than with multiple, hence costly, unicast transmissions. In a more recent work [151] by Poularakis et al., the authors introduce an optimization framework for modelling user movements through random walks on a Markov chain. The framework minimizes the load of the macro cell via a distributed caching model at local BSs that exploits mobility prediction and information-mixing methods based on the principle of *network coding*. Note, that this work focuses on realistic networking conditions where mobiles connect intermittently to multiple BSs and download only parts of the requested IOs, leaving only the ones that failed to be delivered on time via the local BSs to be transferred through the macro-cellular link.

In [152], Borst et al. define two different types of cooperation: intra-level and inter-level, where IOs can be delivered only by other peer nodes or only by parent nodes, respectively. The authors formulate the Data Placement Problem as a linear program in order to benchmark the globally optimal performance, showing that under certain symmetry assumptions the optimal solution has a rather simple structure. Moreover, they propose cooperative cache management algorithms that try to maximize the amount of traffic served from the distributed cache space and to minimize the bandwidth cost. Note that the cache cooperation scheme in the work of [98] allows both intra-level and inter-level cooperation simultaneously.

Taking a different approach from all the above, the work of [153] by Tamoor et al. focuses on storage-bandwidth trade-offs using the probability of not satisfying requests over a given coverage area as a function of signal-to-interference ratio, cache capacity, small cell density and content popularity.

Finally, within an exclusive ICN context, the work of [154] by Kanai et. al proposes a proactive content caching scheme designed for mass transportation systems that provides high-quality and highly reliable video delivery services by placing NDN-based content servers in mass transportation stations. The latter cache DASH video segments proactively, before the transportation vehicles (e.g., trains) arrive at the stations. Zheng et al. [155] work on mobility-aware caching for ICNs to improve QoE, and propose a proactive caching approach with redirection that enhances seamless mobility support and QoS levels. To this end, the authors formulate an integer linear programming problem aiming to derive the optimal balance between content caching and redirection.

3.1.4 Cost-based Web Cache Replacement

In what follows, we discuss two prominent cost-based, web cache replacement algorithms which constitute related work with respect to the cache replacement extension (see discussion on page 83) adapted by the EMC-R and EMPC-R proactive models presented in Chapter 5. GreedyDual-Size (GDS) [58] by Cao et al. is a cost-aware algorithm that integrates locality information as follows: Cached objects are given a score value based on the cost of bringing them into the cache. When a replacement needs to be made, the content with the lowest cost score H_{min} gets replaced and the rest of the objects *reduce* their score values by H_{min}. Cost scores get only restored upon new requests, thus score values tend to become lower for content that is not accessed for a long time. By reducing score values as time progresses and by restoring them only upon a new request, the algorithm manages to seamlessly integrate cost concerns and requests locality in time.

Jin and Bestavros [136] analyse temporal locality beyond the properties of content popularity skewness and generalise GDS. The resulted GreedDual* algorithm uses a utility value u in the place of the cost per data unit for fetching an object that is used in GDS, along with a cache-ageing factor L as follows: Objects are assigned with $u + L$ upon cache hits, while L gets the value that was assigned to the most recently evicted object. Hence, L works as an *inflation* factor for cached objects in order to reflect the importance of locality due to temporal requests correlation, versus the importance of long-term content popularity.

3.2 Name Resolution in Information-Centric Networking

As discussed in Section 2.2.2 on page 21, there are two major design paradigms for ICN NRSs, namely, the *a) lookup-by-name* approach and the *b) route-by-name* approach. The earlier designs resemble that of large-scale distributed databases, as they maintain mappings between IO *names* and their available source location(s), while the latter function by routing resolution requests straight to the location of the named IO itself. In what follows, we first summarise the outcome of studies in the literature which justify why DNS can *not* meet with the requirements of an ICN Future Internet architecture (Section 3.2.1). Then, we provide the needed background on DHTs for understanding the fundamentals of lookup-by-name approaches that are based on DHTs, mainly their overlay routing function and their scalability properties (Section 3.2.2). Finally, we get into the heart of this section by reviewing the most important lookup-by-name (Section 3.2.3) and route-by-name (Section 3.2.4) name resolution solutions in the ICN literature.

3.2.1 Extending the Domain Name System

The DNS is the closest existing equivalent to a lookup-by-name NRS and, therefore, it has been naturally explored as a possible option for ICN as well. Towards this direction, an extension of DNS for ICN was proposed in [156]. However, that work did not delve deeper into the scalability properties of the resulting design while most studies in the literature recognised a series of disadvantages and limitations which constitute DNS as *inappropriate* for the purposes an ICN NRS.

To begin with, DNS is susceptible to security attacks [157], particularly DoS attacks as a result of (i) the limited redundancy of its name-servers, and (ii) the fact that many servers have a single point of attachment to the Internet [158]. Furthermore, DNS adapts a namespace that is both semantic and hierarchical, and which embeds information about organisations, administrative domains, etc.Therefore, it couples identity to location and (may) reveal internal domain organisational aspects, whereas the simplest way to achieve the name persistence and the increased security that is required in ICN is via flat, semantic-free identifiers as justified in [65, 72, 159].

Besides security considerations, many ICN approaches integrate IO dissemination strategies or policies to name resolution, such as in the case PSI [64]. However, the DNS is *not* a mapping service for delivering policy-based information as discussed in [160]. Moreover, the existing mapping between name prefixes and resolution servers in DNS can not address the load expectations in ICN: first, because there may be a huge number of ICN prefixes that will need to be resolved anywhere on the network instead of a static set of root name servers and, second, because the expected resolution load will refer to individual IOs rather than to the servers hosting them, thus being orders of magnitude higher than what is currently encountered by DNS.

To make things worse with respect to the scalability properties of DNS, the implied resolution load is highly *imbalanced* between the DNS root servers because the names are not equally distributed among top level naming domains. For instance, the number of names that fall under "`.com`" exceeds by far that of the names that fall under "`.edu`" and so forth. This imbalance is "convenient" with respect to the role of the DNS in the current Internet as it boosts DNS name aggregation, with many names being able to aggregate under a few popular common prefixes. Yet as we argue in Section 2.2.1 on page 19 on our point about hierarchical naming and aggregation, the benefits of aggregation are debatable in ICN as the name space may involve a huge number of prefixes with unknown, arbitrary popularities.

3.2.2 Distributed Hash Tables

Distributed Hash Tables (DHTs) are decentralized distributed systems that provide lookup services similar to a hash table, with key/value pairs being distributed in the nodes that form the DHT. By nature, they exhibit some desired properties towards the design of an ICN NRS: their system architecture is highly *robust* and *resilient* with changes in the set of participant nodes and "churn" (i.e., the frequent and continuous arrivals, departures, and failures of nodes) causing only a minimal disruption, while the overlay network can *scale* to very large numbers of participants and amounts of resolution (key/value) state due to the good load balancing properties of DHTs. Nevertheless, DHTs schemes are "*flat*"[15] and imply inefficient overlay routing with respect to network proximity, administrative boundaries and the overall BGP inter-domain routing policies, caused by the lack of adaptation to the underlying physical network topology.

In what follows, we outline the basic design concepts of the Pastry [5] distributed object location and routing substrate, the Canon [108] paradigm for adapting a *hierarchical* DHT design and, finally, the fundamental concepts of the H-Pastry hierarchical DHT.

Pastry

Rowstron et al. proposed Pastry in their work of [5], a DHT architecture that unlike other DHT schemes is tailored to adapt to network locality with respect to some employed proximity metric (e.g., hop count, RTT, etc.) during routing state creation and maintenance. This important feature yield Pastry's property to have short routes and constitute one of the reasons for using Pastry as the key-based routing substrate for many other applications. Each Pastry node maintains routing state that enables the forwarding of messages using prefix-based routing: at each routing step, a node forwards the message to another node, whose identifier *shares at least one more digit* with the target IDentifier (ID). IDs are 128-bit strings handled as a sequence of b-bit digits, were b is a configuration, typically 4. The routing state is maintained in two separate structures, the `Routing Table` and the `Leaf Set`. Each entry contains the ID and the

[15] We adapt the term "flat" for the originally proposed DHT schemes to reflect that their design is agnostic to the hierarchical inter-domain relations in the physical underlay.

network address of a node in the system. The `Routing Table` is an array with size $(2^b - 1) \times \frac{128}{2^b}$, with each row i containing entries for nodes whose IDs share only the first i digits with the present node. The `Leaf Set` contains $|L|$ entries (typically, $|L| = 2^b$) for nodes whose IDs are the closest to the present node's ID. The set is split in two parts with $|L/2|$ entries for numerically smaller IDs and $|L/2|$ for the remaining larger IDs.

The Canon paradigm

The work of Ganesan et al. [108] emphasizes on the importance of the DHT adaptation to the underlying network structure, thus proposing the *Canon DHT paradigm*. Canon intervenes in the construction process of a DHT to enable the progressive merging of individual DHT constructions, assuming an underlying hierarchical inter-domain topology. First, each domain creates its own DHT structure. Then at each higher level, the DHT structures of sibling domains are *merged* so that a single DHT structure is created. This process continues recursively up until all individual DHTs have been merged, always by satisfying the following two requirements: the *a) inter-domain paths always convergence*, i.e. all messages originating from the same domain A and targeting the same node located at a different domain B, always exit A via the same node, and that the *b) locality of intra-domain paths* is respected, i.e. all overlay routes between nodes of the same domain *never* exit the domain.

Hierarchical Pastry

Fotiou et al. proposed Hierarchical Pastry (H-Pastry) [161] as a multi-level DHT scheme that combines the benefits of its flat DHT design-base, *Pastry*, with respect to network topology adaptation, administrative structure and routing policies, further enhanced by the *Canon* paradigm. H-Pastry can support inter-domain multihoming and peering relationships and yields shorter overlay paths compared to Pastry, which make it ideal as the look-up routing substrate for DHT-NRS [89].

We leave the details of the design and the performance advantages of H-Pastry for Section 6.3.1 on page 123, where we discuss in further its design aspects in detail with an emphasis on how its complex routing scheme can adapt multihoming and peering inter-domain relations.

3.2.3 Lookup-by-name approaches

MDHT and HSkip

Dannewitz, D'Ambrosio et al. were amongst the first to propose a lookup-by-name approach in their work of [7] where they present Multi-level Distributed Hash Table (MDHT). Multi-level Distributed Hash Table (MDHT) is essentially a multilevel DHT that aggregates IO registrations at the higher levels of the inter-domain hierarchy. MDHT uses an indirection mechanism to resolve content provider names and leaves the resolution of the actual content to be done at a lower level. In addition, the use of a hierarchically extended version of the SkipNet [162] overlay, namely HSkip, has also been proposed in [67] to achieve lower resolution delays and overal system latency.

Name-based inter-domain routing

Likewise to MDHT, the work of [90] by Rajahal et al. proposes an inter-domain rendezvous design that combines policy-based name routing between adjacent networks via hierarchical interconnection overlays for the purpose of scalable global connectivity. This hybrid design uses indirection to map scopes of information to lower level resolution nodes. Nevertheless, the it leaves the routing inefficiency of DHTs to be addressed via the aggregation of information at higher levels of the inter-domain structure, which in turn raises scalability concerns.

A tree *vs.* a DHT-based NRS

The work of Choi et al. [163] puts its focus on NRS design alternatives with respect to (i) how to content is located, (ii) how to content is cached and (iii) how to content is delivered via a comparison between a hierarchical DONA-like architecture and a DHT-based alternative focusing on aspects related to content delivery such as transfer latency, robustness, and the impact of in-network caching. Note that the later mentioned work of [89] also presents a comparison study between the proposed DHT-NRS and DONA. Only unlike the simplified tree topologies used in [163], the comparison there assumes more realistic topology models that consider multihoming and peering relationships between ASs.

Secure Resolution of End-Host Identifiers for Mobile Clients

Varjonen et al. [164] present a DHT-based mapping/resolution system that serves the decoupling of mobile node identity and location, focusing on security aspects and the impact on the achieved response times. The work neglects the routing overheads and focuses its discussion on the features needed to resolve flat identifiers to locators in a secure manner.

DHT-NRS

Finally, the work [89] by Katsaros et al. proposes Distributed Hash Table-based Name Resolution System (DHT-NRS) within the context off the PSI ICN architecture. We leave the details regarding the the rendezvous functionality of DHT-NRS for Section 6.3.2 on page 126. In a nutshell, DHT-NRS tries to address the scalability concerns of the previously referred DHT-based efforts by adapting to the hierarchy of the underlying network topology. To do so, DHT-NRS is built upon the DHT overlay routing substrate of H-Pastry (Section 3.2.2) which adheres to the Canon [108] paradigm, thus it considers the physical network proximity, the boundaries of the administrative domains and the corresponding inter-domain routing policies. However, the simulation-based evaluation results presented in Section 6.3.2 on page 125 denote that DHT-NRS can *not* fully eliminate some of the intrinsic problems of DHTs, manly with respect to path-stretch and BGP policy violations.

3.2.4 Route-by-name approaches

Data-Oriented (and beyond) Network Architecture

The process of name resolution in the Data-Oriented (and beyond) Network Architecture (DONA)[16] is based on an overlay of special Resolution Handlers (RHs), the organisation of which follows the one of the AS-level structure of the Internet. Each AS has its own logical RH that is incarnated by at least one or more physical RHs. The resulting RH topology matches the corresponding customer-to-provider and provider-to-customer relations, and it is further enhanced with peering links following the peering domains of the AS topology. Due to these peering links, the RH topology is not strictly hierarchical, as the peerings introduce cycles in the inter-domain graph.

Regarding the process of *name resolution* itself, publishers advertise with special REGISTER messages the availability of their content or service to the nearest RH, which in turn passes the advertisement to the rest of the RHs in the overlay along the lines of the inter-domain forwarding policies in the underlay. As the message propagates, it sets up the name resolution state throughout the network in the form of <`name, next hop RH`> mapping pairs between advertised information names and the previous RH. The propagation of the REGISTERs terminates at the Tier-1 RHs. ASs which are not willing to act as transit domains do *not* propagate any incoming REGISTERs over their peering links. Likewise, the interested hosts request for their subscription to some published content or service with a special FIND message to their nearest RH. The rest is done by the RH overlay network which matches the request in the closest RH with an advertisement entry and initiates the data transfer via *anycast*, either from an original source, replica or cached copy. Note that as with the REGISTER messages, FIND messages can also reach up to Tier-1 ISPs were, ultimately, *resolution is guaranteed*: all Tier-1 ISPs have peering links to each other, hence their RHs are aware of all the registrations published in their direct and recursive customer serviced ISPs.

Regarding the case of *multihomed* and *peering* links to the domains of other ASs, multihomed hosts submit REGISTER messages to the corresponding local RHs and multihomed domains forward their REGISTERs to each provider, thus enabling the option of multiple paths for consequent FINDs. In case of an inter-domain peering relation, the RHs process the received REGISTERs and FINDs in accordance to some local policy that is consistent with their domain's peering agreements. This statement in the design of DONA translates to the following: if two domains have a peering agreement, then their RHs propagate their REGISTERs and FINDs to their *direct* peers; yet, if the domains are more than just peers, i.e. if they are *siblings*, then they practically behave as a single

[16] Recall that an overview of the rest aspects of DONA is provided in Section 2.3.3 on page 30

domain according to the definition of a sibling domain in [165], which implies that the RHs further relay the REGISTERs and FINDs received from their singling domains' RHs to their own peers and providers[17].

By exploiting the above resolution design, DONA can support both a decoupled and a coupled data routing scheme: *a) decoupled* via standard IP routing to the requester's IP address, and *b) coupled* by using the reverse source-route described in a so-called "path-label" carried by the FIND message. The latter is merely the sequence of the ASs traversed by the FIND message until reaching to the resolving RH. Nevertheless, it implies "symmetric-routing" which commonly does not conform with the established BGP relationships between ASs, unlike the decoupled routing approach which adheres fully to BGP.

Content-Ubiquitous Resolution and deLivery Infrastructure for Next Generation services

The Content-Ubiquitous Resolution and deLivery Infrastructure for Next Generation services (CURLING) [73] shares many fundamental design ideas with DONA on name resolution and with NDN (Section 2.3.2 on page 27) on data routing. But unlike DONA, CURLING is not a complete ICN architecture but a coupled name resolution and routing alternative[18] design for the COntent Mediator architecture for content-aware nETworks (COMET).

Regarding name resolution and its resemblance to DONA, CURLING is a coupled name resolution and routing approach that is also designed to adapt to the inter-AS topology structure. There is, however, a significant difference with DONA: its equivalent of RHs, referred as Content Resolution Serverss (CRSs), propagate registration and subscription requests *only to their provider* ASs. This implies that both types of messages do *not* propagate over peering links, excluding the requests that reach all the way up to the CRS of a tier-1 AS, which have to be *broadcasted* to the CRSs of all the other tier-1 ASs to guarantee resolution. Note though that CURLING does utilise the AS peering links for content delivery itself. In addition, CURLING also offers scoping and filtering features: a subscriber may explicitly limit the propagation of its message to a specific network area or exclude specific network areas from this propagation, hence it can define which ASs can act as its information source(s).

With respect to data routing and its resemblance to NDN, while both name resolution and data routing in NDN use the same nodes, i.e. the CRs, name resolution in CURLING uses the aforementioned CRSs while data routing uses dedicated Content-aware Routerss (CaRs). This enables a more flexible choice of the best paths between the available CaRs within a AS by the CRSs of the AS.

[17] The reader should note the importance of peering and sibling relations in the design of DONA, as well as for the design of the CURLING NRS design. These relations pose significant implications to scalability in both DONA and CURLING. The reader may refer for more on Chapter 6 were we present our related research on this subject.

[18] The other one being a decoupled name resolution and routing scheme which is out of the scope of this dissertation. The reader may refer to [70] for a detailed presentation of the COMET architecture.

Chapter 4

Efficient Proactive Caching for addressing Unique Mobile Demand under strict Delay requirements

This chapter is based on our published work in [48, 93, 94, 98, 166]. It discusses an Efficient Proactive Caching (EPC) distributed model for addressing *unique mobile demand* for users running applications with *strict delay requirements* in *edge network mobility* scenarios. This corresponds to our thesis point ((1)) on page 11 and to contribution ((1)) on page 12, and is related to the general background discussion in Section 1.1.3 on page 8 and to the related work discussion in Section 3.1 on page 34. EPC exploits individual user mobility information and efficiently utilizes cache storage using a congestion pricing scheme in order to prefetch IOs to dedicated *cache points* near to or collocated with mobile APs. Corresponding proactive actions take place on a per-user request basis, following a simple, lightweight and fully decentralised process of autonomous local decisions at neighbouring caching points. The applied cache congestion pricing scheme allows efficient utilisation of the local cache-storage resources and can have advantages over the centralised, mathematical optimisation approximation approaches used in other proposals in the literature for reaching closer to an optimal allocation of the distributed cache resources, mainly, due to being lightweight and due to EPC taking short time-scale online cache decisions that reflect temporal locality. Finally, we evaluate EPC in edge network mobility scenarios without fall-back connectivity, in which case EPC increases reliability apart from reducing delay.

4.1 Chapter outline

This chapter is organised as follows: In Section 4.2 we provide the reader with the needed background and motivation for addressing unique demand in edge network mobility scenarios. Our discussion focusses on the importance of utilising user context as well as of unique mobile demand within the boundary of a LAN on page 49, in order to address application- and user-specific requirements. Then, we outline the fundamental concepts of our proposed proactive caching solution on page 50 and explain how our approach is different from conventional caching and the Data Placement Problem on page 52, before closing the section by summarising the differences and advantages of our solution compared to other proactive models in the literature on page 51 . Next, we get into the details of our system model in Section 4.3 and discuss how our model can be applied in both flat cache configuration scenarios and in two-level hierarchies on page 54 and on page 57, respectively. Note that the discussion there assumes the cost of transfer delay is independent of the transmitted IO size, whereas the analytical discussion that follows in Section 4.4 as a function of transmitted IO size. In Section 4.5 we discuss how EPC can fully integrate into ICN architectures and conclude this chapter with our performance evaluation in Section 4.6.

4.2 Background and motivation

4.2.1 Need for enhancing Mobility Support via User Context[a]

Native (e.g., ICNs) as well as overlay Pub-Sub architectures can seamlessly support subscriber host mobility due to their Pub-Sub communication primitives which allows them to be both *receiver-driven* and *asynchronous* in nature. Prospective IO receivers issue a *subscription* that refers to their desired IO after its *name*, regardless of whether the latter is *published* at the time of subscription or not. Ultimately, when the IO gets published, the pending subscription gets resolved inside the network and the IO data start to get transferred from some source to the current network location of the subscriber with provision of anonymity. In addition, the data transportation from the resolved publisher(s) to the subscribed receiver(s) is performed without the use of any location-dependent address. This *connectionless/stateless* manner of communication is evidently mobile-ready as mobile subscribers can perform handovers to arbitrary network locations and continue to receive the data of their IOs by simply resubmitting their subscriptions. Besides this, Pub-Sub communication is intrinsically *multicast*, which allows to natively and widely distribute the requested IO data around the network "areas" of the mobile subscribers in order to handle large mobile populations.

Nevertheless, Pub-Sub mobility support may be not enough in particular scenar-

[a] This section is based on our detailed discussion on the provided mobility support over the current TCP/IP architecture in Section 1.1.3 on page 8 and the architectures which follow the ICN design paradigm in Section 2.2.4 on page 22.

ios which involve mobile applications with special and/or strict requirements. Increased *delay* for receiving data in Pub-Sub network environments can be incurred when a mobile sends a subscription request for some IO but then disconnects and moves to a another AP prior to receiving the requested data. Delaying, or even failing to deliver the publications issued while the mobile subscribers were in transit to another AP, can significantly deteriorate the application's QoS and reliability requirements. Another issue regards *Information* & *software* mobility which should be expected to be far more common than today within the context of ICN architectures, in particular, and a Future Internet flourished with cloud services, in general. Due to their incorporeal nature, such entities can escalate the problems faced with respect to publisher mobility support (Section 2.2.4 on page 22) by moving frequently, intermittently and in ways not associated to physical mobility: mobile transitions can be rapid or even instant, including handovers to (physically) unrelated locations anywhere on the network, rather than simply to APs lying at the edge of the network. Last, the intrinsic multicast and caching advantages of ICN, which can be (in theory) leveraged to enhance mobility, have disadvantages. Blind multicast and/or caching in all possible network regions that the mobiles can move to wastes the limited bandwidth and cache buffer resources. This can be further aggravated by unique/individual mobile demand[19] discussed on the facing page. This type of demand turns multicast into multi-unicast with implications on network bandwidth consumption, as well as with respect to cache usage as it requires a greater cache consumption compared to (more) popular demand shared by multiple users.

Regarding the legacy TCP/IP architecture of the Internet, contemporary applications manage to hide the implied ID/location coupling with application-specific approaches on the application layer, as we argued on page 9. Therefore, what remains to be addressed in this case as well as in the case of the Pub-Sub-based ICN architectures is the *particular application requirements* along with the *particular user needs* through *improving* the available *QoS* levels and, hence, the corresponding user-perceived *QoE*. To do so, we need solutions that focus on the quantitative and qualitative aspects of network conditions (e.g., bandwidth or latency from remote sources) and demand characteristics. The former can be done by predictions (e.g., based on historic data) or via the use of online tools like ping, while the latter by means of exploiting *users' context* such as age category or IO preferences, and particularly user *mobility information* and behaviour: e.g., individual or group mobility patterns, predicted or known routes via a GPS navigator. Last, to give a complete picture on the significance of the character of mobile demand, part of it is (or appears to be) *unique* to the individual mobile requesters attached at the network edge (see detailed discussion next in Section 4.2.2), thus it seeks for a special treatment in order to meet with application requirements such as the aforementioned ones.

[19] Note that software mobile agents are more likely to request for unique IOs.

4.2.2 Unique Mobile Demand and its importance

The recent study of [167] regarding unique[20] video request characteristics in YouTube observed at an edge network over a period of 20 months, showed that: *"71% of the requested videos are "one-timers" that are requested only once from the edge network [...] demonstrating the need for selective caching policies",* while in another part of the same study the authors state that: *"For less popular videos (less than 1 M[illion] global views), the majority (70-80%) of the observed videos are one-timers".* Additionally, the work of [168] states that "one-hit-wonders" account for three-quarters of the requested IOs and that by *"Not having to store the one-hit-wonders in cache reduces the aggregate rate of disk writes by nearly one-half".* Unlike the conclusions of [167] which are restricted to video IOs, this latter point applies to arbitrary types of IOs and alongside the conclusions of [167], it indicates a strong trend for user requests which are locally *unique* at the network edge, or at least in practice appear as so due to *data encryption.* Specifically, HTTP Secure (HTTPS), which is adapted by YouTube, Facebook and other large content providers for the sake of privacy and security, prevents the identification of common IO requests based on deep-packet inspection techniques and, hence, even locally popular IO requests may appear to the local network facility as unique. Due to such observations, we can come up with the following conclusion:

> *Unique demand within an edge network accounts for 70-80% of total demand in terms of number of requests and approximately 50% of total demand in terms of data volume size.*

The former justify our motivation for focusing on unique mobile demand in edge network mobility scenarios. We use the term "unique" to identify the part of local mobile demand which is *unique to the individual users* that are attached to some particular LAN at the edge of the Internet. This definition does *not* imply that corresponding IO requests are globally unique, but on the contrary it points out their local uniqueness within an edge network such as in a university campus or in a building. To provide a paradigm scenario, let us assume a set of cohabitants, e.g. 3 or 4 family members, who wish to have access to a home video surveillance system via their mobile devices. Evidently, such demand is *not* globally unique, as there are multiple users who are interested in exactly the same video surveillance content. However, each of these users can be attached to a different LAN during working hours and days such as at different office or school network.

Last, demand *uniqueness* may be specified not only in *(i) space* as explained above, but also in *(ii) time.* For instance, real-time notification data are by definition ephemeral, i.e. they are unique in time, such as in the case of real-time sensor-readings. However, as we note next, this uniqueness in time can be in contrast to reliability application requirements, particularly for the IOs that get published during mobile disconnection periods.

[20] Due to the quoted outcome points of the cited works in this section, we use the terms *"one-timers"* and *"one-hit-wonders"* as synonyms to unique demand requests.

4.2.3 Solution outline

The former background discussion gives prominence to the need for *enhancing* mobility support, irrespective of the underlying network architecture, with solutions tailored for covering the special needs of mobile applications and/or mobile users, particularly with respect to delay. Also, it highlights the importance of unique demand within the context of LANs. Within this context, one approach to enhancing mobility support is via proactive caching. Proactive caching for reducing delay in mobility scenarios has been previously proposed in the literature for Pub-Sub network architectures (Section 3.1.1 on page 34) and for vehicular Wi-Fi access as well as (more recently) cellular networks (Section 3.1.2 on page 35). Proactive caching achieves gains by making the IOs requested by a mobile user *immediately available* when the mobile moves to a new network AP, thus reducing the delay for obtaining the data compared to transferring it from the original source where the data is located. The higher delay for obtaining the data from the source can be due to the larger network distance (number of hops), or because the path to the source includes low capacity links, e.g. in the backhaul of a small cells[21] or Wi-Fi hotspots. Also, as a by-product, proactive caching can increases reliability for application data that are ephemerally available during mobile disconnections periods such as while a mobile is in transit to its next AP.

In this direction, we propose an Efficient Proactive Caching (EPC) distributed model which selects cache points near to or collocated with mobile APs to prefetch IOs corresponding to mobile user requests. With EPC, the cache points proactively fetch the desired IOs based on autonomous decisions after examining each mobile request *individually* based on the requesting user's mobility information and on the expected data transfer delay cost gains in case of a cache hit. Due to this individual treatment of requests, EPC (i) can best fit within the context of Pub-Sub networking environments where users explicitly declare their IO interests via subscriptions, and (ii) is able to capture and *serve unique* mobile demand. Therefore, EPC can be fully integrated on the network layer of ICN architectures or be used as an application-layer solution in traditional TCP/IP network architectures in order to enhance seamless mobility by reducing delay for attaining IOs after a mobile's re-attachment to another AP.

Regarding the ability to treat unique demand, this is an important feature of EPC because unique requests represent a significant amount of demand which can be up to 71% [167] of the requested IOs in an edge network, as previously discussed on page 49. Note though, that unlike the discussed work of [168] which suggests to avoid caching "one-hit-wonders", EPC explicitly targets that unique demand. The work of [168] concludes that *"byte hit rate increases from around 74% to 83%"* by filtering out such demand; nevertheless, this conclusion applies to conventional caching for serving future requests as explored in [168], which naturally meant to targets popular demand consequent requests after past requests. But despite conventional caching, EPC acts proactively to target the current mobile requests which are not yet served, hence showing the difference between

[21] See definition of "small cell" in the Appendix A.1 on page 149.

conventional (popularity-based) caching and mobility-based proactive caching.

A list of representative example applications that one could expect to have benefits from EPC includes home video surveillance, real-time and/or emergency notification services, teleconferencing, online gaming, possible future IoT applications and services (e.g. monitoring aged or young family members, personalised medical applications, etc.), as well as document, streaming or likewise download transferring scenario that is sensitive to delay and/or delay *jittering*. Additionally, EPC is a solution that can also suit the need for lossless, highly reliable data transmission such as for data published during mobile disconnection periods.

EPC is applicable to the case where the requested IOs have different sizes and to a two-level cache hierarchy, which are both hard problems, with the objective of reducing the implied data transfer delay costs from remote/external data sources. In the examined edge network mobility scenarios, this applies mainly to the delay experienced by the mobiles for attaining their desired IOs after they complete a handover to another AP. However, the objective can include other forms of data transfer costs as well. Some examples include the cost of network bandwidth consumption, e.g. in the form of monetary data transit service charges [74] paid by the mobile provider for fetching the IOs from a remote location, or the cost of missing QoS and related QoE goals.

Finally, a novel aspect of the proposed approach is the use of *congestion pricing* that considers the demand for caching and the available storage to efficiently utilise limited cache capacity, while reducing the cost for transferring the requested data to a mobile. The applied cache congestion pricing scheme has important advantages regarding solving the proactive caching problem in a distributed manner. Moreover, it has advantages with respect to addressing the unique mobile demand that implies increased cache requirements relative to popular content demand, as a result of considering each user request individually.

4.2.4 Differences from conventional Caching and the Data Placement Problem

Both the problem and the proposed EPC solution outlined above are fundamentally different compared to *a) conventional caching* and, notably, *b) the Data Placement Problem* [146]. Both of these problems and their corresponding solutions in the literature involve caching (resp., replicating in the case of Data Placement Problem) IOs closer to potential requesters in order to serve future requests based on IO *popularity* information that is either elicited from past requests or predicted via some strategy. But on the contrary, our solution aims at improving the service of *unique* demand in edge network mobility scenarios at the *network's edge*. This has two important implications: First, unique implies that there is no way to utilise cached content from a past IO request. Such IOs are consumed only once, thus when a mobile eventually moves to a new AP, the reserved cache space gets immediately freed with EPC. Second, the acts of EPC involve pulling data from sources and storing it for a *short-term* period after current temporal locality conditions in a local part of the edge network, whereas the Data Placement Problem refers to pushing data to caches after their popularity and storing it for longer. Normally, IO popularity evolves from the moment an IO gets cached, within a period that spans from days to even months [169]. However, the individual mobiles can follow significantly different mobility patterns, irrespective of the IOs that they request. This can alter the locality of requests in space and in time, resulting in a need for individual proactive caching decisions, as done with EPC.

Traditionally, the literature on the Data Placement Problem does not focus on mobility (and notably, edge network mobility) scenarios, as the proposed solutions have a broader context that involves IO replication at various parts of the (inter-)network, rather than at the network edge where the mobiles attach. Moreover, an integral part of the Data Placement Problem problem is that users can utilise any of the available replicas in the (inter-)network and not just one source as in most mobility scenarios, which makes the optimisation problem even harder. Most solutions proposed for the Data Placement Problem such as in [145, 137] use offline mathematical optimisation approximation and do not *not* cover the short time-scale needs of addressing the dynamics of mobility. In fact, Data Placement Problem solutions adapt orchestrated approaches for a long-term content replication in strategic points of the (inter-)network based on offline input, as done in [137]. This particular work adapts long-term and predicted video popularity as it is tailored for video content such as TV-shows, TV-series and blogbuster movies. This as well as other similar Data Placement Problem solutions can be run offline during bigger time-scales in order to approach an optimal allocation of the distributed cache space, but clearly they are not appropriate for addressing mobility demand and, notably, unique mobile demand.

Finally, our proactive caching approach does not perform eviction or replacement when a cache is full, as in conventional caching; rather, our approach uses congestion pricing to ensure that the IOs proactively cached are those for which the highest delay gains are achieved. Interestingly, prior work has found that hierarchical or cooperative caching is not helpful when the user population is above some relatively small threshold [57], which is due to the heavy-tailed content popularity distributions. However, such results are not applicable to the two-level proactive cache hierarchy discussed later in Section 4.3.2, where the decision to cache data in leaf and mid-level cache points depends on the mobility patterns and the gains from reduced transfer delay.

4.2.5 Differences and advantages compared to other Proactive solutions

Compared to the rest of the state of the art in proactive approaches (Section 3.1 on page 34), EPC introduces a distributed model for cache decisions based on localised, short time-scale mobility and demand predictions, and a local cache congestion pricing scheme. The former allow cache decisions to adapt quickly to changes in the mobility model or to changes in the demand model, thus EPC has a good feature with respect to capturing temporal locality.

A second difference is that EPC does not simply aim at improving the service of popular content as the rest of the approaches do, but on the contrary it exploits mobility and individual requests information to capture unique demand, which can represent up to 80% of user requests within LANs. Even though addressing individual requests for large IOs (specifically, videos) could imply a higher cache space requirement compared to caching other types of popular content (web pages, pictures, etc.), still our approach can utilise cache-storage efficiently based on its dynamic congestion pricing scheme.

The third and most striking difference of our model regards the *problem formulation*. Most of the prominent proactive solutions discussed in Section 3.1 use mathematical optimisation approximation for non-tractable distributed cache problems which are equivalent to the NP-hard Data Placement Problem. But as explained on page 52, content popularity-based offline mathematical optimisation approximation solutions are *not appropriate* for addressing mobile demand because they can not meet with the short time-scale characteristics of mobility, particularly with respect to unique demand.

Last, the rest of the solutions are centralised, running in long time-scales, and neglect mobility or employ a static adaptation of content popularity that is *not* conducive to capturing the dynamics of temporal locality. Even the work in [54] that does consider mobility apart from popularity, *statically* splits storage for popularity- and mobility-based caching.

4.3 System model

Proactive caching is used to prefetch an IO s requested by a mobile, so that it is (immediately) available when the mobile connects to its new network AP with a minimum delay transfer cost T_s. The former notation denotes the more generic term: *"transfer"* cost. As we mentioned earlier on page 51 and as denoted later by our analytical discussion in Section 4.4, we emphasise on the case of of *delay* costs due to the assumed edge network mobility scenarios. However, we stress upon the fact that the decision model has a more generic approach to the transfer costs -one of which is delay- traded for buffer space in the cache.

Regarding the valuation of transfer costs, including delay, this depends on the requirements set by the mobile application. For instance, if a streaming application promises no stalling during playback to its users, then a transmission delay or jittered delay causing stop-and-buffering periods induces a high QoE-related cost. By extension, such a cost can be assessed in monetary terms due to, e.g., loosing unsatisfied customers.

For the rest of this analysis, we consider two cases for the transfer delay, which are independent of the IO size: *a) propagation delay* that is independent of the size of the requested IOs, and *b) transmission delay*, which is a function of the IO size:

- **Propagation delay:** In this case, the transfer delay is independent of the requested IO size when the IO size is small, e.g. fire/security alerts. The delay for obtaining an IO from its original remote source is denoted T_R, whereas the delay for obtaining an IO from the local cache is denoted T_L. These delays can depend on the distance to the source or cache, e.g. in number of hops. As mentioned above, apart from delay, T_R and T_L can in general include the cost (e.g. network, monetary) for obtaining data from the source or a remote location, which is independent of the requested IO size. For the above assumption of the transfer delay, Section 4.3.1 describes the proactive caching approach for a flat set of caches, while Section 4.3.2 considers proactive caching in a two-level cache hierarchy.

- **Transmission delay:** In this case, transfer delay depends on the IO size and regards large IOs such as video files. For example, when the mobile is connected to a Wi-Fi hotspot or a small cell, the delay for transferring an IO from a local cache is proportional to the IO's size and inversely proportional to the data rate of the Wi-Fi hotspot or small cell. On the other hand, if the IO is transferred from a remote location over a lower capacity backhaul link (e.g. ADSL) that connects the hotspot or small cell to the Internet, then the delay is inversely proportional to the backhaul rate, which is smaller than the Wi-Fi or cellular rate. As above, the delay can actually involve any cost that is proportional to the IO size, e.g. cost per unit of transferred data in the case of data volume charging. Section 4.4 describes our proactive caching approach when the delay is a function of the requested IO size.

4.3.1 Proactive Caching in a flat cache structure

In this section we present our approach for selecting the caches that should proactively fetch IOs, in the case of a flat set of caches and when the transfer delay is independent of the size of the requested IOs. Under this assumption, as we will see, the IO is either fully cached or not cached. Our objective is to minimize the average delay across all requested IOs, subject to the cache storage constraints. Note that in a flat cache structure, the caches are independent, hence prefetching decisions are also independent.

Let q_s^l denote the probability that the mobile requesting IO s moves to cache l and B_l denote the maximum storage at cache l; we initially assume that all IOs have the same size o. Note that the probability q_s^l can depend on the specific mobile, its current or past location, and the time instant, but for simplicity we do not make this dependence explicit in the notation. Also, let S_l be the set of IOs requested by mobiles that have non-zero probability to move to cache l and L be the set of caches. We define the following optimization problem:

$$\min_{b_s^l} \quad \sum_{s \in S_l} \mathcal{T}_s \tag{4.1}$$

$$\text{subject to} \quad \sum_{s \in S_l} o \cdot b_s^l \leq B_l\,, \ \ \forall l \in L, \tag{4.2}$$

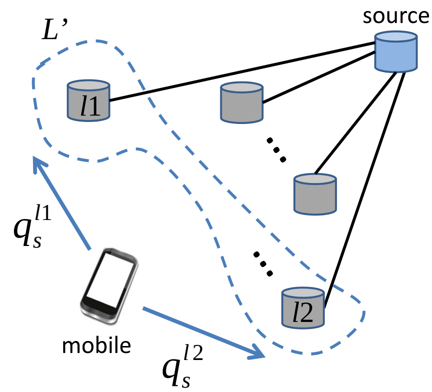


Figure 4.1: Flat cache point configuration. The proactive caching problem involves selecting, based on the mobile transition probabilities, the set L' of caches to proactively cache IO s, in order to achieve the optimisation target of (4.1) while satisfying the constraint of (4.2). In the proposed scheme the decision to proactively prefetch an IO is taken independently for each cache.

where

$$\mathcal{T}_s = \sum_{l \in L} \mathcal{T}_s^l$$

stands for the average delay cost for obtaining IO s and b_s^l equals one if the IO s is proactively fetched in cache l and zero if it is not proactively fetched in cache l. $\mathcal{T}_s^l$ is equal to $q_s^l T_R$ if the IO is not in cache l ($b_s^l = 0$) and needs to be obtained from its original remote location and $q_s^l T_L$ if the IO is stored in cache l ($b_s^l = 1$). The above optimization problem involves selecting for each IO s requested by a mobile, based on the mobile transition probabilities, Figure 4.1, the subset $L' \in L$ of caches that will proactively fetch IO s so that it is immediately available to the mobile when it connects to an AP close to the cache, in order to achieve the optimization target (4.1) while satisfying the cache storage constraint (4.2).

In order to efficiently utilize the cache storage, we introduce a congestion price p_l which is adapted based on the demand for caching and the available storage. Specifically,

$$p_l(t+1) = \left[p_l(t) + \gamma\left(o \cdot b^l(t) - B_l\right)\right]^+ , \tag{4.3}$$

where $b^l(t)$ is the aggregate demand at cache l at time t and γ is the price update factor, which determines how quickly the cache congestion price adapts to changes of the demand for caching.

The decision to proactively fetch an IO s at cache l is based on the following rule:

$$b_s^l = \begin{cases} 0 & \text{if} \quad q_s^l(T_R - T_L) < p_l \\ \\ 1 & \text{if} \quad q_s^l(T_R - T_L) \geq p_l \end{cases} \tag{4.4}$$

The above decision rule provides a *distributed* and *decentralized* approach to decide autonomously in each cache l whether to cache the IO s or not. The decision essentially trades the cost of transferring s to a mobile in case it handovers to l with the cost of occupying space in the cache point for s instead for another IO. Note at this point, that trading transfer and cache costs, which are different in nature, presupposes a common measurement *value* (e.g., a monetary unit) which expresses how much it costs to consume cache space for s instead of another IO, in exchange for a reduced delay cost for obtaining s.

Adjusting the cache price using (4.3) directs the system towards efficient use of cache storage, while achieving the optimization target (4.1). Specifically, when the cache is underutilized, i.e. cache storage is available, the congestion price decreases, thus allowing more IOs to be proactively fetched in the cache based on the decision rule (4.4). On the contrary, when the cache demand is larger than the cache size, then the price increases which in turn, due to (4.4), reduces the number of IOs that are requested to be cached. Furthermore, when the cache price is such that the amount of requested cache is equal to the cache storage, then due to the decision rule (4.4) we are certain that these requests correspond to the highest values of $q_s^l(T_R - T_L)$, hence the minimum in (4.1) is achieved. Note that content popularity can be incorporated in the above model if we replace q_s^l with the sum $q_s^l + r_s$, where r_s is the popularity of IO s.

Two practical issues related to the application of the decision procedure (4.4) include where the decision is taken and if the decision is to proactively cache an IO, when should prefetching start. Regarding where the decision is taken, one option is for the mobile requesting s, or some proxy on behalf of the mobile, to inform all the cache points collocated with or close to its possible future APs about its transition probability: When cache l learns the probability q_s^l, then together with the delays T_L, T_R and the cache congestion price p_l it can apply the decision rule (4.4). Alternatively, the caching decision can be taken at the mobile, or its proxy, in which case it would need to learn the delays T_L, T_R and the cache price p_l from all caches it has some probability to connect to. The second issue of when to start to prefetch an IO is related to the time interval after which the mobile connects to its next network AP and the time for the cache to download the requested IO from its remote location.

When a mobile moves to its new AP, then it can directly receive the requested IO from the local cache, if the cache had prefetched the requested IO. Otherwise, the mobile obtains the requested IO from the original source. When a mobile connects to its new AP, then the space occupied in all caches that proactively cached that mobile's data can be freed; of course, if the local cache at the mobile's new AP had prefetched the mobile's data, then the corresponding space will be freed after the data is transferred to the mobile. The above actions require communication and cooperation among caches.

The model presented above can be extended to IOs with different sizes, by replacing the constraint in (4.2) with

$$\sum_{s \in S_l} o_s \cdot b_s^l \leq B_l \,,$$

where o_s is the size of IO s. Additionally, the cache price p_l on the right side of the inequalities in (4.4) should be replaced with $o_s \cdot p_l$. For IOs with different sizes, the optimization problem becomes identical to the *0/1 Knapsack Problem* [170], which implies that it can be solved with Dynamic Programming as in [171]. However, even if a solution is found, it may take too long to calculate as the 0/1 Knapsack Problem falls within the class of NP-hard.

Another extension is when the remote and local delay is different for different IOs and caches, hence they can be denoted ${T_R}^{s,l}$, $T_L^{s,l}$ and the decision rule (4.4) can be adapted accordingly. Instead of maintaining different delays for different IOs or mobiles, delays can be associated with IO or mobile types, or can depend on the mobile's initial network AP. In all these cases, the actual values of the delay can be estimated in a measurement-based manner.

Optimal for equal-size IOs

For a given set of cache requests, the optimal in the case of equal size IOs, can be obtained for a flat set of caches as follows: For each cache l, we order the cache requests in decreasing value of $q_s^l(T_R - T_L)$. Then, starting from the request with the highest $q_s^l(T_R - T_L)$, we fill the cache until the constraint B_l is reached.

The above procedure for obtaining the optimal is performed in rounds: in the beginning of each round we have a given set of cache requests. This is unlike the solution based on cache congestion pricing, where the decision of whether to cache an IO is taken iteratively, based on (4.4), for each cache request, hence can be applied on-line. A practical issue with the optimal solution is the duration of each round, which determines the number of cache requests considered; this duration depends on the time interval after which a mobile connects to its new AP and T_R.

4.3.2 Proactive Caching in a two-level hierarchy

EPC can be also applied in a *hierarchical* cache space where *leaf* caches are under only one *mid-level* cache and the cost for fetching data from the mid-level parent T_M satisfies:

$$T_L < T_M < T_R.$$

An IO can be proactively fetched to a leaf cache, a mid-level cache, or even to both. For practical reasons and *not* because of any model limitations, our analysis is focused specifically on the case of a *two-level* cache hierarchy (Figure 4.2 on the following page), because this configuration reflects the current network reality where a leaf cache point can correspond to a WLAN such as a small cell or hotspot, or at a wired home/office LAN, which perform local/edge caching, and mid-level caches can correspond to CDN servers placed inside the ISP that connects the leaf to the rest of the Internet. *Generalising* to arbitrary hierarchies *is possible* via a recursive application of the decision procedure which we discuss next, yet this comes at a cost of added complexity with limited –if any– practical relevance.

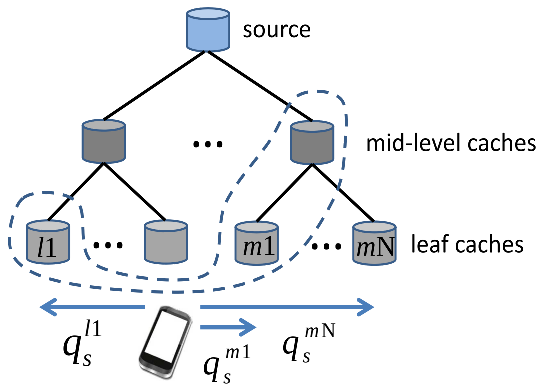


Figure 4.2: Two-level cache hierarchy. Each mid-level cache cooperates with its leaf caches to decide which will proactively cache an IO. No such cooperation is necessary between mid-level caches.

The addition of a mid-level cache point one level above the leaves forms turns the problem into a *Generalized Assignment Problem* [170], which is a variation of the *0/1 Multiple Knapsack problem* involving multiple knapsacks, each with a possibly different maximum weight limit. At any time instance, there is a given set of cache requests from the mobiles that are active at that time instant. For each such time instance, the proactive cache problem for a two-level cache hierarchy has similarities with the Data Placement Problem [146, 145], where the probability of an IO being requested at a specific cache is given by the probability of the mobile moving to the corresponding network AP. The authors of [145] show that the Data Placement Problem with different data sizes is *NP-complete*. Although there are cases where the placement problem in a hierarchical network with equal size IOs can be solved in polynomial time as shown in [172], such solutions have a high polynomial degree and apply to an offline version of the problem.

Solution procedure

Our approach to solve the proactive caching problem in a two-level cache hierarchy first considers two flat cache selection problems: one assuming that the IO is proactively fetched in the mid-level cache and the other assuming that the IO is not proactively fetched in the mid-level cache. Each of the aforementioned flat cache problems can be solved using the distributed approach presented in Section 4.3.1, by having the mid-level cache send the leaf caches the delay T_R, which is the delay for obtaining the IO from the remote source, and the delay T_M, which is the delay for obtaining the IO from the mid-level cache. Each leaf cache decides whether to cache or not the specific IO by using formula (4.4) for each of the two problems:

1. For the problem where the IO is assumed *not* to be cached in the mid-level cache point, the leaf uses formula (4.4) to decide if the IO should be prefetched in the leaf cache point.

2. For the problem where the IO is assumed to be cached in the mid-level cache point, the leaf cache uses (4.4) again, only replacing T_R with T_M to decide if the IO should be prefetched to the leaf cache point.

Next, each leaf informs its parent mid-level cache point about the expected IO delay for each of the two problems, i.e:

$$q_s^l \cdot T_R \quad \text{or} \quad q_s^l \cdot T_M \tag{4.5}$$

if the decision is *not* to proactively fetch the IO and

$$q_s^l \cdot T_L \tag{4.6}$$

if the decision is to proactively fetch the IO to the leaf cache point.

After receiving from all the leaf caches the delay for the two problems, the mid-level takes the *summation*[22] of the expected delays for each problem:

- $\mathcal{T}_M^{mid}$, in the case the mid-level cache proactively fetches s and
- $\mathcal{T}_R^{mid}$, in the case the mid-level cache does not proactively fetch s.

The decision of whether to cache or not cache an IO in the mid-level cache is determined based on a decision rule that resembles formula (4.4):

$$b_s^{mid} = \begin{cases} 0 & \text{if} \quad \mathcal{T}_R^{mid} - \mathcal{T}_M^{mid} < p_{mid} \\ \\ 1 & \text{if} \quad \mathcal{T}_R^{mid} - \mathcal{T}_M^{mid} \geq p_{mid} \end{cases} \tag{4.7}$$

That is to say that if the expected delay gain from caching the IO in the mid-level $\mathcal{T}_R^{mid} - \mathcal{T}_M^{mid} \geq p_{mid}$ is greater than implied cost (i.e., the congestion price p_{mid}) for occupying buffer space in the mid-level cache point for IO s, then s is proactively fetched to the mid-level cache; otherwise it is not. Note that the congestion price for the mid-level cache p_{mid} is updated in a similar manner as the congestion price for the leaf cache points along the lines of formula (4.3), but based on the demand for caching in the mid-level cache and the storage available in the mid-level cache. Following its decision, the mid-level cache point informs the leaf caches which delay factor (T_R or T_M) they should use in (4.4) to eventually decide whether to cache an IO.

The above procedure is distributed and requires some cooperation between the mid-level cache and its leaf caches. Moreover, it can be applied to a hierarchy with more than one mid-level caches, as long as each leaf cache is a child of only one mid-level cache. The proactive caching decision for each mid-level cache and its leaf caches follows the above approach, and can be performed independently of other mid-level caches.

4.4 Transfer Delay as a function of data size

In Sections 4.3.1 and 4.3.2 we assumed that the transfer delays T_L, T_M, T_R are independent of the IO size. In this section we consider the case where the cost of delay

[22] Using the summation of the expected costs for transferring a certain IO s, makes the decision taken by the mid level identical to EMC decisions after rule (5.1) on page 82.

depends on the IO size. As we will see, in this case there can be still gains if only a part of an IO is proactively cached. Note that apart from delay, our consequent analysis is applicable to any type of cost that can be defined as a function of requested IO's size, as in the case of monetary charges defined after the volume of consumed data, which we discuss later in Section 5.3 on page 88.

Let R_L be the rate for transferring to a mobile data stored locally at a cache close to the mobile's AP, and R_R be the rate for transferring to a mobile data from a remote source. We assume that $R_R < R_L$, which justifies why proactively caching can be beneficial. As an example, R_L can be the rate for transferring data across a Wi-Fi or cellular interface, while R_R can be the rate for transferring data across a hotspot or small cell's backhaul link, which is typically smaller than the rate of a Wi-Fi or cellular interface.

Assume that some part x_s^l of IO s that has size o_s is proactively fetched to cache l, whereas the remaining part $o_s - x_s^l$ would need to be obtained from the IO's original remote location. If the mobile requesting IO s moves to an AP close to cache l, then the delay for transferring the whole IO s is given by

$$\mathcal{T}_s(x_s^l) = \frac{x_s^l}{R_L} + \frac{o_s - x_s^l}{R_R} = \frac{o_s}{R_R} - \left(\frac{1}{R_R} - \frac{1}{R_L}\right) x_s^l \, . \tag{4.8}$$

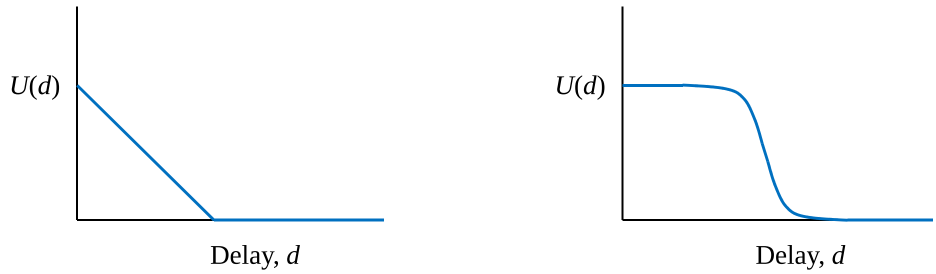
In the last equation we have assumed that the rates R_L, R_R are the same for all caches l. The results presented below are similar when the rates are different for different caches.

Consider a utility function $\mathcal{U}_s(d)$ that represents a mobile user's valuation for delay d in transferring IO s. $\mathcal{U}_s(d)$ is a decreasing function, and possible shapes for it are shown in Figure 4.3. Specifically, Figure 4.3(a) corresponds to the case where a user obtains no value when the delay is above some maximum threshold, and obtains a value that increases linearly as the delay approaches zero. The sigmoid utility in Figure 4.3(b) corresponds to the case where a user obtains maximum value when the delay is below some minimum threshold, while he obtains zero value when the delay is above some maximum threshold; such a curve approximates the exact step utility of hard real-time applications, with strict delay requirements.

We can define the utility:

$$U_s^l(x_s^l) = \mathcal{U}_s(\mathcal{T}_s(x_s^l)/q_s^l) \, , \tag{4.9}$$

which is a function of the part x_s^l of the IO s that is proactively fetched in cache point l. Note that the factor $1/q_s^l$ accounts for the transition probability to cache point l, hence if a mobile has a *higher probability* to move to cache l compared to some other cache, then it would need to proactively cache a *larger* amount of data for s to achieve the same utility. We assume that the utility function $U_s^l(x_s^l)$ is *continuous* and *strictly increasing* in the interval $[m_s^l, M_s^l]$, where $m_s^l \geq 0$ and $m_s^l < M_s^l \leq o_s$ are minimum and maximum values of x_s^l for which the following hold: $U_s^l(x_s^l) = U_s^l(m_s^l)$ for $x_s^l \leq m_s^l$ and $U_s^l(x_s^l) = U_s^l(M_s^l)$ for $x_s^l \geq M_s^l$. As an example, a utility function that corresponds to



a: Linear utility. Value increases as delay approaches to zero.

b: Sigmoid utility. Max value below a minimum delay threshold.

Figure 4.3: Example utilities as a function of delay. Users obtain *no* value above a maximum delay threshold.

Figure 4.3(a) is:

$$U_s^l(x_s^l) = \frac{\mathcal{R}}{q_s^l}(x_s^l - m_s^l)\,, \tag{4.10}$$

for $x_s^l \in [m_s^l, M_s^l]$, where $\mathcal{R} = \frac{1}{R_R} - \frac{1}{R_L}$. As previously described in Sections 4.3.1 and 4.3.2, we define a cache congestion price that is updated in a similar manner as formula (4.3), only with $o \cdot b^l(t)$ being replaced by

$$x^l(t) = \sum_{s \in S_l} x_s^l(t)\,,$$

which gives the aggregate demand for cache storage at l, when IOs can be partially cached:

$$p_l(t+1) = \left[p_l(t) + \gamma\left(x^l(t) - B_l\right)\right]^+\,, \tag{4.11}$$

where as before γ is a price update factor.

The framework below follows the model of Wang et al. [124]. We define the following decision rule for selecting the amount x_s^l of s to be proactively fetched in cache l:

$$x_s^l = \begin{cases} m_s^l & \text{if } \frac{1}{U_s^l(m_s^l)} \leq p_l \\ U_s^{l^{-1}}\left(\frac{1}{p_l}\right) & \text{if } \frac{1}{U_s^l(M_s^l)} < p_l < \frac{1}{U_s^l(m_s^l)} \\ M_s^l & \text{if } p_l \leq \frac{1}{U_s^l(M_s^l)} \end{cases} \tag{4.12}$$

Unlike the binary decision rule (4.4), where an IO is either fully cached or is not cached, with (4.12) it may happen that only a part of IO s is fetched at cache l: x_s^l is a continuous variable with values in $[m_s^l, M_s^l]$, where $m_s^l \leq 0$ and $m_s^l < M_s^l \leq o_s$.

Using the results from [124], it can be shown that a system where the amount x_s^l of s to be proactively fetched in cache l is determined by formula (4.12) and the cache price is updated according to (4.11), with an appropriately small value γ, will converge and solve the following optimization problem:

$$
\begin{aligned}
\max_{m_s^l \le x_s^l \le M_s^l} \quad & \sum_{l\in L}\sum_{s\in S} V_s(x_s^l) \\
\text{subject to} \quad & \sum_{s\in S_l} x_s^l \le B_l\,, \quad l \in L
\end{aligned}
\tag{4.13}
$$

where S is the set of all IOs, S_l is the set of IOs cached at l, L is the set of all caches, and

$$
V_s(x_s^l) = \int_{m_s^l}^{x_s^l} \frac{1}{U_s^l(y)} dy\,, \quad m_s^l \le x_s^l \le M_s^l\,.
$$

As an example, for the utility $U_s^l(x_s^l) = \frac{\mathcal{R}}{q_s^l}(x_s^l - m_s^l)$, we have $V_s(x_s^l) = \frac{q_s^l}{\mathcal{R}} \log(x_s^l - m_s^l)$.

Moreover, at the optimum the requested amount of data to be proactively cached $\{x_s^{l\,*} : s \in S, l \in L\}$ achieves utility proportional fairness: for any other set of cached data sizes $\{x_s^l : m_s^l \le x_s^l \le M_s^l, s \in S, l \in L\}$,

$$
\sum_{l\in L}\sum_{s\in S} \frac{x_s^l - x_s^{l\,*}}{U_s^l(x_s^{l\,*})} \le 0\,.
$$

Utility fairness can be seen as a way to allocate limited resources in a manner that is fair from an application perspective, since it takes into account the actual valuation (utility) for a specific amount of resources (cache storage in our case). In contrast, resource-oriented fairness definitions, such as proportional fairness and max-min fairness, seek to allocate resources in a fair manner from a resource-centric perspective, which does not necessarily reflect the requirements at the application level. Thus, considering utility fairness in the model of this section results in an application-oriented approach for performing proactive caching.

4.5 Implementation over Information-Centric Networking architectures[b]

EPC can be appropriately integrated with the network layer of ICNs. ICN architectures implement a receiver-driven (or pull-based) model. For instance, NDN (Section 2.3.2 on page 27) implements a pull-based model as the basic (low-level) communication primitive, or PSI (Section 2.3.1 on page 24) implements a receiver-driven model at the rendezvous (or resolution) layer, whereas at the separate forwarding layer it supports both a receiver-driven and sender-driven model. Hence, in both of these as well as other ICN architectures a mobile's information requests can be easily extracted and used to proactively cache the IOs which match their requests.

Regarding the transfer costs and the mobile transition probabilities, they can be estimated locally at each neighbouring cache point l. The only information that needs

[b] The discussion in the current section applies also the case of the models presented in Chapter 5.

to be communicated is the total number of transitions from the old (source) AP i to any of its neighbouring AP, which is necessary for estimating the transition probabilities. This communication involves a one-to-many (multicast) dissemination of the same information, matching the underlying Pub-Sub communication model in ICN. Similarly, the transmission of a mobile's requests and notifications to start or stop proactive caching at the neighbouring caching points also involves a one-to-many dissemination of the same information. If cache decisions are taken at the neighbouring cache points, then the aforementioned one-to-many communications can be performed in a receiver-driven (pull-based) fashion. Hence, they can be appropriately implemented over NDN's pull-based communication primitive, exploiting NDN's ability to effectively disseminate that same information to multiple interested receivers. Similarly, it can be implemented in the PSI architecture. Moreover, one-to-many dissemination can exploit both NDN and PSI's native multicast capabilities. An additional flexibility of PSI compared to NDN is that the former can support both receiver-driven but also sender-driven information transport; this is possible through the separation of the rendezvous (or resolution) function which matches publication announcements with subscriptions, and the transport/forwarding of information items from the publishers to the subscribers; this could be an advantage if the decision to proactively cache information was taken at the original proxy, rather than at the neighbouring proxies.

Finally, the last communication mentioned above involves the mobile's new cache point sending a notification to its old one announcing the mobile's new AP. This can be implemented in both PSI and NDN by having the old caching point issue a subscription message (in PSI) or an Interest (in NDN) that corresponds to an IO containing the mobile's connection status. Once the mobile connects to a new AP, the latter issues a publication announcement (in PSI) or a Data packet (in NDN) that matches the aforementioned subscription or Interest, respectively. Note that this communication exploits the ability to asynchronously issue subscriptions or Interests prior to publication announcements or Data packets in the PSI or NDN architecture, respectively. Moreover, the above *any-to-one* communication is compatible with NDN's single Data per Interest requirement, since after disconnection a mobile will connect to *one* proxy that will publish the *single* IO that satisfies the subscription or Interest.

4.6 Evaluation

In this section we evaluate the proposed Efficient Proactive Caching (EPC) scheme using the OMNeT++ simulation framework. We present results for a flat and a two-level cache hierarchy, for IOs that have the same size and when the delay is independent of the IO size. Evaluation results for different IO sizes and when the delay is a function of the IO size will be included in a followup of this paper.

Table 4.1: Simulation model parameter values. The values designated with * are default values, i.e. the values if the specific evaluation scenario does not indicate otherwise.

Parameter	Values	
	Fixed delay	Scaled-down Internet topology
# of active mobiles	160	160 per neighbourhood 10 neighbourhoods
# of APs	8	8 per neighbourhood
Avg. mobile transition probs	SKD50%: $50\%, 20\%, 10\%, 7.5\%, 5\%, 3 \times 2.5\%$ SKD70%*: $70\%, 2 \times 10\%, 3 \times 2.5\%, 2 \times 1.25\%$ SKD90%: $90\%, 3 \times 2\%, 4 \times 1\%$	
Stdev. of transition probs	$5\%^{*}, 30\%$	
Delay	$T_M/T_L = 2, 5^{*}$ $T_R/T_L = 10^{*}, 18$	$T_M/T_L = 5$ $T_R/T_L = \frac{9}{5}$(#hops-1)+1
Total cache (leaf+mid)	$0 - 320$ IOs, default:240^{*}	

4.6.1 Simulation model

We consider scenarios where the delay for obtaining an IO is the same for all mobiles (referred to as fixed delay scenarios) and scenarios where the delay for obtaining an IO from the source is variable and depends on the number of hops between the source and the mobile (receiver); the latter scenario involves a scaled-down Internet topology containing 400 nodes, with each node representing an AS [173]. The value of various system parameters considered in the simulations are shown in Table 4.1. In the fixed delay scenarios, there are a total of 160 mobile users that issue requests for IOs of the same size. The mobiles can move to 8 different network APs, where there is a corresponding local leaf cache. At a higher level, there is a mid-level cache. Whenever a mobile performs a handover, we assume that a new mobile enters the system, thus the total number of active mobiles in the system always remains 160. In the scenarios with scaled-down Internet topology, we conduct simulations over a set of 10 different neighbourhoods, with each neighbourhood having 160 mobile users, yielding a total of 1600 mobile users. Each neighbourhood is composed by 8 ASs lying on the edge of the topology. The vast majority of such ASs are stub (access) networks to the Internet. For each neighbourhood, we randomly select an initial stub AS and then form the neighbourhood by selecting 8 stub ASs based on minimum hop distance from the initial stub AS. The underlying idea is that the selected nodes are neighbours of the initial stub AS, thus typically they should be a few hops away for mobile users hosted by the initial stub AS. Note that the neighbours are selected so there are no overlaps between different neighbourhoods.

A mobile user can move with some transition probability to one of the 8 different APs, each with its own local cache. In the fixed delay scenarios there is one mid-level cache, whereas in the scaled-down Internet topology scenarios there is one mid-level cache in each neighbourhood. We consider 4 different sets of mobile transition probabilities, Table 4.1, with different *skewness*, where a higher *skewness* corresponds to a higher probability to move to a particular AP (equivalently, cache). We also assume that the destination network AP (equivalently, the destination cache) with the highest transition probability is different for different mobiles, such that the number of active mobiles moving to a specific cache is on average the same, and equal to 20 throughout the simulation. Finally, note that with the EPC and OPTIMAL models the transition probabilities used in the caching decisions are measured as the simulation progresses.

The performance of the EPC model depends only on the ratio of delays T_R/T_L and T_M/T_L, because the decision rule for both the leaf cache (4.4) and the mid-level cache has a linear dependence on the delays and the congestion price is adjusted to achieve high utilization; hence we consider the delay ratios rather than the absolute delay values. Assuming that the leaf cache performs caching at a LAN and that the mid-level cache performs caching at the ISP that connects the local network to the Internet, we have taken $T_M/T_L = 2$ and 5, which can be seen as the number of hops or the actual delay for obtaining an IO from an ISP cache relative to a local network cache. We have considered $T_R/T_L = 10$ and 18; together with the values of T_M/T_L, these give values of T_R/T_M in the range $[2, 5]$; the lowest value $T_R/T_M = 2$ corresponds to the case where an ISP has a direct peering link with the content provider network (source for an IO), in which case their distance is two AS hops. At the other side, studies have shown that the average inter-AS path length has remained practically constant and equal to 4.2 over the last 12 years [174], and for this reason we have selected the highest value of T_R/T_M to be 5. For the scaled-down Internet topology scenarios, we consider $T_M/T_L = 5$ and $T_R/T_L = \frac{9}{5}(\# \text{ hops} - 1) + 1$; the latter gives an average T_R/T_L equal to 10, since the average number of hops between a source and receiver in the scaled-down Internet topology is 6.

The performance of the proposed EPC model is compared to the OPTIMAL scheme, to a NAÏVE scheme, and an ORACLE[23]. It is important to note that the OPTIMAL scheme in the case of a flat cache structure is implemented such that the cache allocation is performed whenever cache storage is freed. In addition to being time consuming, the ability to implement frequent cache allocations can be constrained by the time for actually transferring the IOs to the caches where they are proactively fetched.

With the NAÏVE scheme, a mobile requests caching for all IOs, provided that storage is available. With the oracle, for each new cache request we assume that the IO is prefetched (provided there is cache space) by the cache located at the AP where the mobile will eventually connect to (hence the name oracle). Unlike the OPTIMAL scheme, which allocates cache storage in rounds considering the requests from active mobiles, the EPC, NAÏVE and ORACLE iteratively, for each new cache request, take caching decisions that do not change until the corresponding handover is performed.

4.6.2 Evaluation results

In this section we compare the various schemes in terms of the gains in reducing the average delay for the mobiles to obtain the requested IOs, compared to the delay if

[23] See the Appendix A.1 on page 148 for the definitions of NAÏVE, OPTIMAL and ORACLE.

no caching is used, i.e. when all IOs are obtained from the original sources. Because the performance of the Naïve and Oracle schemes showed very small dependence on the mobile transition probabilities, for these schemes we do not show results for different transition probabilities.

The results shown are the average of 10 runs for each scenario in the fixed delay case and 10 runs for each neighbourhood in the case of the scaled-down Internet topology. Each run has a duration that corresponds to 10.000 handoffs. For these parameters, the 95% confidence interval was within 5% of the average values, hence they are not shown in the graphs. The conclusions from the evaluation are the following:

- The gains of EPC are higher when there is more mobility information, i.e. for a higher skewness of the mobile transition probabilities; these gains are close to those of the Optimal scheme for a flat cache structure and the oracle.

- For a higher skewness of the mobile transition probabilities, EPC achieves higher gains when more storage is allocated to leaf caches. Moreover, the gains of EPC are significantly higher than the Naïve scheme when more storage is allocated to the leaf caches.

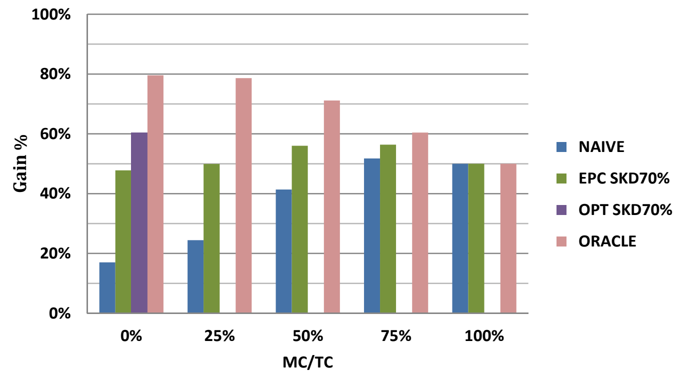
- Even for a relatively high variation of the mobile transition probabilities, EPC's gains are robust, and significantly higher than the Naïve scheme, when the skewness of transition probabilities is high and when more storage is allocated to leaf caches.

Comparison of EPC with the Naïve, Optimal, and Oracle schemes

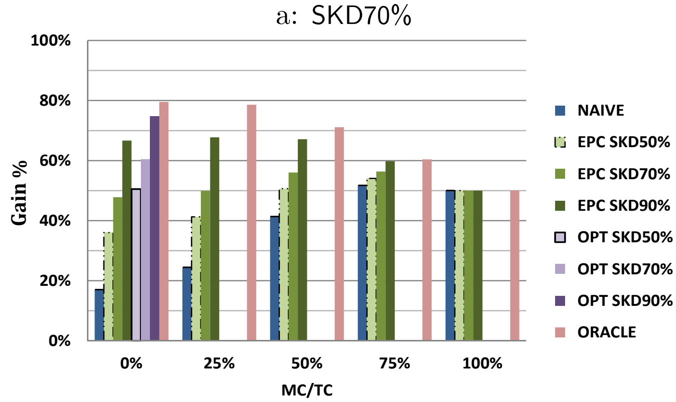
Figure 4.4a considers a fixed amount of total cache storage TC. The x-axis shows the percentage MC/TC of the total cache storage that is allocated to the mid-level cache; hence, 0% indicates that all cache storage is equally distributed to the leaf caches, whereas 100% indicates that all cache storage is allocated to the mid-level cache. Figure 4.4a shows that EPC achieves a higher gain than the Naïve scheme for small values of MC/TC. In further, EPC's gain is close to the Optimal scheme, in the case of a flat cache structure ($MC/TC = 0$). On the other hand, for larger values of MC/TC the gains of all schemes are close, and become equal when all storage is allocated to the mid-level cache; this occurs because the capacity of the mid-level cache is larger than the total demand and all requested IOs can be cached, thus the gains for all schemes are equal and determined by the delay of the mid-level cache. Figure 4.4a also shows that the Oracle achieves the best performance when all the storage is allocated to leaf caches, whereas the best performance of the EPC and Naïve schemes for the specific system parameters is achieved when $MC/TC = 75\%$.

Influence of mobile transition probabilities

Figure 4.4b shows that when the *skewness* of the mobile transition probabilities increases, then the gains of EPC are higher when more storage is allocated to the leaf caches. Moreover, when the skewness is large (i.e. SKD90%), then EPC achieves more than 80% of the gains achieved by the ORACLE and almost 90% of the gains achieved by the OPTIMAL scheme for a flat cache structure. It is important to note that, for large skewness, the EPC model achieves a higher gain than the NAÏVE scheme even when the allocation of storage to leaf and mid-caches is selected *to achieve the best performance for each scheme:* Specifically, for $MC/TC = 25\%$ EPC achieves gain 68%, which is more than 30% higher than the highest gain achieved by the NAÏVE scheme, namely 52% for $MC/TC = 75\%$.



a: SKD70%

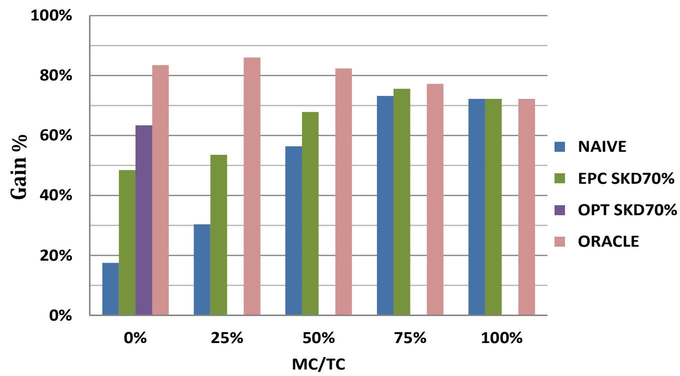


b: SKD50%, SKD70% and SKD90%

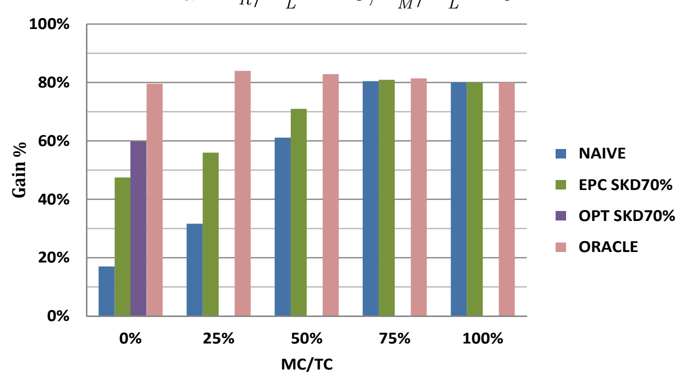
Figure 4.4: Influence of mobile transition probabilities on gain. Fixed-delay, $T_R/T_L = 10, T_M/T_L = 5,\ TC = 240$ (total cache).

Influence of delay ratios

Comparing Figure 4.5a to Figure 4.4a on this page shows that when the delay for obtaining IOs from their original sources is higher, then the performance of all schemes is generally higher, especially when more storage is allocated to the mid-level cache (larger values of MC/TC). Figure 4.5b shows that when the delay for obtaining data from a mid-level cache is close to the delay for obtaining the data from a leaf cache ($T_M/T_L = 2$), then EPC achieves the highest gains when all storage is allocated to the mid-level cache.



a: $T_R/T_L = 18, T_M/T_L = 5$

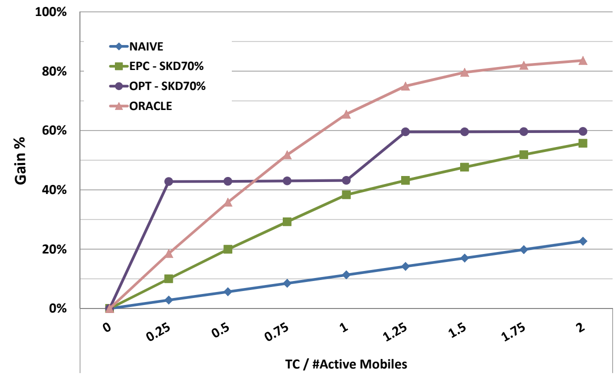


b: $T_R/T_L = 10, T_M/T_L = 2$

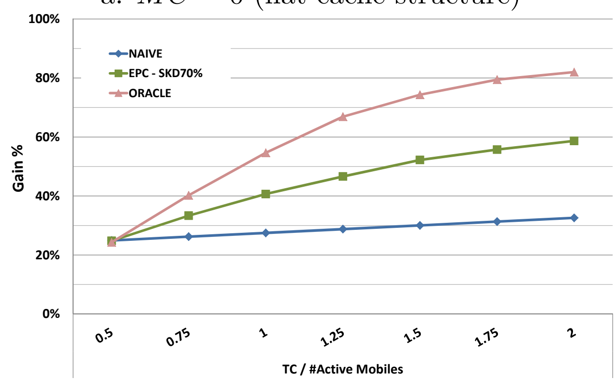
Figure 4.5: Influence of delay ratios on gain. Fixed delay, SKD%70%, $TC = 240$.

Influence of total cache storage

Next we consider a scenario with the scaled-down Internet topology. Figure 4.6a shows the influence of the total cache storage on the performance in the case of a flat cache structure, while Figure 4.6b shows the same influence when the mid-level cache of each neighbourhood can store 80 IOs. Observe that while the gain of the Naïve scheme increases linearly with the total cache, the gains for the EPC and Oracle increase slower for larger values of the total cache. Furthermore, the Optimal scheme exhibits a stepwise behaviour, which is a result of its operation: the Optimal scheme recomputes the cache allocation whenever cache storage is freed; this is also the reason that the Optimal scheme has, for small values of cache storage, a higher gain than oracle: the Oracle scheme decides where an IO is cached when the corresponding request appears; this decision does not change until the corresponding handover is performed. However, it may happen that when the decision was made there was no available storage at the cache where the mobile will eventually move to. On the other hand, the Optimal scheme



a: $MC = 0$ (flat cache structure)



b: $MC = 80$ (mid-level cache)

Figure 4.6: Influence of total cache size on gain. Scaled-down Internet topology.

continuously recomputes the cache allocation when storage space becomes available, hence can take advantage of any free cache space right before a handover is performed.

Transient behaviour

Figure 4.7 shows that when the system starts with no knowledge of the mobile transition probabilities, EPC's gain converges to its steady state value slower than the other schemes; this is due to the dynamic updating of cache congestion prices and the measurement-based estimation of mobile transition probabilities. As expected, the convergence time is larger for smaller values of factor γ: after a change of the mobile transition probabilities from SKD%70% to SKD%90% at time 5000, the gain reaches 95% of its steady state value after approximately 750 handovers for $\gamma = 0.5$ and 490 handovers for $\gamma = 8$.

Influence of variation of mobile transition probabilities

Comparison of Figures 4.8 and 4.4a shows that EPC's gains are robust to the variation of the mobile transition probabilities: for values of MC that EPC achieves the highest gains (50% and 75%), an increase of the standard deviation of the transition

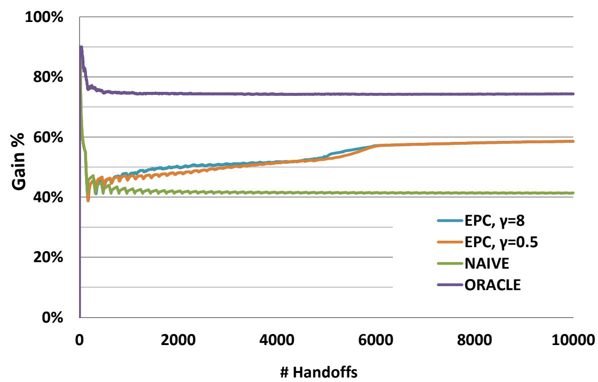


Figure 4.7: Transient behaviour. The EPC model converges slower for smaller values of the price update factor γ. At point 5000 the mobile transition probabilities change from SKD70% to SKD90%. Fixed delay.

probabilities from 5% (Figure 4.4a) to 30% (Figure 4.8) reduces the gains of EPC by less than 10%. Furthermore, the higher variation of the transition probabilities appears to influence the ORACLE more than the other schemes. This is because the higher variation can lead the ORACLE to require more leaf-level caching than what is available, hence is forced to use the mid-level cache which achieves lower delay gains; this is verified by the higher use of the mid-level buffer by the oracle, for a higher variation of the transition probabilities.

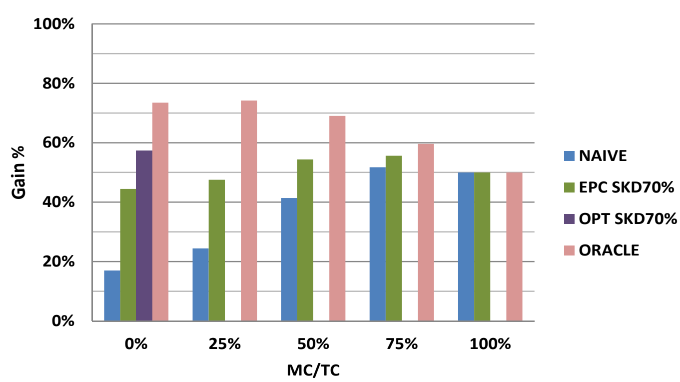


Figure 4.8: Influence of variation of mobile transition probabilities. Fixed delay, standard deviation of mobile transition probabilities 30%.

Comparison of fixed delay with scaled-down Internet topology

A comparison of Figure 4.9 on the facing page, which is for the scaled-down Internet topology, to Figure 4.4a, which is for fixed delay, shows that the variation due to the Internet topology has a larger influence on the ORACLE scheme; similar to the above explanation for the higher variation of the mobile transition probabilities, this is because the higher variation results in the ORACLE scheme requiring leaf-level caching more than the available storage, thus forcing it to use the mid-level cache for which the delay gains are lower.

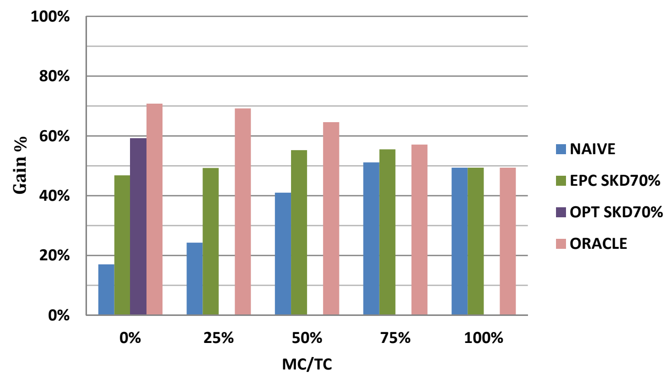


Figure 4.9: Scaled-down Internet topology, $T_R/T_L = 10, T_M/T_L = 5,\ TC = 240$ (total cache).

4.7 Conclusions

In this chapter, we presented and evaluated an Efficient Proactive Caching (EPC) scheme for reducing the delay in edge network mobility scenarios, which exploits mobility information and uses congestion pricing to efficiently utilize cache storage and address unique user demand. Our modelling framework includes the case of a flat cache structure and a two-level cache hierarchy, and the case where the delay is independent of the size of the requested objects and where the delay is a function of the object sizes. Our evaluation results show how various parameters influence the delay gains of the proposed scheme, which achieves robust and good performance relative to a scheme which attempts to naively cache all data objects, an optimal scheme for a flat cache structure, and an oracle which knows a priori a mobile's future attachment point.

Chapter 5

Mobility-based Caching for addressing Niche Mobile Demand, with Content Popularity & Legacy Caching model extensions

This chapter is based on our work of [48, 49, 55, 97]. It discusses an efficient mobility-based proactive caching model for addressing *niche* mobile demand, along with popularity-based and legacy caching model extensions. The former corresponds to our thesis point ((2)) on page 11 and to contributions ((2)) and ((3)) on page 13. Moreover, it is related to the general background discussion in Section 1.1.3 on page 8 and the literature overview provided in Section 3.1 on page 34. The presented basic model approach, which we refer to as the Efficient Mobility-based Caching (EMC) model, has similarities to our formerly presented EPC model in Chapter 4. But unlike EPC, its design is focused on *niche* rather than on unique mobile demand, as according to notable studies in the literature, niche demand represents a *significant* 20-40% of Internet demand and follows a growing trend due to the increasing popularity of the personalised videos posted in social networks. Additionally, the decision model in EMC can be extended to adapt (i) cache replacements (EMC-Rmodel), (ii) content *popularity* information and (iii) to exploit legacy-cached IO data (EMPC-Rmodel), with benefits also for stationary users. Finally, we demonstrate how the former models can be applied in heterogeneous wireless networking scenarios with the purpose of reducing monetary and/or delay costs for both users and their providers, with possible positive implications on the user-perceived QoE.

5.1 Background and motivation

5.1.1 Soaring wireless Mobile Demand

The advent of smart mobile devices along with strides in wireless technologies that enable higher throughput have created the necessary but not sufficient conditions for wireless mobility to bloom. On the one hand, wireless networking may be experiencing a continuous and unprecedented growth in terms of devices and content demand as discussed in Section 1.1.3 on page 8; but on the other hand, this situation pushes wireless mobility support, and specifically *macro-cellular capacities*, close or even beyond their limit. Particularly the soaring *mobile video* traffic arises as the ultimate challenge for wireless providers: it is projected [4] that by the year 2020 it will grow x11 relative to only 2015 and that it will account for 75% of the total traffic on its own. This increase comes as a result of the combined widespread of wireless mobile devices with video play abilities and freely available online video content particularly in *social networks* (Section 5.1.4). The problem of depleting wireless capacity resources gets further aggravated by the aggressive buffering policies of most HTTP video streaming services that account for 20-25% of web traffic in all networks [29] and which cause 25-39% of unnecessary mobile traffic [32]. Especially the latter is responsible for a significant waste of wireless resources, as 60% of videos are watched for less than 20% of their duration due to user aborts [32].

The above reality is in direct contrast to the continuously increasing user requirements for seamlessly and ubiquitously enjoying good quality videos from their mobile devices and in exchange for a desired (as low as possible) price. Clearly, the situation calls for cost-efficient solutions which do not employ charging mechanisms to control demand peaks, as that would make user unhappy, and which can preserve or even improve the offered QoS levels with positive implications on QoE.

To this direction, several solutions in the literature try to *offload* [175] (Section 5.1.5) the more costly macro-cellular traffic to smaller wireless cells as an alternative to expanding their macro-cellular support. According to thorough studies such as in the work of [175, 176] and references therein, using *small cells*[24] and third-party *Wi-Fi hotspots* is highly efficient, especially when the deployment of added macro BSs is required to cover demand [176]. This is also verified in practice, as offloading has yielded significant gains in the recent past, with 45% of global mobile data traffic having been offloaded to Wi-Fi or other small cells in 2013 [177], without which mobile traffic would have grown by 98% rather than by 81%. On the down side though, installing and running new small cells implies considerable Capital Expenditures (CapEx) and Operational Expenditures (OpEx)[25], while exploiting the already established Wi-Fi hotspot infrastructure is mostly limited due to the typically low capacities of backhaul connections that create a *bottleneck*.

To address the problem of backhaul congestion for either the purposes of traffic

[24] See definition of “small cell” in the Appendix A.1 on page 149.

[25] CapEx includes business expenditure on assets such as upgrading or extending facilities (i.e., the capital), while OpEx covers the expenses for running a service or the depreciation of capital with time.

offloading, in particular, and/or for improving the offered QoS levels, in general, recent efforts in the literature (Section 3.1.2 on page 35) as well as in the industry [4, 178, 179], adapt *proactive* caching of *popular* IOs in small cells so as to directly serve mobiles from a low-access delay local cache upon their attachment to a small cell, and hence, to fully exploit the *high* wireless capabilities of small cells [180, 181, 182]. Additionally, reusing cached IOs avoids needless data transfer repeats that imply extra inter-domain data transit costs [74] for providers, which can also add up to user charges.

But similarly to the prevailing solution of CDNs (Section 5.1.3) that employ servers across the Internet topology, the former local proactive approaches are designed to focus on *popular* IOs, hence leaving a *significant* (Section 5.1.2) part of demand that refers to *niche* requests for less popular or personalised content falling in the "long tail" of typical content popularity distributions largely unaddressed. In additions, some of these solutions [132] employ perfect knowledge of popularity, which is admittedly [134] difficult to achieve with respect to *locality* in both *space* and *time* [183], or they employ a centralised approach and/or a rather static[26] adaptation of content popularity that are *not* conducive to capturing temporal and spatial locality.

5.1.2 Niche demand and its significance

Regarding *web content*, Figure 1 in [57] shows that popularity-based caching reaches a 60-70% practical *hit ratio upper limit* depending on the homogeneity of the client population. A similar conclusion is drawn from Figure 2 in [56] which shows that the top 1% of the documents accounts for about 20-35% of all requests seen by a cache proxy, and the top-most 10% of popular web documents may account for about 45-55% of all requests seen by a cache proxy; yet it is the 25-40% of the top-most popular documents that account for 70% of all requests seen by a web cache proxy, and 70-80% of documents to account for 90% of requests. In a more recent study in [116], the authors shows that a large portion of web traffic is *dynamically* generated and *personalized* data, contributing up to 30-40% of the total traffic. These findings lead to the conclusion that:

> *At least a significant 30% of less popular or personalised web requests that account for 30-40% of traffic, are not addressed based on popularity.*

Regarding *video demand*, the work of [59] and references therein, acknowledge the significance of niche demand and identify the opportunities of leveraging *latent* demand for niche videos that are currently not reached due to wrong categorisation or ranking. According to the same study, caching only 10% of the most popular videos accounts for 80% of views on YouTube, i.e. a 20% of videos accesses on YouTube corresponds to less popular content. However, this profound popularity skewness could be the result of a *distortion* imposed by the applied search or recommendation algorithms or due to wrong categorisation or ranking. It is debatable if this is rooted in the nature of the

[26] Even the work of [54] that does consider user mobility apart from popularity, *statically* splits storage between popularity- and mobility prediction-based caching.

videos' popularity or if it is the result of a an imposed popularity distortion by search and recommendation algorithms. This speculation is also backed by another study [61] on different centralised video services, which reveals that 10% of the top-most popular videos accounts for ∼60% of requests, i.e. 90% of the *least* popular videos accounts for *40% of all requests.* This means that requests can be spread more widely than what is reported [59] about YouTube. For all that, we can conclude that a *significant* part of video requests, which reportedly lies between 20% and 40%, is *not* addressed by popularity-based caching approaches, and that:

> *Even if 60-80% of video accesses in the Internet continuous to account for popular content in future, a significant 20-40% of the total video demand will still refer to niche content.*

The above quoted conclusion points about niche web and video requests largely overlap with each other, paralleled by more *recent* strong indications [29] of a quickly growing niche traffic. And although it is unclear if the former percentages about requests apply also in terms of traffic, we note that there is *no* indication whatsoever in the literature about popular IOs being subject to a different file size distribution than the ones in the long tail.

Finally, our discussion in Section 4.2.2 on page 49 on unique demand within the boundaries of an edge network is highly related to the current one, as unique represents an *extreme* form *niche demand.* According to our discussion there, which is based on the latest work in the literature over the last two years [167, 168]:

> *Extremely niche demand within an edge network accounts for 70-80% of total demand in terms of number of requests and approximately 50% of total demand in terms of data volume size. Regarding video content in particular, up to 80% of requests for less popular videos are further uniquely requested.*

5.1.3 CDN functionality *vs.* Niche Demand

Content Delivery Networks (CDNs) are large distributed systems deployed within ISP facilities across the Internet with the goal to serve content with high availability and performance. To achieve so, CDNs use extensive caching (or replication) of *popular* content in multiple network sites as first step, to bring content "*close*" to where it is consumed. This is reportedly [19] done with the increasing cooperation of ISP networks, in order to remove any performance bottlenecks such as transmission delay, jittering or latency due to low throughput and/or network hop distance from the origin content servers that reside in the premises of content providers. Thus, they offer an efficient content (re)distribution service to both ISPs and content providers by eliminating the implied costs (monetary, QoS-related, etc.) of redundant data traversal on the network via transit ISPs and by enhancing the user-perceived QoE. Then as a second step, CDNs *redirect* user requests via DNS redirection [160, 184] to an appropriate server which (most commonly) reduces the network distance between end-user consumers and cached/replicated content.

Even though CDNs have their own proprietary design, most of them adapt the following architecture of key-component servers: *a) delivery* servers which are essentially caches deployed as close to the edge as possible, running one or more delivery applications; *b) storage* nodes which feed delivery caches with data or can host pre-published content rather than serving it on demand from origin servers; *c) origin* servers, i.e. the network servers originally hosting the content within the content owners' premises or, less commonly, within ISPs' own facilities; and *d) control* nodes with the primary purpose of managing the components of the CDN. Based on this model, CDNs address the problem of *scalable* and *efficient* content *delivery* to end users.

As mentioned above, CDNs target the part of the total demand that is currently (or it is predicted to become) *popular* to users in a given network region. They exploit the observed truncated power-law popularity distributions of web and video demand, particularly for the videos served by large video hosting websites (video hosting websites), and use distributed algorithms [152] to bring popular IOs closer to prospective end-users. By hosting popular content within the ISPs' own facilities, CDNs also reduce inter-ISP data transit charges [74], avoid bandwidth bottlenecks along the data transfer path and large network propagation distances which relegate QoS and QoE. Their success in enhancing performance for video content distribution has been so high, to the point that many ISP, telco and carrier companies themselves have adapted the idea of running their own CDNs within own facilities. Compared to traditional CDNs, the latter have full control over the "last mile", which enables them to deliver content closer to end-users via caches deep inside their networks and, hence, to further minimise network distances, to deliver data faster and more reliably.

Nevertheless, CDNs can still fail to assist a considerable part of demand: First, because cache *effectiveness* for content in the "long tail" of popularity distributions increases *only logarithmically* with the size of the cache [14, 56], whereas, second, CDNs deploy a network of caches that are sufficiently big for hosting *only* the most popular videos. This leaves *niche* requests to "pass through" the CDN level, although they represent a significant portion of mobile demand (see Section 5.1.2 and Section 5.1.4). A third reason lies in evidence which indicate that content popularity is *not* adequate for intercepting post-CDN demand. According to empirical findings [185], the requests missed by the CDN caches are *unlikely* to hit Facebook's internal cache due to wrong cache actions as a result of a plethora of content of similar popularity in the "long tail" of power-law distributions.

To make things worse, our discussion on the next page about the distinctiveness of video demand on Facebook and, in general, in Social Networks, indicates that traditional popularity-based CDN functionality may be not always sufficient. This is (perhaps) why Facebook [186] has it own CDN(-like) functionality tailored for live broadcast feeds from popular sources user profiles. However, such an infrastructure is not designed to address live broadcasts from multiple, less popular individuals.

5.1.4 Video demand in Social Networks[c]

Social Networks (Social Networks) reflect social relations among users who share a common background such as news interests or real-life connections, thus forming a *personalized* online social network that is unique per user profile. Since their advent, they are having an increasing impact on ISP traffic and on wireless mobility: Cisco [4] verifies that the observed increase in wireless traffic volumes is part due to the increasing popularity of Social Networks and predominantly Facebook, in addition to the increased usage of technologically advanced smart phones and tablets with higher recording and playback resolution capabilities, and the traffic from large video streaming providers like YouTube or Netflix.

While initially focused mostly on textual and image content, currently Social Networks mark a new era of *personalised videos*. Before that, demand was driven by developments in large video hosting website, which led to solutions designed for increasing the offered QoS of such services. Especially Facebook is a pacesetter for mobile[27] video and other developments in Social Networks. Due to the advanced capabilities of modern mobile smart devices, one should expect that users will in future be sharing "selfie" and other user-generated videos, perhaps, as frequently as status and photo updates currently are. Moreover, the latest web and mobile versions of Facebook enable video *auto-play* while users scroll down their "timeline" pages. This has important implications on wireless mobility, for it implies an added traffic for videos that are *not* explicitly requested to be viewed by users. This can cause a waste of network resources as it is the case with aggressive buffering techniques [32] applied by services like YouTube.

But despite its importance, little is known about video demand in Social Networks. To our knowledge, the only work about video requests in Social Networks is presented in [188]. This work reveals a striking level of popularity skewness (merely 2% of videos represent 90% of all requests) and some special popularity dynamics with respect to the correlation between early and future views of popular videos compared to the case of video hosting websites. However, the work is based on a non-international Facebook-like Social Network located in China, thus is conclusions may be not universal or applicable to the case of Facebook and/or other large international Social Networks. Moreover, the work focuses only on the effect of the "word-of-mouth" content dissemination.

Therefore, to complete one's understanding on the level of impact of Social Networks on video demand, we must focus on the differences with respect to the demand models in the case of "traditional" video streaming services such as YouTube. The latter holds a leading role [29] among all video services and, consequently, it is a de facto case-study [32, 59, 189, 190] for Internet video demand. Videos are stored on its cloud servers and prospective viewers are offered with a keyword-based *search* mechanism, paralleled

[c] The discussion in this section is related to the ones on Social Networks and video hosting websites on page 8.

[27] According to ComScore's "U.S. Mobile App Report", Facebook's mobile app was the most popular among US users in 2014 [187].

by *suggestions* based on popularity trends[28,29], in order to *discover desired* videos. As identified in [59, 60] and references therein, the *filtering* mechanisms applied on search results and on recommended content[30] can be held responsible for the occurring truncated power-law distributions. These distributions are characterised by *a small fraction* of videos that stands out substantially for its *popularity* and has the lion's share of the produced traffic volumes. Given this convenient popularity distribution, CDN services and local caching strategies (Section 3.1.3 on page 36) are designed to bring the topmost popular IOs "closer" to its prospective viewers. Nonetheless, leveraging popularity skewness is less likely to be an appropriate approach for addressing users' personalised demand within the context of social networking, due to the following important *qualitative differences* from "traditional" video hosting websites:

ta recomendation na einai se allhpedrash me to caching

a) User-specific filtering: As previously mentioned, video hosting websites try to influence popularities [59] to manipulate demand. Unlike that, demand in Social Networks is mainly dictated by the users' personalized networks of peers, liked/followed accounts and preference settings. For instance, Facebook users do *not* directly search for content based on keywords. as in the case of video hosting websites. Their "time-line" of content suggestions is defined after their own settings used as input to a special algorithm which aspires to stop overwhelming amounts of news by featuring only desired content from within their social network instead. Twitter on the other combines personalised (filter) lists of followed accounts, with hashtag searching and news "trends" *outside* of the users' networks.

b) Individuality, transitiveness & transience: Another difference from video hosting websites lies on the fact that content popularity depends on the characteristics of the *individual* users' social networks. To provide an example, the chance that a published video becomes popular grows with the publisher's network size. Also, it is highly *transitive* compared to traditional video hosting sites, as content can gain increased visibility if republished by popular users. Nevertheless, popularity falls sharply after two days on Facebook because publications stop from showing in the default news feed [185].

c) Latent popularity: Finally, a significant part of content popularity is latent, particularly in the case of user-generated videos, as videos become either extremely popular (tens of millions of views), or end up being consumed by niche audiences [60]. In the case of Social Networks, latent user-personalised video popularity is difficult to capture: users may view content without necessarily interacting with it via commenting, republishing, liking, etc.Even click-based view-counting

[28] `http://youtube-trends.blogspot.com/`

[29] `https://www.socialbakers.com/statistics/youtube/channels/`

[30] Noteworthy, interesting recent work [191] in the literature involves localised interaction between caching decisions and user recommendations. Unlike traditional recommendation systems, such a strategy uses controllable *bounded* distortion of personalised content preferences, rather than "shaping" user content preferences, which is aimed to manipulate the consequent IO popularities.

provides a questionable measure of popularity, especially nowadays that users can "silently" watch or neglect auto-played videos without any detectable interaction. Furthermore, studies [32] show that users can easily abort playback, i.e they are *not* interested in the videos they click. And although there appears to be a strong correlation between video popularities and rankings in [59], the authors of the same study also state that requests on YouTube are so skewed that it is *debatable* whether this is because viewers primarily want to watch what others watched before, or because of a distortion caused by wrong ranking or categorisation of videos.

5.1.5 Benefits and issues of Offloading traffic to Small Cells

Due to their small range and low-power needs, small cells and Wi-Fi hotspots imply a series of advantages compared to traditional macro-cellular wireless mobility support. First, because they allow to densify coverage and, hence, to *reuse* the available *spectrum* more *efficiently*. This is even more critical in the case of the operator-controlled small cells which operate within licensed spectrum. Second, because the providers can *improve* the *QoE* of their mobile customers by offering a good signal coverage in home, office and even public spaces where macro-cellular coverage is poor and/or the offered level of QoS is below user application requirements, e.g. due to difficulties with respect to the landscape. A third reason regards *energy consumption*. Signal strength fades with distance square, thus wireless communication within smaller ranges implies less power needs for both BS and user devices. Note that power consumption for macro-cellular operation is a highly significant OpEx for providers and, likewise, device battery consumption is also very important to users running energy-hungry services and applications [192] such as streamed video playing. Last, Wi-Fi hotspots in particular, allow providers to exploit the rich economic and technological benefits of the IEEE 802.11 protocol family [193], as well as to *utilise* the part of the *non-licensed spectrum* that is used in hotspots in addition to the licensed spectrum. Wi-Fi hotspots offer very high wireless *capacities* that are not met by LTE, UMTS nor WiMax cells, spanning from 54 Mbps to even 866.7 Mbps [180, 181, 182].

On the down side though, installing provider-controlled small cells in places that lack or have poor backhaul support, e.g. on lamp posts or traffic lights, implies a significant CapEx for site acquisition and backhaul facility installation, as well as a considerable OpEx for running and maintaining new infrastructures and services that can meet with demand volumes. In further, there are scenarios in which the need for data offloading is merely *occasional* such as in sport or concert venues, festivals or camping sites, etc. In such cases it is uneconomic and impractical to establish high-capacity (optical or wireless) backhaul links.

Many offloading solutions have recognised this fact and have turned to exploiting the existing backhaul infrastructure, albeit typical residential Asymmetric Digital Subscriber Line (ADSL) and even Very-high-bitrate Digital Subscriber Line (VDSL) links can pose a *bottleneck* for certain throughput-demanding traffic types like streamed video. As reported in [29], the capacity needs per streamed YouTube video are at least 2 Mbps to allow users to enjoy the service, which implies that supporting multiple mobile users concurrently can clearly deplete the backhaul and harm QoE. And even if higher backhaul capacities are available such as VDSL rather than ADSL, users reportedly [4] tend to also yield even larger traffic volumes when offered with higher capacities. Evidently, the solution lies in *prefetching* and *caching* content locally in small cells in order to offer low-access times to cached content and, as a result, to fully leverage the high throughput of small cells.

Regarding the role of third-party Wi-Fi hotspot contributors, they are strongly motivated to share access to their own resources in exchange for free access privileges to other owners' hotspots. Gaining access to public and private wireless resources can be also achieved with cooperative sharing mechanisms such as in [194, 195, 196] or through roaming agreements between providers and hotspot owners [197]. Note, that switching to Wi-Fi wireless from macro-cellular connections does *not* imply revenue loses for providers. Users merely replace macro-cellular access with access to Wi-Fi with *seemingly* no charges. Many providers offer deals which integrate both Wi-Fi home and outdoors Internet access through their UMTS or LTE macro BSs. In such cases, the providers suffer no essential revenue loses if their customers use a Wi-Fi connection supported by them or by means of [198]. On the very contrary, the providers manage to skip the extra cost of macro-cellular services without relegating user QoE due the higher Wi-Fi capacities. This is acknowledged by existing solutions in literature such as a market-based approach in [199], according to which macro BSs *pay* Wi-Fi hotspots for offloading mobile data through them in order to fight macro-cellular congestion and to enhance QoS.

5.1.6 Background synopsis and motivation

In summary, the most important points made in this section are:

- *Wireless networks struggle with an unparalleled mobile video demand:* Mobile video demand grows with an outstanding rate and poses an unprecedented challenge to wireless cellular providers. Particularly the *mobile video* traffic is projected to grow x11 and to account for 75% of the total traffic by 2020 relative to only 2015.

- *It is important to address niche, rather than popular demand:* Based on credible literature studies we argue that even if 60-80% of video accesses continuous to account for popular content in the future, a significant 20-40% of the total mobile video requests will still refer to niche demand. Moreover, CDNs are already highly successful with accommodating popular content and, notably, popular video requests. What we need is to address the remaining, less popular, post-CDN demand.

- *There are possible implications from niche personalised demand in Social Networks and the way CDNs are designed to address user demand:* The success of all popularity-based solutions, including that of CDNs, relies on the surprising skewness of truncated power-law popularity distributions that characterise demand in big video hosting websites. Still, this approach leaves niche demand largely unaddressed, whereas such high popularity skewness should be *not* taken for granted regarding the growingly popular user generated videos in Social Networks.

For all that, we argue for locally applied proactive caching solution that is beyond the scope of popularity-based approaches and which can efficiently address mobile demand for post-CDN niche videos. To our knowledge, that can be achieved via our contributed EMC model, along with its popularity and legacy-caching model extensions presented next in Section 5.2.

5.2 System model

We present our basic mobility-based cache decisions model and how it can be applied in both a "flat" and a hierarchical configuration of the available distributed cache space (Section 5.2.1). Furthermore, we present content popularity and legacy caching model extensions (Section 5.2.2), and demonstrate an application scenario for users of heterogeneous wireless networks who wish to reduce their monetary charges (Section 5.2.3). Last, we discuss and an analyse how (i) monetary charges or (ii) delay costs and their impact on QoE can be integrated in our model decisions (Section 5.3).

5.2.1 Basic Efficient Mobility-based Caching model

Cache actions with the basic Efficient Mobility-based Caching (EMC) model consider (i) the cost of fetching data from an *expensive* source, e.g. from a mobile carrier's macro cell or a remote source T_R, and (ii) the cost for consuming data from the local *cache point* T_L. "Cache points"[31] are mobile network APs, either Wi-Fi hotspots or small cells, equipped with low-access time storage resources which take cache actions that aim to *trade* their locally available buffer space efficiently for a reduced cost of obtaining data by their ephemerally hosted mobiles. Our model can be applied when $T_L < T_R$, i.e. when caching content closer to consumers or within the provider' own resources reduces transfer delay or data consumption charges for mobiles. Both T_R and T_L, which we discuss in more detail in Section 5.3, can be assessed per individual content or even content chunks, or per user. Additionally, such costs can be associated with certain content types (e.g. premium quality videos) and/or QoS requirements for certain categories of mobiles and applications.

Likewise to the EPC model (Section 4.3 on page 53), cache actions consider the cost of consuming cache space to capture the *trade-off* between occupying cache space for some content instead of another, which helps cache decision to achieve the

[31] Note that this definition is slightly different than the one given for the model of EPC, where the caches of the cache points are not necessarily collocated with the APs.

same optimization target (4.1) set for EPC on page 54 while satisfying the cache storage constraint (4.2). Since EMC captures requests in the long-tail of popularity distributions, demand expectations are higher compared to popularity-based solutions. In order to tackle cache congestion and to utilise the local storage resources efficiently, EMC uses the same *dynamic pricing* model presented in (4.3) on page 55 for EPC.

Autonomous caching in a "flat" structure of cache points

Figure 4.1 depicts a neighbourhood of caches organised in a "flat" structure in which cache decisions $b_s^l(t)$ in every cache point l are *autonomous* according to rule (5.1):

$$b_s(t) = \begin{cases} 1 & \text{if } \; Q_s^l(T_R - T_L) \geq p_l(t)\,, \\ \\ 0 & \text{Otherwise}\,. \end{cases} \tag{5.1}$$

The rule captures user mobility prediction and individual requests information via Q_s^l, which aggregates the individual transition probabilities $q_s^{i,l}$ of *all* mobiles with an active request for object s from their current cache point i to the target cache point l:

$$Q_s^l = \sum\nolimits_{\forall i} q_s^{i,l} \tag{5.2}$$

The left-hand side the rule is divided by the object's size to adapt the rule for different-sized objects, i.e. $Q_s^l \cdot (T_R - T_L)/o_s$. Also, the rule can be accordingly adjusted to address cases in which transfer costs are individual to different content, caches, users, etc, hence the latter can be accordingly adapted as T_L^s and T_R^S in (5.1).

Following a mobile's handover to l, (i) the mobile starts to consume s from the local cache (in case of a cache hit), while (ii) the rest of caches l' in the neighbourhood of i get notified to reevaluate $Q^{l'}$ and accordingly to redecide upon evicting or keeping s. Likewise, l redecides upon keeping s or not after its data are transferred to the mobile.

Regarding *where* decisions are taken, one option is for the requesting mobile (or some proxy) to inform the neighbouring caches about the corresponding mobile's transition probabilities; alternatively, the mobile (or its proxy) can decide after learning the transfer costs and the cache prices from all the neighbours. As for *when* should prefetching start and for *which parts* of the content, this is a function of (i) the mobile's expected handover and connection (a.k.a. residence) durations to the next cache and (ii) the time needed for prefetching data from the remote location(s) to the cache.

Problem hardness and the feasibility of an optimal solution

Finding an optimal cache placement for a flat set of caches given a set of requests for *equally-sized* objects of content can be obtained as follows: For each cache l we order requests in a decreasing order of value $Q_s^{i,l}(T_R - T_L)$. Then, starting from the request with the highest $Q_s^{i,l}(T_R - T_L)$, we fill the cache until the constraint B_l is reached. This procedure for obtaining the optimal is performed in *rounds*, unlike the *online*-applied solution based on cache congestion pricing, where cache actions are taken iteratively for each request according to formula (5.1). Furthermore, there is a serious practical issue regarding the duration of each round, which determines the number of requests that are considered in the beginning of a round. For objects with *different sizes*, the optimisation problem becomes identical to the *0/1 Knapsack Problem*, which falls within the class of NP-hard problems and for which the cache congestion pricing can have advantages towards approaching an optimal solution.

5.2.2 Model extensions

We extend our basic model to integrate cache *replacements* and/or to jointly exploit content *popularity* information along with user mobility prediction for cache actions. This yields two alternative extended model versions:

a) Efficient Mobility-based Caching model with cache Replacements (EMC-R), and

b) Efficient Mobility and Popularity-based Caching model with cache Replacements (EMPC-R)

The goal of the adapted extensions is twofold: it lies in better capturing temporal locality so as to prefetch and keep content that is more likely to be served to multiple users, and second, in exploiting *legacy-cached* content from past decisions in the case of EMPC-R, for which there are currently no active mobile requests.

Cache replacements

Adapting cache replacements causes two important differences relative to the basic model: (i) requests are not immediately satisfied or rejected with a decision rule like (5.1) and (ii) actions do *not* consider local cache prices. Instead, the extended model directly caches an object s if its size o_s can fit in the unallocated cache space, otherwise it explores the possibility of evicting one or more cached objects e, either legacy or active. To decide which object(s) to evict, we follow a procedure according to which cached objects e with size o_e are polled for eviction in order of increasing $G(e)/o_e$ until there is enough free space to cache s and as long as

$$\sum\nolimits_{\forall e} G(e)/\sum o_e < G(s)/o_s\,, \tag{5.3}$$

where $G(x)$ is the expected gain from caching (resp. keeping cached) content x, computed after a special *gain valuation formula* that is subject to the considered requests information (Section 5.2.2).

If evicting the next e violates (5.3) during this incremental process, then the cache request for s is dismissed and no objects are evicted. The purpose of (5.3) is to optimise the *total gain per utilised cache buffer unit* and it can be omitted if all objects have the same size, e.g. when cache decisions are taken on the level of equally-sized content chunks. Note that (5.3) works also as a heuristic for tackling the knapsack combinatorial optimisation problem that arises from maximising the total gain of the cached objects given their different individual gains, sizes, and the limited capacity of the cache.

Extended decision models

Next, we present two extended model variations with respect to gain valuation. Notice that only EMPC-R can exploit the possible benefits of legacy cached IOs e, i.e. objects for which $Q^l_e = 0$ due to the lack of currently active mobile requests.

1) EMC-R: This model variation adapts only the cache replacements extension, thus the gain valuation formula is defined as:

$$G(s) = Q^l_s \cdot (T_R - T_L)\,. \tag{5.4}$$

The resemblance to the decision rule (5.1) is evident, only without the comparison against the dynamic cache congestion price that exists in the basic model.

2) EMPC-R: This model variation adapts both cache replacements and content popularity to cache decisions, hence the gain valuation formula is defined as:

$$G(s) = (Q^l_s + w \cdot f^l_s) \cdot (T_R - T_L)\,. \tag{5.5}$$

This formula integrates both user mobility information Q^l_s and popularity information f^l_s. Whereas Q^l_s refers to mobile users who are not connected to l, as defined in formula (5.1) on page on page 82, f^l_s is the probability of an IO s to be requested by a user connected to l (such as a stationary user or a long term connected user), weighted by a special tuning factor w. This factor adapts a dynamic value that is used to *"tune"* the balance between mobility prediction and content popularity information on gain valuation by growing the significance of popularity expressed via f^l_s relative to mobility prediction expressed via Q^l_s. For instance, w can adapt to the relative rate between (i) the number of requests served by l to its currently attached users and (ii) the number of active mobile requests served by l, with both (i) and (ii) adapting dynamic values computed during one handover interval.

We can approximate f^l_s as:

$$f^l_s = T^l_{req}/I^l_s\,, \tag{5.6}$$

where T^l_{req} is the time duration between two consecutive requests for any object submitted to cache l and I^l_s is the time between two consecutive requests specifically for s submitted to l. If there are no past requests for s, we define $f^l_s \equiv 0$, as f^l_s reflects the contemporary popularity of s in cache point l with respect to the most recent requests' information about s.

Cooperative Caching: a two-level hierarchy

EMC as well as EMC-R and EMPC-R can be applied in a two-level hierarchy along the lines of the hierarchical and cooperative distributed caching procedure discussed in Section 4.3.2 on page 57. Consequently, an IO can be proactively fetched to a leaf, its parent mid-level, or even to both cache points, with leaves having one and only one mid-level parent as portrayed in the paradigm configuration of Figure 4.2 on page 58. Likewise, the cost for fetching data from the mid-level parent (T_M) must satisfy $T_L < T_M < T_R$. In addition, the solution procedure remains essentially the same compared to the one presented on page 58, with *only* minor adjustments needed to reflect the integration of the aggregated mobility information $Q_s^l = \sum_{\forall i} q_s^{i,l}$ in the cases of EMC and EMC-R (resp., $Q_s^l + w \cdot f_s^l$ in the case of EMPC-R) in the place of the individual mobility information q_s^l. Consequently, again each leaf takes a local cache decision about a specific IO s for the following two problems:

1. For the problem where s is assumed to be *not* cached in the mid-level cache point.
2. For the problem where s is assumed to be cached in the mid-level cache point by replacing T_R with T_M to decide if s should be prefetched to the leaf cache point.

Then, each leaf informs its mid-level parent about the expected transfer cost of s for each of the former problems. In the cases of EMC and EMC-R this is defined as:

$$Q_s^l \cdot T_R \;\text{ or }\; Q_s^l \cdot T_M\,, \tag{5.7}$$

if the decision is *not* to proactively fetch the IO, and

$$Q_s^l \cdot T_L \tag{5.8}$$

if the decision is to proactively fetch the IO to the leaf cache point. Correspondingly, in the case of EMPC-R:

$$(Q_s^l + w \cdot f_s^l) \cdot T_R \;\text{ or }\; (Q_s^l + w \cdot f_s^l) \cdot T_M\,, \tag{5.9}$$

if the decision is *not* to proactively fetch the IO, and

$$(Q_s^l + w \cdot f_s^l) \cdot T_L \tag{5.10}$$

if the decision is to proactively fetch the IO to the leaf cache point.

After receiving from all the leaf caches the transfer cost for the two problems, the mid-level considers the *summation* of the expected transfer costs for each problem:

- $\mathcal{T}_M^{mid}$, in the case the mid-level cache proactively fetches s and
- $\mathcal{T}_R^{mid}$, in the case the mid-level cache does not proactively fetch s.

The decision of whether to cache or not cache an IO in the mid-level is based on exactly the same decision process as in the case of the leaves. This means that in the case of EMC, the mid level cache uses the following rule:

$$b_s^{mid} = \begin{cases} 0 & \text{if } \; \mathcal{T}_R^{mid} - \mathcal{T}_M^{mid} < p_{mid} \\ \\ 1 & \text{if } \; \mathcal{T}_R^{mid} - \mathcal{T}_M^{mid} \geq p_{mid} \end{cases}$$

that is similar to (4.7), while for the cases of EMC-R and EMPC-R the mid level uses the gain-based cache replacement procedure discussed on page 83 by adapting gain

$G = \mathcal{T}_R^{mid} - \mathcal{T}_M^{mid}$. While this is straightforward to apply for EMC-R, we need to define w and f_s for the mid level in the case of EMPC-R. The simplest way for that, is via aggregating all the available information on both w and f_s from the leaves of the mid level cache.

Finally, as also noted on page 58, the solution remains distributed but requires some cooperation between the mid-level cache and its leaf children. Moreover, it can be applied to a hierarchy with more than one mid-level caches, as long as each leaf cache is a child of only one mid-level cache. The proactive caching decision for each mid-level cache and its leaf caches follows the above approach, and can be performed independently of other mid-level caches.

Problem hardness and the feasibility of an optimal solution: Introducing an intermediate level of caching between the leaf caches and the remote data sources turns the problem into a Generalised Assignment Problem [170]. The latter is a variation of the 0/1 Multiple Knapsack problem, involving multiple knapsacks, each with a possibly different maximum weight limit. At any time, there will be a given set of cache requests from the mobiles that are active at that time instant. For each such time instance, the problem for a two-level cache hierarchy has similarities with the Data Placement Problem (see discussion on page 52), where the probability of an IO being requested at a specific cache is given by the probability of the mobile moving to the corresponding network attachment point. In another work by the same authors [145], it is shown that the Data Placement Problem with different object sizes is NP-complete. Although the problem may be solved in polynomial time [172] in a hierarchical network with equally-sized objects, such solutions have a *high* polynomial degree and apply to an offline version of the problem.

5.2.3 Model application in Heterogeneous Wireless Networks

Heterogeneous Wireless Networks (HWNs) are characterised by the coexistence of different radio access technologies. Within the context of this paper, we assume HWNs as the one portrayed in Figure 5.1 on the facing page, that are comprised by one macro BS that is typically using either of the UMTS, LTE or Worldwide Interoperability for Microwave Access (WiMax) wireless communication standards, and multiple small cell BSs or Wi-Fi hotspots within its coverage. Each small BS or hotspot is equipped with a local cache storage and has a running instance of our model basic EMC or extended model to enhance mobility support for mobile users with *niche* requests who are primarily concerned about download charges. Such *cost-concerned* users may choose to postpone entire downloads of considerable volumes such as for high quality images, video or sound files until they connect to a small cell, or to download data concurrently from the macro and any ephemeral small cells connections in order to avoid the induced macro-cellular service charges. Unlike macro cells, there can be favourable charges[32]

[32] Similar to calling rates as described in `www.thinksmallcell.com/Operation/billing.html`

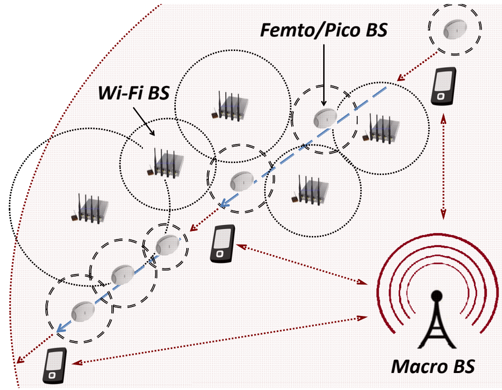


Figure 5.1: Macro-cellular coverage area including provider-controlled small cells and third-party Wi-Fi hotspots. Blue-dashed (resp. red-dotted) arrows denote mobility periods during which a mobile is connected to a small (resp. the macro) BS. Once a cost-concerned user gets connected to a small cell, she wishes to enjoy a good QoE by downloading video IOs with a minimum transfer cost. The desired minimum transfer cost combines low monetary charges and low delay/delay-jitterring to avoid experiencing stop-and-wait buffering times.

to motivate femto/pico cell usage, and many third-party Wi-Fi hotspots that offer a free-of-charge open-access model such as in cafeterias, campuses, shopping malls, etc.

Along these lines, social networking users are a notable example of cost-concerned users with niche demand requests: first, because they request for content that is dynamically assembled and tailored to their own personalised social network (Section 5.1.4); and second, because it is a common practice for many users to receive only notifications or small objects of content from the macro BS as they induce minimum charges, and to engage into data downloads of considerable size *later* via a Wi-Fi hotspot. In a similar manner, users running mobile chat services may postpone non-textual data downloads such as when their counterparts send them files or links to High Definition (HD) videos, images, etc.

Figure 5.1 depicts a cost-concerned mobile user and a set of distributed small cells within the range of a macro cell, each of which runs an EMC instance. The user moves across the macro coverage area, connecting to the available small cells along its path in order to avoid macro-cellular service charges. The blue-dashed arrows denote the periods during which the mobile is connected to a small cell and the red-dotted arrows denote the periods that the user is connected only to the macro BS. The location and the range of coverage of the small cells is up to device settings and landscape properties. Scenarios that involve clusters of contiguous coverage such as in the figure can denote the presence of a provider-orchestrated[33] coverage.

To apply our model, each running instance[34] has to attain information about the

[33] Such as in the case of wireless access devices installed on lighting facilities by Ericsson and Philips cooperation. Announcement: `www.ericsson.com/news/1763971`, and the case of the Los Angeles deployment: www.philips.com/a-w/about/news/archive/standard/news/press/2015/20151106-Los-Angeles-is-the-worlds-first-city-to-deploy-Philips-SmartPole-Street-Lighting.html

[34]The reader can refer to Section 5.5 for a more detailed description of the implementation details.

user's (i) transition probabilities, (ii) expected handover and ephemeral small-cellular "residence" session durations, (iii) content requests and, last, (iv) the corresponding data-transfer costs, in order to use them as input for the model's cache decisions. To better utilise the available backhaul throughput and storage resources during prefetching, requests can be fine-grained to the level of the content chunks *predicted* to be consumed during the mobile's ephemeral residences.

Soft **Vs.** ***Hard*** **cost-concerned users**

Cost-concerned users can be subject to either "soft" or "hard" requirements regarding delay, delay jitter or capacity. Accordingly, we can classify cost-concerned users as:

a) Soft users: Such users are primarily interested in minimising their macro-cellular charges. In case of a cache miss they are willing to download data exclusively via the (possibly) congested backhaul link of their hosting small cell or hotspot, or even to further wait until they attach to another small cell in order to complete their downloads.

b) Hard users: Users of this type choose to either switch to their macro link or to jointly utilise their Wi-Fi and UMTS/LTE interfaces in order to seamlessly proceed with completing their downloads as soon as possible, similarly to multi-path/source video streaming scenarios such as the ones studied in the work of [200].

Accommodating cost-concerned users with proactive caching at small cells and hotspots, particularly the hard ones who will anyway utilise their macro link, yields benefits for providers as well, as the behaviour of such users helps to *offload* significant traffic volumes corresponding to niche content demand from their capacity-saturated macro cells. But also the gains with respect to soft users can be significant for providers: for instance, reduced delay and/or a higher bandwidth available to users after attaching to a small cell can translate to a monetary benefit from satisfying users, as in this case the provider is in the position to offer higher QoS levels and, hence, promise an enhanced QoE for its customers relative to competitor providers.

5.3 Transfer costs integration

Our model's cache actions are designed to reduce monetary costs or download delay via trying to increase transfer cost gains T_R-T_L. *Monetary* costs may refer to user service charges for macro-cellular downloads, but also to charges and other expenses for providers, e.g. due to inter-domain data transit services used for fetching data from external sources. Delay on the other may refer to the propagation or transmission delay suffered by users when downloading their desired content with an impact on QoE.

In this chapter we adapt a user-oriented approach by addressing niche mobile requests for video content, hence our following discussion is focused upon (i) user monetary *charges* and (ii) download *delay*. Note that the analysis of Section 4.4 on page 59, which regards cost gains as a function of the IOs' sizes, applies also to our discussion here: either directly due to the analysed user utility of delay there, or by replacing delay with monetary charges in the user utility functions presented on page 60 and the corresponding graphs of Figure 4.3.

5.3.1 Monetary charges

We assume monetary charges M_{MC} and M_{SC} for macro-cellular and small-cellular usage in HWNs respectively, thus the corresponding transfer costs adapted by our model are:

$$T_L \equiv M_{SC}, T_R \equiv M_{MC} \tag{5.11}$$

Our model can be applied only when $M_{SC} < M_{MC}$ such when users utilise a third-party Wi-Fi hotspot connection free of charge or in case they are offered with favourable charges[35] for connecting to their own providers' pico/femto resources as a means of motivation for increasing the amount of the offloaded traffic to small cells.

5.3.2 Transmission delay

Delay implies an important cost for both users and their providers. It can cause video buffering breaks or lead to lower video qualities such as with the DASH protocol in order to match the offered level of QoS. Therefore, on the one hand there is delay as a metric, and on the other hand there is the users' own utility of delay which directly relates to the perceived user QoE and can contribute to providers' (monetary) cost for losing unsatisfied mobile customers.

In the following, we provide an analytical discussion on delay. The discussion has similarities with the one provided in Section 4.4 on page 59. Only here, we focus on the case of Wi-Fi/small cellular wireless connectivity, as well as on big objects and particularly video content. Therefore, the corresponding analysis is restricted to *transmission delay*, which is a function of the available throughput relative to the transited content volumes and the QoS requirements of the mobile application, and does not apply on propagation delay which is a function of the network hop distance that matters only for small objects.

Let R^{sc}_{W} be the throughput of the local wireless small cell or Wi-Fi interface and and R_{bkhl} be the corresponding throughput of the backhaul link or the path to the source of data via the backhaul. Also, assume that $R_{bkhl} < R^{sc}_{W}$, which justifies that prefetcing and caching data can yield benefits. Without proactive caching, the delay for downloading content s with size o_s is $d \equiv d_{bkhl} = o_s/R_{bkhl}$ as the throughput to a mobile is constrained by R_{bkhl}. However, *with* proactive caching, delay is $d \equiv d_{W} = o_s/R^{sc}_{W} < d_{bkhl}$. Due to the lower delay and the higher throughput levels $R^{sc}_{W} > R_{bkhl}$ that can be utilised to transfer more data to mobiles, the users' QoE can be improved such as by transferring

[35] `www.thinksmallcell.com/Operation/billing.html`

videos in a higher quality. If only a part x_s^l of s is prefetched to small cell l, then the remainder $o_s - x_s^l$ is compensated from a remote location via the backhaul. If the mobile requesting s moves to l, then the small-cellular delay for transferring the whole of s is:

$$d \equiv \mathcal{D}_{sc}(x_s^l) = \frac{x_s^l}{R_W^{sc}} + \frac{o_s - x_s^l}{R_{bkhl}} = \frac{o_s}{R_{bkhl}} - \left(\frac{1}{R_{bkhl}} - \frac{1}{R_W^{sc}}\right) x_s^l. \tag{5.12}$$

However, the above function assumes fetching the remaining data $o_s - x_s^l$ only form the local wireless interface via its backhaul, i.e. it does not cover the case of fetching data jointly from the Wi-Fi and the macro-cellular device interfaces. To extend the equation in order to include joint data consumption from both device interfaces, we need to consider the rate of the macro-cellular link R_W^{mc} and the part x_s^{mc} of s that is consumed from the macro cell. If only a part x_s^l of s is prefetched to small cell l and x_s^{mc} of s is consumed from the macro cell, then only the remainder $o_s - x_s^l - x_s^{mc}$ is compensated via the small cell's backhaul. Therefore, the overall delay D for consuming s is now:

$$d = Max\left(\frac{x_s^l}{R_W^{sc}} + \frac{o_s - x_s^l - x_s^{mc}}{R_{bkhl}}, \frac{x_s^{mc}}{R_W^{mc}}\right). \tag{5.13}$$

Finally, regarding the user valuation for delay d in transferring the IOs, we remind the reader that (s)he may refer to Section 4.4 on page 60 were we discuss utility as a function of delay $U_s(d)$.

5.4 Performance evaluation

We present an extensive performance evaluation conducted with a custom Java simulator built especially for evaluating our model and model extensions in Heterogeneous Wireless Networking application scenarios along the lines of our discussion in Section 5.2.3, All performance results show the mean monetary or delay cost *gains* from small-cellular data consumption as a percentage of the cost of the corresponding pure macro-cellular communication. Apart from EMC, EMC-R and EMPC-R, we include the performances of three benchmark models, namely, NAÏVE, Max Popularity (MAXPOP) and NOCACHE[36].

We begin our evaluation by studying the impact of different geo-spatial and wireless setup scenarios for HWNs on the performance of our basic EMC model in Section 5.4.1. The main setup parameters used in various combinations regard (i) the range of Wi-Fi or small-cellular coverage, (ii) the handover strategy adapted by the mobiles and (iii) the overall cache-storage supply over mobility demand ratio. Additionally, the simulation model implies *unique* demand by soft mobile users, which makes the conclusions drawn here suitable also for the case of adapting the EPC model discussed in Chapter 4 to HWN mobility scenarios. In Section 5.4.2, we evaluate EMC and its popularity and legacy-caching model extensions EMC-R and EMPC-R by adapting

[36] See the Appendix A.1 on page 148 for the definitions of NAÏVE, MAXPOP and NOCACHE.

complex synthetic traces of video demand, again in scenarios that involve soft users. Finally, in Section 5.4.3, we proceed with scenarios that involve hard mobile users engaging into concurrent macro/small-cellular downloads and by adapting both real and synthetic mobility traces, as well as real and complex synthetic traces of video demand, etc.

5.4.1 Geospatial and Wireless properties

All gain performance results presented in this section along with their corresponding 95% confidence intervals refer to 100 simulation repetitions. We adapt the simulation model presented below in Table 5.1. This setup is convenient for focusing our analysis exclusively on the performance impact of the different combinations of geospatial and wireless parameters. Notice that the simulation model implies *unique* demand by mobile soft users. Since unique, the assumed mobile requests are equally-popular and the proactive model has to consider only one mobile transition probability per object request. Thus, the results in this section refer only to the basic EMC model, leaving the popularity and legacy caching extensions for the latter evaluation sections.

Table 5.1: Simulation model parameter values for Section 5.4.1. All values included in the table are default values unless denoted otherwise in the graphs and excluding evaluation scenarios with a variable supply over demand ratio (S/D).

Parameters	Values
Topology:	21 small cells uniformly placed across a 700 m^2 macro-cellular area. The range of each small cell is fixed to 70 m.
Mobility model:	"Uniform" user mobility pattern. See Table 5.2 on the following page for details.
Cache supply:	Cache capacity supply of 100 IOs per small cell.
Gains upon hits:	Fixed value: 90% upon a cache hit.
User demand:	2800 mobile *soft* users initialised at random points of the area, each with a single, unique request.

Regarding the fixed gain upon a cache hit, the actual gain value is of little interest. What is important, is that users either have 0% gains upon a cache miss or a 90% gains by consuming the whole IO exclusively from the cache point. Finally, one important difference with respect to to the scenarios used in Section 5.4.3 is that mobile requests refer to to equally-sized IOs which are both cached and consumed in an "atomic" manner, i.e. they have the same impact on local cache prices and they are either cached and consumed as a whole or not at all. Consequently, IOs are either fully downloaded from the cache or from the macro cell upon cache misses.

Table 5.2: Mobility models (i.e., patterns) of probabilistic user mobility in space. All models integrate a ±2.5% probability jitter and unique velocities per mobile along the lines of daily human activities such as walking or driving a car. Note that each model creates the necessary conditions to approximate (yet *not* to impose) the corresponding handover probabilities between cells, as this is also subject to the location and range of the small cells or hotspots.

Notation	Description
Uniform	The number of mobiles moving along each direction is uniformly equal.
SKDX%	Skewed mobility model with "X%" of the mobile users moving towards the same direction. If X<100%, then the remainder (100-X)% mobiles adapt "uniform" mobility.

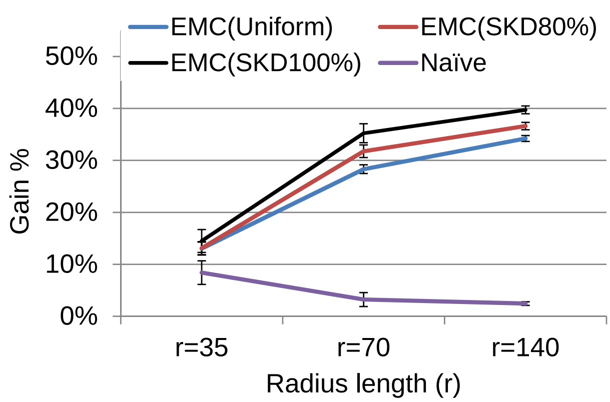


Figure 5.2: Impact of user mobility on performance using 35 small cells with a short (r=35 m), medium (r=70 m, default setup) or long radius (r=140 m).

User Mobility

Figure 5.2 shows EMC and NAÏVE gains for the mobility models of Table 5.2. The more skewed the mobility model is, the greater are the performance gains for EMC. This is expected due to the fact that mobility prediction becomes more accurate with increased mobility skewness, hence mobile decisions can better utilise the available buffer space resources. Nevertheless, even EMC (Uniform) manages to utilise its buffer space better than NAÏVE by ∼6% due to the cache pricing scheme adapted in EMC. In another conclusion, the graph shows that *increasing* the *radius* length r of the small cells can significantly increase the performance of EMC. The former causes demand to increase, which leads to a better cache-storage utilisation in the case of EMC due to its congestion pricing mechanism. On the very contrary, exposing NAÏVE to a higher demand degrades its performance. Its gains difference from EMC range from ∼6% for $r = 35$, to ∼25-32% for $r = 70$ and even to ∼31.8-37.3% for $r = 140$, as NAÏVE (i) lacks the ability to handle higher demand and (ii) neglects mobility and cache space congestion.

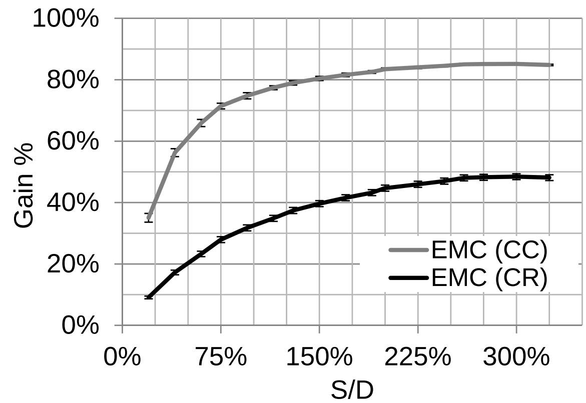
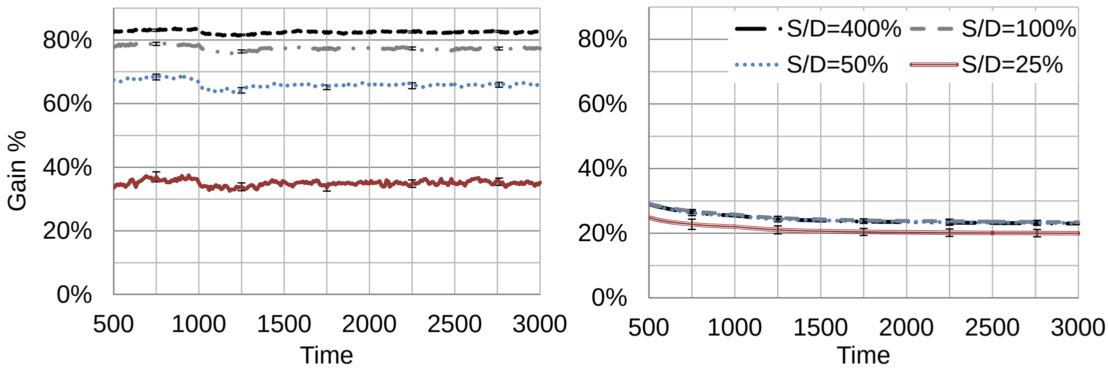


Figure 5.3: Impact of handoff policies on performance. The X-axis shows the supply of storage-cache from 21 small cells as a percentage of total demand from all mobiles.

Handover strategies

Figure 5.3 shows EMC gains as a function of the total cache supply over total demand for two handover strategies, namely: *a) Cached Content (CC)*, which allows mobiles to attach to cells that have prefetched their requested content; and *b) Closest Range (CRn)*, according to which mobiles attach to the closest cell within range. The former policy is closer to ICN, while the latter is closer to current wireless architectures along the line of which the mobiles offered with multiple connectivity options make a selection either randomly or based on load and other wireless conditions metrics in seek for higher data-rates. Data-rates can be impacted by various factors such as the number of hosted mobiles or their distance from BSs. Evidently, handover probabilities are driven by cache decisions with CC, whereas cache decisions are in turn based on handover probabilities. Fortunately, dynamic pricing in mitigates the possibility of a bias for mobile attachments to specific small cells or hotspots, as with higher demand comes a higher cache price that balances cache decisions and consequent CC-based handovers.

Returning back to the results of the graph of Figure 5.3, both performances continuously increase, yet with a decelerating growth rate. Performance is higher for CC by ~37.5-42% as users maximize their gain by attaching to the cells which offer their requested objects directly from a local cache. Note though that the presented gain percentage measurements say nothing about actual data-rates, hence *hybrid* CC/CRn strategies may combine cache hits and corresponding low access latencies with higher wireless rates in order to improve QoE and satisfy users. Note, that towards this direction, the work of [201] on information-aware connectivity decisions combines link-layer connectivity and information-availability. Also, the work of [202] investigates the combination of caching with dynamic user associations.



a: EMC b: Naïve

Figure 5.4: Transient performance for different supply over demand ratios (S/D).

Transient performance

The graphs of Figure 5.4 show the transient performance of EMC compare to Naïve for different levels of storage supply over a fixed level of demand. Specifically, the total cache supply remains equal to 2100 IOs throughout the simulation, as there are 21 small cells with a local cache that can fit up to 100 IOs, while a different demand value is adapted by requesting more than one IOs per mobile user, accordingly.

1) *EMC gains adapt quickly to changes in demand and user mobility:* Both graphs show that gains *converge* to a *steady-state*. For EMC, this happens *shortly* after the beginning of the simulation, i.e. from a initial state of *no* knowledge on information about user mobility and requests. This strongly indicates that EMC *can adapt quickly to changes* in demand and user mobility.

2) *Expanding cache buffer can add significantly to the performance of EMC:* Furthermore, Graph 5.4a shows that doubling supply S over demand D from 25% to 50% nearly doubles average gains from ~36% to ~65.5%, whereas further doubling S/D to 100% or 200% corresponds to a ~13.5% and ~5% of extra cost gains on average. Evidently, cache expansion can yield significant performance benefits in the case of EMC, particularly, when increasing small capacities.

3) *EMC vastly outperforms* Naïve*:* On the very contrary, cache expansion offers minor extra gains to Naïve. As portrayed in Graph 5.4b, transient performances for Naïve overlap for $S/D \geq 50\%$, being broadly lower than EMC. Recall that Naïve's performance in Figure 5.2 on page 92 shows its inability to handle higher demand due to neglecting mobility and cache space congestion. Likewise, here we observe that it can not exploit higher cache buffer supplies for the same reason.

5.4.2 Video Catalogue Size and Mobile Demand

In this section we adapt a synthetic trace of video file requests produced with the GlobeTraff workload generator [183] in order to study the impact of diferent mobility demand levels and video catalogue sizes on performance gains. As with the former simulations, we assume soft mobile users but adapt a different simulation model setup that is summarised in Table 5.3:

Table 5.3: Simulation model parameter values for Section 5.4.2. All parameter values are default values. The setup tries to simulate human walking speeds across an area that resembles, e.g., a university campus or a fest field.

Parameters	Values
Topology:	25 small cells uniformly placed across a 700 m^2 macro-cellular area. The range of each small cell is 71 m on average with standard deviation 5 m, implying that most of the area is covered by the small cells.
Mobility model:	"SKD80%" user mobility pattern with speed 5 Km/hour and standard deviation 1.25. See Table 5.2 on page 92 for details on mobility models.
Cache supply:	Cache capacity supply of 8 GB per small cell. Total cache supply from all small cells is $S = 200\ GB$.
Gains upon hits:	90% on average upon a cache hit. Dependant on network hop distance (see detailed description).
Video requests:	5 synthetic traces produced with GlobeTraff. Zipfian parameter s=0.90.
User demand & video catalogue:	1000 mobile soft users initialised at random points of the area. Requests are updated when entering the area and upon completing a handover. Four scenario combinations of mobility demand over total cache supply and video catalogue size over local buffer size ratios (see detailed description): • low D/S=0.4 and high D/S=10, • low |VC|/BF=21.5 and high |VC|/BF=301.

Gains upon hits: The cost T_R upon a cache miss depends on the hop distance and the delay for transferring data over the access network connecting the small cell to the core network. Specifically, $T_R = n + \frac{5}{8}$, where the number of hops n is normally distributed with mean 4.2 and standard deviation 1.05, based on the work of [174] which shows that the inter-AS path length has remained practically constant and equal to 4.2 for over a period of 12 years, while approximately 95% of the AS-hop distances fall in the range [2.1, 6.3]. The factor $\frac{5}{8}$ reflects the higher delay for transferring data over the provider's access network. The above give an the average transfer cost for a cache miss $T_R = 10$, while the cost of a cache hit is $T_L = 1$. Based on this, an upper bound for the reduction of the total cost, compared to the total cost when caching is not used, is 90%.

User demand & video catalogue size: We adapt two mobility-based proactive caching demand over total cache storage ratios, low D/S=0.4 and high D/S=10, and two Video Catalogue (VC) over cache size ratios, low |VC|/BF=21.5 and high |VC|/BF=301. The former yield four scenario combinations in total as presented in Figure 5.5 on the next page. The average video file size is set to 80 MB, while popularity and temporal locality is defined by GlobeTraff as described in [183]. To increase demand compared to a *single 80 MB video request* per each mobile user in the case of *low D/S* scenarios, we adapt *25 video requests* per mobile user in the case of *high D/S* scenarios. Such an extreme IO demand per simulated mobile can refer to running, e.g., social networking and likewise applications which prefetch and auto-play videos in user's devices, or to simulating groups of users walking together per simulated mobile, or to requesting for very large videos such as a whole movie, or to any possible combination out of the aforementioned.

Table 5.4: Cache gains with respect to a low and a high mobile video caching demand over total cache storage ratios (D/S), combined to a low or large video catalogue size over local cache buffer size (|VC|/BF).

D/S	|VC|/BF	EMC	EMC-R	EMPC-R	MaxPop	Naïve
Low	Small	81.4%	89.1%	89.1%	49.0%	62.6%
Low	Large	49.2%	78.7%	79.0%	8.5%	30.1%
High	Small	46.2%	76.8%	77.2%	51.0%	36.7%
High	Large	11.8%	32.3%	32.5%	8.5%	11.6%

Main conclusions

The main conclusions out of the exhibited performance gains in the four scenario combinations of Table 5.4 are summarised below. Note that Figure 5.5 on the next page portrays a visual representation of these conclusions.

1) *Low demand for mobility-based proactive caching & small video catalogue:* This corresponds to the bottom-left area in Figure 5.5, hence both mobility and popularity-based caching yield benefits. For this reason, as well as because replacements help to reach closer to an optimal allocation of the cache space, the gain of EMC-R/EMPC-R is highest. The gain of EMC is also high, verifying that it indeed indirectly captures popularity through the aggregate mobile transmission probability in the proactive caching decision. The gains of MaxPop and Naïve are high, relative to their gains in the other scenarios, indicating that they exploit knowledge of popularity and neighbouring caches, respectively.

2) *Low demand for mobility-based proactive caching & large video catalogue:* This case corresponds to the bottom-right area in Figure 5.5, hence only mobility-based

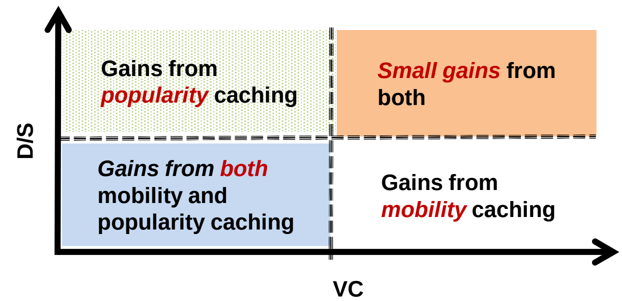


Figure 5.5: Summary of conclusions on performance gains from Table 5.3 with respect to various combinations of mobility demand over supply and VC size over local buffer sizes.

proactive caching yields benefits. For this reason, the gains for EMPC-R and EMC are the highest, significantly higher than MaxPop and Naïve. MaxPop has the lowest gains.

3) High demand for mobility-based proactive caching & small video catalogue: This case corresponds to the top-left area in Figure 5.5, hence content popularity caching yields the most benefits. For this reason, the gains for EMPC-R and MaxPop are highest, followed by EMC. The Naïve scheme has the lowest gains.

4) High demand for mobility-based proactive caching & large video catalogue: This case corresponds to the top-right area in Figure 5.5, hence both mobility and popularity-based caching yield small benefits. Nevertheless, the gains for the EMPC-R scheme are the highest, but lower than its performance in the other cases. The other schemes have small gains, while the gains of MaxPop are the smallest.

General conclusions

Gains with cache replacements are the highest in all cases. EMC, which considers mobility-based proactive caching and only indirectly content popularity, exhibits worst performance compared to MaxPop when the demand for mobility-based proactive caching is high, due to the small benefits of mobility-based caching in this case. MaxPop has its highest gains when the video catalogue size is small, since in this case caching the most popular videos is more beneficial compared to the case where the video catalogue is large.

The Naive scheme has its highest gains when the demand for mobility-based proactive caching is low, since it considers knowledge of neighbouring caches in its caching decisions. However, these gains are smaller than those achieved with our models, verifying once again (as throughout the results of Section 5.4.1) that utilising mobility information and cache congestion pricing are beneficial.

Finally, EMC-R and EMPC-R appear to have no significant performance differences despite the fact that EMPC-R also exploits content popularity information. However, this is *not* a general truth. It merely holds for the examined scenarios in the presented results. To provide more insight on the added value of popularity information in EMPC-R, we proceed with further investigating the impact of mobility skewness in the next subsection.

Impact of different mobility skewness

The results of Table 5.4 are for a skewed mobility model (SKD80%), whereas one would expect the contribution of mobility to be less significant relative to content popularity for mobility models closer to uniform. In what follows, we use different mobility skewness levels to examine the validity of our intuition.

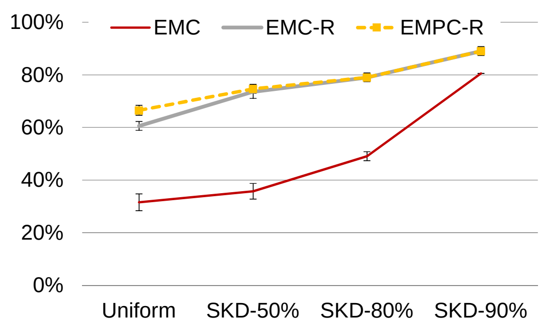


Figure 5.6: Impact of mobility skewness on gains.

Figure 5.6 shows the impact of mobility model skewness on the average performance gains for the case of a low demand for mobility-based proactive caching and a large video catalogue. As expected, a higher skewness leads to higher gains for all of our models and closes the performance gap between EMC and EMC-R/EMPC-R by ∼20.5%/26.5% for SKD90% relative to uniform. Yet, the main conclusion drawn from this graph is that *content popularity* adapted in EMPC-R leads to *increased gains* compared to EMC-R for uniform mobility. Evidently, the closer the mobility model of the users is to uniform, the *less* is the *utility* of the *mobility* information used by our models. In the depicted results for uniform, this translates to an added ∼6% compared to EMC-R and ∼35% compared to EMC. However, we also see that even if just a half of the users move towards the same direction (SKD50%), the utility of the mobility information is enough to yield marginally the same performances for EMC-R and EMC-R.

5.4.3 Mobile Video gains for hard users

In this part of the evaluation we provide a meticulous performance study of EMC, EMC-R and EMPC-R with respect to user charges and delay gains in scenarios that involve mobile video demand. Table 5.5 on the facing page provides a summary of the default parameter values used in the performance results presented throughout this section and a more detailed description provided below:

Table 5.5: Simulation model default parameter values for Section 5.4.3. All values are default except those designated with ‡. These values are used in evaluation scenarios that explicitly state the use of a non-default parameter value.

Parameters	Values
Wireless throughput:	1‡, 2 and 8‡ Mbps to macro-cell; 4 Mbps to small cell; 2 Mbps from small cell's backhaul.
Video requests:	• 5 synthetic traces produced with GlobeTraff, each with respect to a catalogue of 100K videos and zipfian parameter s=0.75. • Video requests from a real web traffic trace‡.
Topology & user mobility:	22 small cells stochastically distributed over a 1000^2 m area. The range of each small cell is 70 m on average with standard deviation 5 m. User mobility: • A real taxi cab mobility trace (536 taxis). • Synthetic walkers mobility trace‡.
Mobile & Stationary demand:	1 active request per mobile. Once fully consumed, the video request is replaced with a new one. 20 "stationary" (long-lasting) connections with a standard deviation 5.
Cache capacity:	2‡, 4 and 8‡ GB cache-storage buffer in each small cell.
Cost gains:	• Monetrary: Macro-cellular (resp., small-cellular) transfers cost 10 (resp., 0) monetary units per downloaded data unit. • Delay: Macro-cellular (resp., small-cellular) transfers cost 10 (resp., 1) per downloaded data unit.

Wireless throughput: Mobiles have a constant 2 Mbps access to the macro-cell and an extra 4 Mbps of wireless throughput when linked to a small cell. Mobiles jointly download data from the macro cell and the small cell (either from the local cache or the backhaul of the small cell) during their sojourn in the small-cell. Apart from the default values, we also study performance with respect to different macro-cellular throughput values in on the next page.

Content requests: We use a set of five different synthetic traces produced with the GlobeTraff [183] workload generator, which comply with the models of [189, 203] on requests popularity and content sizes. Each trace refers to a different VC of 100K videos with average file size ∼100 MB, split into ∼2.5 MB chunks. We adapt s=0.75 as the default value of the Zipf distribution exponent parameter used for video popularity and also study performance with respect to (i) a series of different exponent parameter values $s \in [0.1, 1.5]$ on page 102, and (ii) real web traffic [204] after filtering-out non-video requests on page 103.

Topology & user mobility: We use 22 small cells with an average range of 70 m and with standard deviation 5 m stochastically distributed over a 1000^2 m area, along with a real mobility trace [205] based on GPS data of 536 taxi cabs over a period of 30 days in the San Francisco Bay Area, USA. The chosen small cell density corresponds to data about the area extracted from the publicly available WiGLE [206] database and after grouping[37] together Wi-Fi hotspots within less than a 100 meter distance from each other. Besides the default setup, we adapt different mobility traces in on page 104 and study the impact of alternative densities in on page 107.

Mobile & Stationary demand: Mobiles have 1 active request at a time which is *jointly* served via the macro and the small BSs. During a mobile's *"residence"* in a small cell, it consumes data from the local cache in parallel to consuming the non-cached video chunks from the small cell's backhaul and the macro link. Apart from the mobiles, small cells also host "stationary" devices with long-lasting connections, e.g., laptops. We assume an average number of 20 active stationary requests in each target small cell, which last for a time that is equal to the average mobile *handover time* to the small cell. Also, we study the impact of the number of stationary requests in on page 105.

Cache capacity: We assume 4 GB cache-storages for simulating a low-cost, highly-distributed HWNs along the lines of Section 5.1. Also, we compare the performance of our model against different cache storage capacities in on page 107.

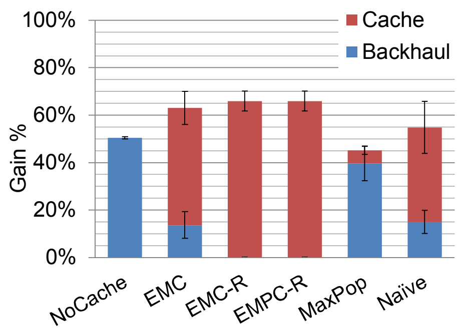
Cost gains: Macro-cellular (resp., small-cellular) transfers cost 10 (resp., 0) monetary units per downloaded data unit. Recall that there can be gains even upon cache misses due to utilising the backhaul of a small cell. Also, there may be limited or *no* gains at all when a mobile's transition to the next small cell lasts for too long, and limited gains for mobiles which stay connected to small cells for short periods. In the first case, most or even all of the data are consumed by the mobile from the macro cell while in transit to the next small cell, and in the latter case, the sojourn time in the small cell(s) is too short to exploit either the cached data or the data from the backhaul of the small cell. Last, apart from monetary, the results presented on page 108 adapts a series of cost combinations which refer to transfer delay.

Performance against benchmarks and macro-cellular throughput

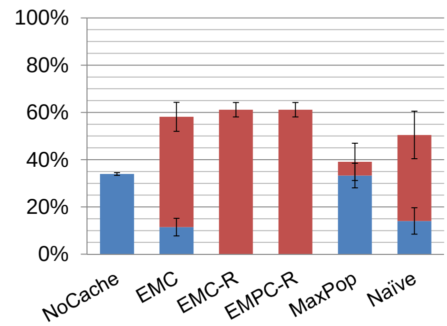
Figure 5.7 on the next page shows performance for 3 different wireless macro-cellular throughput values MC_t: (i) low throughput (Graph 5.7a); (ii) average throughput (Graph 5.7b); and (iii) high throughput (Graph 5.7c).

As a general comment, we observe that all performances drop with MC_t. This is expected because the mobiles consume an increased part of their requests from the macro cell during both their handover and residence periods. The extend of the drop is better perceived via observing NoCache: its performance falls from 50% to 34% and from there to 20% in correspondence to a 2x and an 4x MC_t increase respectively. The

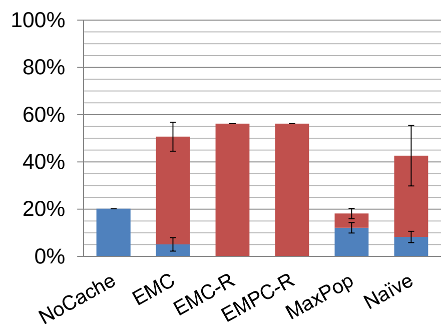
[37] A similar merging process is adapted in [54].



a: $MC_t = 1\ Mbps$



b: $MC_t = 2\ Mbps$



c: $MC_t = 8\ Mbps$

Figure 5.7: Performance with different macro-cellular wireless throughput levels. Graphs portray a performance breakdown between download gains from the cache (in red) and from the backhaul link (in blue). Note that Graph 5.7b adapts the default setup described on on page 99.

impact on cache gains is less for EMC (∼-1.5%) with each MC_t increase compared to the extended model variations, whose gains fall from 66% to 61% and 56% respectively. Last, cache gains for MaxPop are robust but low (6%), and Naïve cache gains drop by 4% and 2% for each corresponding MC_t increase.

Next, we analyse the performance against the benchmarks. Without loss of generality, we focus on the average-throughput scenario of Graph 5.7b, which is the default scenario discussed on page 99; yet the same conclusions apply to the rest scenarios as well:

1) *Maximum gains from cache with replacements:* The absence of backhaul gains for EMC-R and EMPC-R is due to exploiting the available small-cellular wireless throughput exclusively with data straight from the local cache. This achievement comes as a result of the added efficiency of cache replacements that yields ∼14% more cache gains relative to EMC.

2) *EMC yields a robust and overall good performance:* The difference between EMC and EMC-R/EMPC-R is *only* 3% when including backhaul gains. This outcome is *highly important* as cache replacements imply a considerable computational over-

head relative to the basic model (Section 5.5.6).

3) EMC and EMC-R adapt well to temporal locality: Even though only EMPC-R directly addresses content popularity, EMC and EMC-R decisions appear to adapt well to contemporary popularity conditions via the aggregated transition probabilities Q_s^l of the mobiles with an active request for content s. Such *short timescale*[38] and dynamic *mobility information* help to adapt quicker and better to temporal locality conditions than long-term popularity information. A comparison against the cache gains of MaxPop (6%), verifies that EMC corresponds better to temporal locality than the long-term popularity used by MaxPop, thus exceeding the latter's performance by 41% (19% including the backhaul). Also, EMC-R and EMPC-R appear to have the same performance. This does *not* imply that popularity information in (5.5) is needless, as it allows to decide upon legacy-cached objects with the cache replacements extension.

4) Intelligent Vs. Naïve *caching:* Naïve has 11% less cache gains (8% including the backhaul) relative to EMC. This fair performance difference is due to (i) the total user *demand* of the taxi trace, which is generally *not* high. Many out of the simulated taxi cabs can have long and varying handover periods that make them consume most (or even the whole) of their requested content before entering a small cell. Moreover, (ii) Naïve decisions are not totally "blind" and regard only a part of the requested contents (Section 5.5.4), while the gains of intelligent caching can be much higher against a "pure" Naïveas we show in Sections 5.4.1 and in 5.4.1. In addition, (iii) Naïve's gains come with larger confidence intervals, which implies that EMC and especially EMC-R/EMPC-R are less susceptible to demand fluctuations.

5) Compared to NoCache*:* Gains from mobility-based proactive caching are significant compared to using only the backhaul. EMC (resp., EMC-R/EMPC-R) outperform NoCache by ∼24% (resp., 28%), while MaxPop by only 6% and Naïveby 16%.

Content popularity skewness

Figure 5.8 on the next page shows the gains of our model as a function of the Zipfian video popularity distribution exponent parameter (s). We use a fixed 100 MB size for all videos so as to focus exclusively on the impact of content popularity *skewness* as it increases with s. The results show a robust performance for the model extensions and an increasing trend for EMC. EMC better utilises the cache with more skewed content popularities (greater s), which is expectable as cache decisions regard mostly specific content.

[38] An outcome which largely coincides with [54], which concludes that "We need to take into consideration the latest mobility information from nearby devices to make accurate predictions" with respect to urban environments.

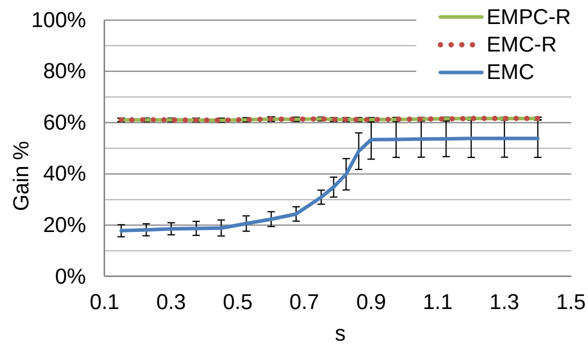


Figure 5.8: Performance against an increasing Zipf exponent parameter s. Content popularity skewness increases with s.

Also, its performance approaches to that of the model extensions, being only ∼7.6% lower for $s \geq 0.9$. Note, however, that our model targets niche requests, while practice shows that the highly popular content can be anyway addressed by CDNs, (see also the results on page 110).

Real trace requests

Figure 5.9 presents gain performance with respect to a real trace of web requests [204]. A comparison between Graphs 5.7b and 5.9 reveals that EMC gains are lower than with GlobeTraff: Cache gains are less by 14% (53% including the backhaul). Opposite to EMC, the gains of the extended models are increased by 2%. Gains for Naïve also drop, remaining lower by 9% relative to EMC.

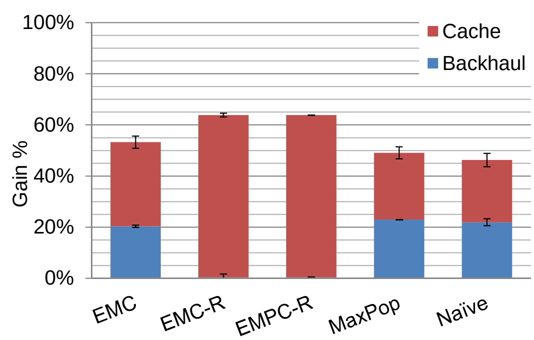


Figure 5.9: Performance using real video requests.

Interestingly, MaxPop's cache gains are 20% higher (10% including the backhaul) compared to Graph 5.7b, along with smaller confidence intervals. Additionally, they exceed Naïve's cache gains by 2% despite being 30% lower than Naïve's in Graph 5.7b. This dramatic increase can be due to an increased popularity skewness and/or due to the size of video catalogue (see results in Section 5.4.2).

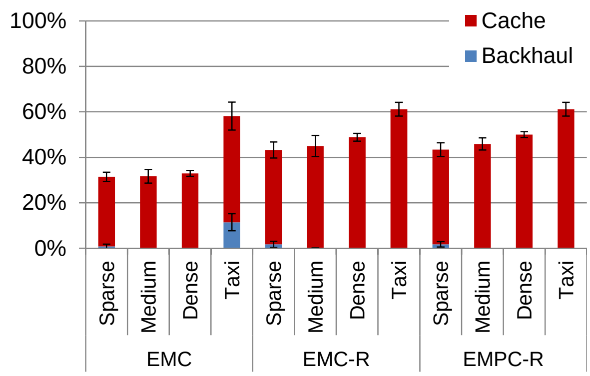


Figure 5.10: EMC and model extensions gains for the taxi cab [205] and the KTH/Walkers [207] mobility traces.

Walkers mobility traces

We use the “KTH/Walkers” dataset [207] as a benchmark to the results with the taxi cab mobility trace and present the results in Figure 5.10. We adapt the same small cell density as with the taxi cab trace, to ensure a fair comparison between the corresponding results, only for a 400 × 400 area to which the KTH/Walkers traces refer to. The walkers trace comes in three versions with respect to the density of users in space: (i) “sparse”, (ii) “medium” and (i) “dense”. User density with the taxi trace is stable and closer to “medium” relative to the rest. However, density can vary with time with KTH/Walkers. In addition, walking speeds are less variable than the taxi driving speeds which can be anywhere between very slow (e.g., due to traffic jams or traffic lights) and very high. These differences lead to a smaller level of cache demand with KTH/Walkers, either because the active users are at times less, or because the handover or residence periods last less, which implies that less data can get prefetched to the caches or consumed while being connected respectively. As a result:

1) *EMC cache gains are lower:* They drop by ∼15% and there are no backhaul gains as the residence in the small cells lasts too little to utilise the backhaul link.

2) *User density appears to have small impact:* There is no impact on EMC and only a small one on EMC-R/EMPC-R whose gains increase from 42% to 45% and 47% for “sparse”, “medium” and “dense” respectively. Given the smaller residence in the cells compared to the taxi trace simulations, evidently replacements manage to reach closer to an optimal allocation of the cache space. Interestingly, there are some backhaul gains (0.5-1.7%) for “sparse”. The corresponding cache gain differences from the taxi trace are: 19% (∼17% including the backhaul), 16% and 12%.

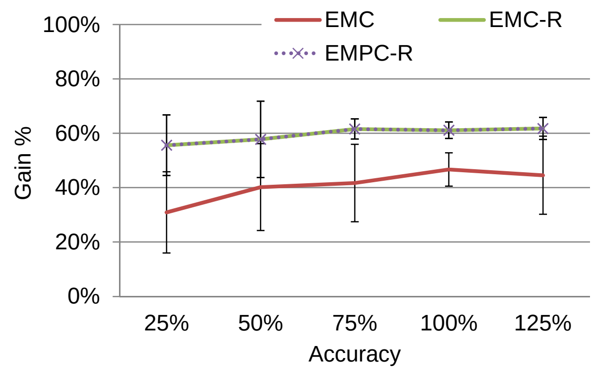


Figure 5.11: Performance against different mobile prediction accuracy levels. Assume handover time h. Accuracy x on the x-axis denotes that mobile prediction considers time $x \times h$, i.e. the performance for $x = 100\%$ corresponds to the most accurate prediction. The same applies with respect to the predicted residence time.

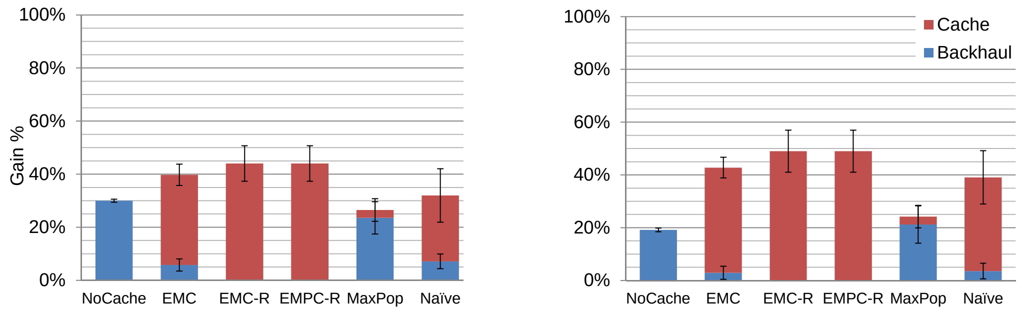
Mobility prediction accuracy

As discussed in Section 5.5.4, the prediction of the *consumable* part of a mobile's content request is an integral part of the cache allocation process that is based on the mobile's predicted handover and residence duration times. Inaccurate prediction can degrade performance because of requesting more (resp., less) buffer space from the cache than what is actually needed, in which case requests waste (resp., underutilise) buffer space.

To clarify the impact of these mobility predictions on performance, we present the graph of Figure 5.11 which shows the performance against different accuracy levels. As explained in the legend, prediction is most accurate for $x = 100\%$, while it can lead to buffer underutilisation (resp., waste) for $x < 100\%$ (resp., $x > 100\%$). Indeed, the graphs shows that gains increase with accuracy and that the best average performance along with the smallest confidence intervals corresponds to $x = 100\%$. There is though a difference between the basic and the extended models. The latter converge earlier and preserve their performance for $75\% \leq x \leq 125\%$ due to cache replacements, while gains for EMC are lower and the corresponding confidence intervals are more than twice larger for $x \neq 100\%$.

Stationary user requests

Figure 5.12 on the following page shows the aggregate performance for all users' requests, both mobile and stationary, for a different percentage of mobile users against the number of stationary user requests issued in every small cell. The total number of connected mobiles per small cell converges to $\sim$7 within an average handover interval from any small cell to another one and can, therefore, differ between different couples of source/destination small cells. Cache decisions are the same in both graphs due to using the same input information: requests, transition probabilities, handover and residence



a: #MU25% b: #MU50%

Figure 5.12: Overall performance with respect to requests by both mobile and stationary users. Graph 5.12a (resp., 5.12b) corresponds to a scenario of 25% (resp., 50%) of the total number of all downloads accounting for mobile requests, hence the notation #MU25% (resp., #MU50%). Note that Graph 5.12b adapts the default setup described on page 99.

times, long-term popularities.

Our conclusions are summarised as:

1) *Less backhaul gains for less stationary requests:* Stationeries have less hits as cache actions are based on mobiles' requests. Thus, they consume more data from the backhaul than the mobiles, which explains the reduced backhaul consumption in Graph 5.12b by 10.8%, 2.9% and 3.6% for NoCache, EMC and Naïve respectively. Notice
maxpop's backhaul decrease by 3% and the higher impact on NoCache because of the proportionally more mobiles in Graph 5.12b that consume data also from the macro cell.

2) *Cache gains increase for less stationary requests:* Cache gains are higher in Graph 5.12b by 5.9%, 5.0% and 10.6% for EMC, EMC-R/EMPC-R and Naïve respectively, due to the increased proportion of mobile consumers. The former outweigh the corresponding backhaul loses, thus the aggregated backhaul and cache gains increase by 3%, 5% and 7.2% for EMC, EMC-R/EMPC-R and Naïve, respectively. Note that cache gains remain stable for MaxPop and approximately equal to 3%.

3) *EMC-R and EMPC-R have the same performance:* Caching with EMC-R takes content popularity into account via $w \cdot f_s^l$ in formula (5.5), where w balances online the impact of popularity f_s^l based on the number of requests served to attached users during a handover interval. Thus, w adapts a different value in the two graphs, but only causes a marginal gain difference compared to EMC-R, verifying our former conclusion on page 100 that the model can adapt well to temporal locality via *only* mobility information.

Small cell density

Figure 5.13 shows performance gains for three different densities over the 1000^2 m simulation area: (i) "sparse": 11 small cells; (ii) "medium": 22 small cells; and (iii) "dense": 55 small cells. Increasing the density increases cache gains in a twofold way: First, because it increases the available cache supply and, second, because it increases (resp., decreases) the aggregate residence time in small cells (resp., handovers) during which the mobiles can download more cached or backhaul data from the small cells (resp., the mobiles can download only from the macro cell). In addition, the prediction of mobiles' residence and handover times based on historic data becomes more accurate (see related results on page 105), hence cache actions, particularly with cache replacements, can avoid video chunks that will not be requested during mobiles' residence.

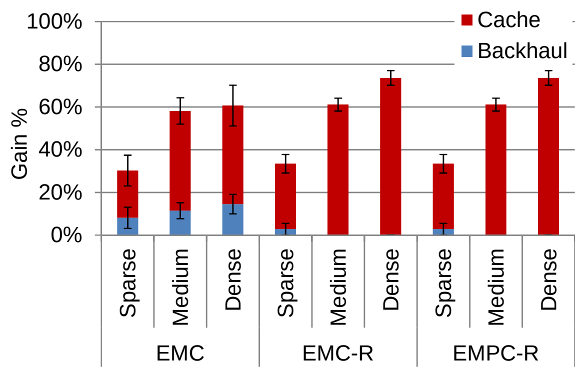


Figure 5.13: Performance against different small cell densities.

To sum up:

1) *Cache gains increase with cell density:* Doubling the number of small cells from sparse to medium increases from 22% to 47%, and EMC-R/EMPC-R cache gains from 31% to 61%. This translates to a x2.14 and a 1.96x increase for EMC and EMC-R/EMPC-R respectively. Further increasing density from medium to dense by 2.5x has no impact on EMC cache gains but it adds up 12% to the cache gains of EMC-R/EMPC-R.

2) *Backhaul gains increase with density only for EMC:* Backhaul gains for EMC also increase with density from 8% to 11%, and to 15%. However, the few backhaul gains (~3%) with a sparse density for EMC-R/EMPC-R cease to exist. While a greater residence duration helps EMC to cover up for cache misses with backhaul data, it helps EMC-R/EMPC-R caching to become even more efficient and not need to compensate from the backhaul.

Cache storage supply

Figure 5.14 on the next page portrays the performance of our model variations for three different cache storage sizes in each small cell: (i) 2 BG, (ii) 4 BG and (iii)

8 BG. The results have similarities with Figure 5.13 due to increasing the available cache supply, yet with less significant added cache gains. Our main conclusions are as follows:

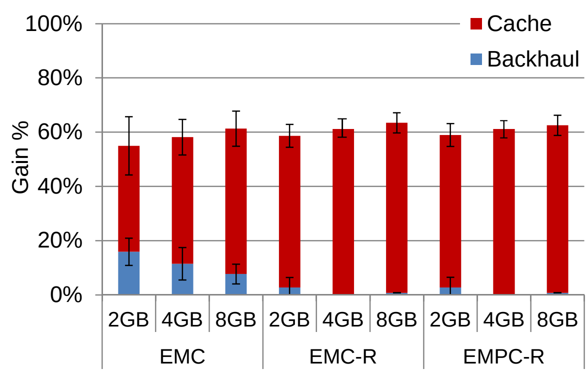


Figure 5.14: Performance against different levels of cache storage supply.

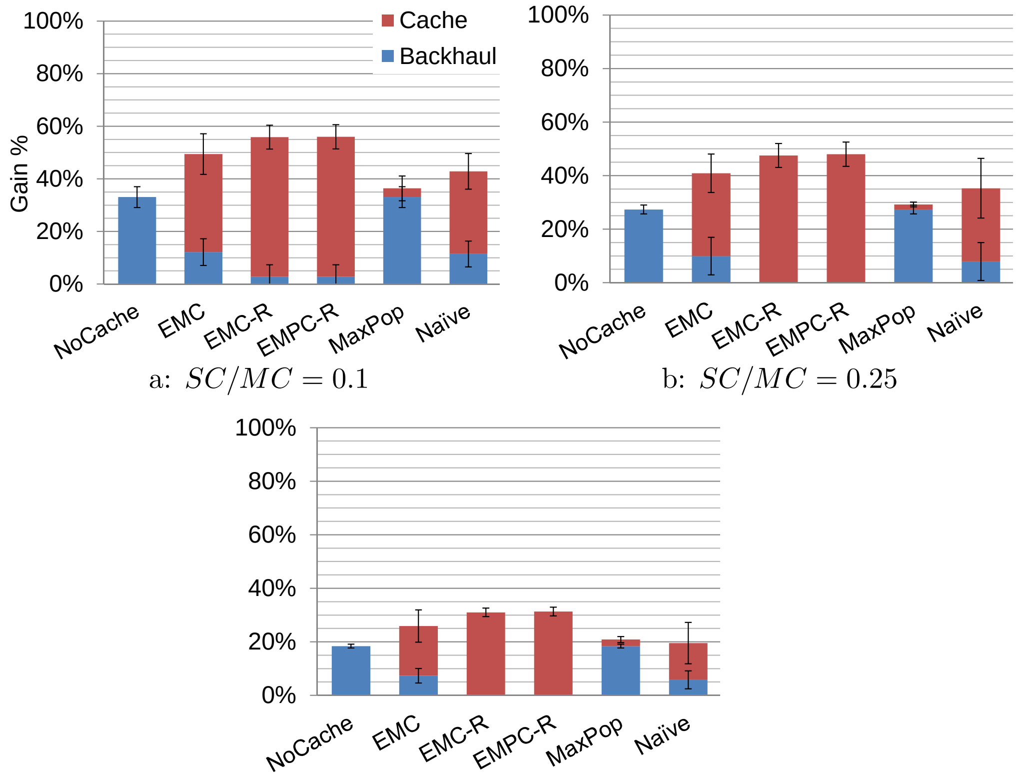
1) *Cache gains increase with storage size:* Doubling the storage buffer from to 4 GB increases EMC cache gains by 8% and EMC-R/EMPC-R by 5%. Further doubling to 8 GB adds up 7% to EMC cache gains and only 2% to EMC-R/EMPC-R.

2) *Declining backhaul gains:* Backhaul gains for EMC drop from 16% to 11%, and from 11% to 8%. Likewise, any backhaul gains (∼2.5%) that exist with the 2 GB scenario for EMC-R/EMPC-R cease to exist because increasing storage improves cache hits, thus decreasing the need for utilising the backhaul to cover up for cache misses.

Delay gains

We complete our evaluation by studying our model's performance from a download *delay* perspective. We adapt the same simulation model explained on page 99 with the difference of assuming transfer costs which reflect users' cost valuation of download delay, and try to evaluate our approach under the incremental presence of CDN caching which reduces the delay cost of the top-most popular videos. Despite its growing scale in user numbers, demand and corresponding traffic, the Internet manages to offer sufficient QoS levels thanks to CDNs. However, to our knowledge the current state of the art does not consider the possible implications caused by CDN content and service replication on local caching performance. Opposite to that, we try to study the impact of a CDN on local caching in support of our discussion in Section 5.1.3, where we argue for complementing CDNs by targeting niche requests locally.

Figure 5.15 on the next page shows gains for 3 different small-cellular over macro-cellular delay cost ratios SC/MC, each of which can reflect how users assess, e.g. based on QoE measurements, the benefit of using a small cell relative to pure macro-cellular usage: (i) low ratio, high benefit (Graph 5.15a); (ii) average ratio and benefit (Graph 5.15b); and (iii) hight ratio, small benefit (Graph 5.15c).



a: $SC/MC = 0.1$ b: $SC/MC = 0.25$

c: $SC/MC = 0.5$

Figure 5.15: Performance for different small-cellular over macro-cellular transfer cost ratios (SC/MC).

As a general comment, all performances, either with respect to cache hits or the backhaul, drop with increasing SC/MC. Also, the gain differences between the different cache models also tend to decrease. This is normal to expect as the corresponding gains from caching also drop from 90% to 75% and 50% for the presented cost ratios in the figure respectively. As with other graphs, we observe that EMC-R and EMPC-R share the same performances, which are higher than EMC and any other benchmark. EMC in particular, has a robust gain performance relative to the extended model versions. Its corresponding cache gain difference from the extended models grows smaller from ∼16% to 12% with increasing cost ratios. Moreover, this performance difference is significantly lower when considering also the gains from the backhaul: ∼5-6.5%. As we also highlight on page 100, this is important due to the computational overhead of the cache replacements extension.

In addition, we observe that the cache gains from MaxPop are robust but low, and that all the caching models have benefits over NoCaching. For instance, cache gains from EMC are higher than from NoCache; if backhaul is further included, then the added gains from EMC are 14%, 8% and 7% for the portrayed cost ratios respectively. Last, the further gains from EMC relative to Naïve are fair and robust: ∼5-7% (resp., ∼5.5-7.5% with backhaul).

The impact of CDNs delay gains

Next, the results of Figure 5.16 correspond to a scenario of $SC/MC = 0.1$, where CDN presence reduces the transfer delay costs of the top-most popular videos (CDN%) that it serves by 75%. Notice that cache gains for EMC-R/EMPC-R are quite *robust*

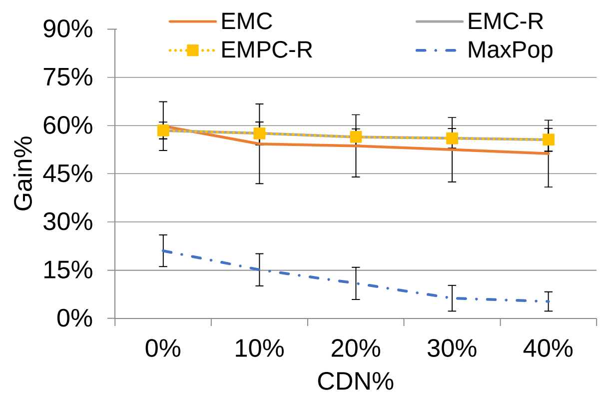


Figure 5.16: Cache gains against the percentage (CDN%) of the topmost popular videos cached and served by CDNs.

and that the performance of EMC for CDN%>10% experiences only a small gradual drop. This is due to the cache replacements extension which identifies what content is mostly beneficial to preserve in the cache, while in the case of EMC, due to its ability to adapt well to temporal locality via Q^l_s.

Contrarily, *popularity*-based caching appears to have a significant *loss*. Its gains in general decrease with CDN%, which is intuitively expected due to the reduced transfer costs offered by the CDN level for the topmost IOs which MaxPop chooses to cache. With merely 20% of the top-most popular videos being addressed by the CDN level, the performance of MaxPop drops below 15% and converges to only ~6% for CDN%>30%, whereas the corresponding performances of EMPC-R/EMC-R and EMC are ~56% and 52.5%.

5.5 Implementation[d] & complexity analysis

Our model employs a distributed design which can be incarnated via Cache Decision Modules (CDMs) running in small cell BSs of HWNs such as portrayed in Figure 5.1 on page 87. CDMs are responsible for orchestrating cache actions after receiving and processing information about mobiles' requests, and by maintaining a distributed state via exchanging control messages within their neighbourhood. In what follows, we present 4 basic CDM functions, namely, (i) user requests, (ii) mobility prediction, (iii) content

[d] The reader may also refer to Section 4.5 on page 62 for a detailed discussion on integrating our mobility-based proactive caching models to the network layer of ICN architectures, with direct references to the cases of the PSI and NDN ICN architectures.

popularity adaptation and (iv) cache replacements, from an implementation perspective along with a corresponding analysis of the implied computational, memory and intercommunication complexities costs. The latter two functions are applied only when adapting the corresponding model extensions, hence they imply a further implementation and running complexity compared to the basic model. Apart from the analytical approach which follows next, note that in practice the implied costs depend on the level of integration in the network "stack". CDMs can *best integrate* with the network layer of ICNs to directly exploit named requests and other network primitives [74], particularly multicast communication from/to publisher/subscribed mobiles and CDMs; albeit the solution can be also implemented as an application over standard IP. Next, we discuss the implementation details and the corresponding complexity analysis using the notation of Table 5.6 on the next page.

5.5.1 User requests

Users and content must be identified by an ICN name, URL or IP address, used for proactively fetching the objects via the backhaul network to the caches, as well as indices for maintaining a required state (see Sections 5.5.2–5.5.6) for cache requests. While IP addresses have a fixed size, URLs or ICN names can be arbitrarily long, unless using a corresponding hash value such as a 20 byte-long (160 bits) SHA-1 cryptographic hash. The messaging cost for submitting such data to each neighbour of source i via unicast messages is $O(|M_i| \cdot |N_i|)$. However, it can be significantly reduced to $O(|M_i|)$ via multicasting each mobile's requests to all members of N_i. In any case, the corresponding memory cost in a target cell l depends on $|\hat{M}_l|$ and $\bar{r}$, i.e. it is $O(|\hat{M}_l| \cdot \bar{r})$.

5.5.2 Mobility prediction

CDMs must have good and timely information about the mobiles' transition probabilities, handover and small cell residence/handover *duration* times to make accurate mobility predictions and correspondingly accurate cache decisions. Both probabilities and time durations can be estimated either based on a local history between *neighbouring* small cells, or they can be *predicted* by external, mobility tracking mechanisms. Within this content, we identify 3 alternative paradigms analysed next.

Centrally coordinated prediction

A neighbourhood of small cells includes all the possible next-hop transition APs that mobiles can attach after leaving their origins small cell. Neighbourhoods adapt a static configuration and cooperate with the a macro cell which can track user mobility within its coverage.

Decentralised prediction

A fully distributed and decentralised approach involves the mobiles themselves notifying their possible destination cache point cells about their origins cache point cell upon completing their handover. This allowing for a lightweight distributed neighbourhood discovery and probability estimation.

External mechanisms

Cache point attachment probabilities can be attained by *external mechanisms* such as the ones described in [208, 209], or through mobility prediction [210, 211] and tracking [212, 213]. Mobile transition probabilities can be attained by external mechanisms such as [208], or through mobility prediction [211] and mobility tracking [213] mechanisms. These approaches have been extensively explored in the literature using diverse mobility scenarios which include driving and roaming habits of cell phone users, or students' mobility in campuses. Extracting user-specific mobility patterns is feasible in such scenarios with the cooperation of mobile providers or simply by using common navigation and web mapping services such as Google Maps, to which users can explicitly declare their intended route and destination.

Table 5.6: Notation used for complexity analysis.

Notation	Description
l:	A handover destination (target) small cell.
i:	Origins (source of handover) small cell.
CDM_x:	The CDM running at small cell x.
s or s_c:	A requested content or the c-th chunk of the content.
M_i:	A set of mobiles attached to some origins cell i.
N_i:	i's neighbourhood, i.e. set of next hop destinations from i.
n:	The number of objects (chunks of IOs or whole IOs, depending on the assumed scenario) currently in a local cache.
I_l:	The set of the different origins cells from where mobiles can move to l.
$\hat{M}_l$:	The set of all mobiles that can move to l.
$\bar{r}$:	The average number of content requests per mobile.
$\bar{r_c}$:	The average number of requested chunks per mobile request r.

5.5.3 Memory & messaging

Complexity costs are low when using navigation services or with centrally coordinated prediction (e.g., due to the wireless MAC layer multicast ability of the macro BS). Decentralised prediction on the other requires some communication, computations and state maintenance in neighbouring CDMs: Assume a target l for mobiles requesting s, each of which corresponds to its own origins i. CDM_l maintains a small state per each i for tracking the number of incoming handovers $h_{i,l}$ from i and the total number of outgoing handovers H_i from i. The former are used to predict the *probability* of future mobile transitions from each i to l as defined in (5.14). Likewise, a history of handover and residence *duration* times from past handovers per each i can be kept. We refer collectively to all the former as *"mobility prediction information"*.

$$q^{i,l} = h_{i,l}/H_i. \tag{5.14}$$

Noteworthy, all mobility prediction information imply *lightweight* messaging and memory costs. $h_{i,l}$ and H_i can be retrieved either from incoming mobiles from i or from the mobiles' requests submitted to l. Either way, there is no extra messaging implied as the needed information can be retrieved by l via piggy-backing 2 bytes for $h_{i,l}$ and another 2 bytes for H_i in existing communication for either the mobiles' attachment to l or for the requests submission to l (Section 5.5.1) respectively. Though small, this amount of control data is reasonably big enough to correspond to the locally maintained counters in memory which do not have to be bigger than 2 bytes as they must be periodically reset or readapted (e.g. with exponential smoothing) to reflect the currentness of the predicted transition probabilities. Given that $|I_l|$ counters are needed to maintain $h_{i,l}$ and another $|I_l|$ to maintain H_i for each i, the memory cost for transition probabilities in each CDM_l is $O(|I_l|)$. The same messaging and memory complexities, only with respect to 4–byte[39] long counters, apply to handover and residence duration times information, which can be maintained after past mobile transitions and residence sessions. Mobility prediction can be more *accurate* if the mobiles update l about their status (e.g. GPS coordinates) while in transit to l. This comes though with a tradeoff in terms of messaging, $O(U_f \cdot |\hat{M}_l|)$, where U_f is the average frequency of updates per mobile sent to l.

5.5.4 Cache decisions

CDM_l can take cache actions with each incoming request for some content s by integrating the predicted $q_s^{i,l} \equiv q^{i,l}$ from all the mobiles with an active request for s, along the lines of formula (5.2) as $Q_s^l = \sum_{\forall i} h_{i,l}/H_i$. By further leveraging the knowledge of macro/small-cellular wireless and backhaul throughputs, as well as the predictions about the handover and residence times for mobiles, CDM_l can take more *accurate* and cache-*efficient* actions at the level of separate chunks s_c. The reason lies in the fact that different mobiles can consume different chunks of the same s from l, due to the different number of: (i) chunks κ that each individual mobile will already have consumed from the macro link before connecting to l; (ii) chunks λ that CDM_l will to be able to prefetch and cache via its backhaul while the mobile is in transit; and (iii) cached chunks ν that the mobile will be able to consume during its residence in l

[39] Assuming Unix time for simplicity. It can be further reduced as it is used for time differences.

via its wireless interface. The earlier two are based on information about the mobile's predicted handover duration and the latter based on information about the mobile's residence duration. Hence, the chunks predicted as *consumable* from l and for which CDM_l must decide, are those en $[s_{\kappa+1}, s_{\kappa+\rho}]$, where $\rho = min(\lambda, \nu)$. Accordingly, we adapt the predicted mobile transition information, this time per chunk, as $q^{i,l}_{s_c} \equiv q^{i,l}$ in formula (5.2), i.e. $Q^l_{s_c} = \sum_{\forall i} q^{i,l}_{s_c} = \sum_{\forall i} h_{i,l}/H_i$.

Based on all above, the practical cost of cache actions at l is subject to the granularity of requests. The computational complexity of each cache decision is $O(1)$ according to rule (5.1) and there are $\bar{r} \cdot \hat{M}_l$ decisions to make by CDM_l. However, if decisions consider chunks rather than whole objects, this rises to $\bar{r} \cdot \bar{r_c} \cdot \hat{M}_l$. Note that the former hold for final and irrevocable cache decisions taken upon receiving a mobile's request(s). We discuss the computational complexity cost of cache decisions again when analysing the impact of model extensions.

5.5.5 Content Popularity adaptation

The popularity extension (see notation and definitions on page 84) implies no extra messaging costs, again $O(1)$ computations per request with an added small burden for computing $f^l_s = T^l_{req}/I^l_s$ on the fly, and a low memory cost which we analyse next: The memory cost of T^l_{req} is constrained to only a 4-byte long counter for keeping the time difference between the latest two consecutive requests. For I^l_s, we need to have $O(|S|)$ of such 4-byte long counters, where $S = \cup\{s\}$ is the set of all the different objects requested to be cached by l. Each of these counters has to be mapped to the corresponding content or chunk names which –as mentioned above– can be 20-byte long hash values. Assuming an example of 100 requests for different contents, the total memory requirement is merely $(20 + 4) \cdot 100 = 2400$ bytes, i.e. less than 2.5 KB. If these contents correspond to big objects, e.g. videos, which are split to ξ chunks, then the former requirement raises by a factor of ξ, e.g. less than a quarter of a megabyte for $\xi = 100$. Finally, maintaining w requires only 4 bytes for keeping the average handover duration from any source cell to l, and another 4 bytes, at most, for the number of requests currently served with data via l's cache or backhaul.

5.5.6 Cache Replacements

Extending our model with cache replacements has no impact on messaging; yet it does have an *impact* on the *computational cost* of new object insertions to the local cache, which is dominated by the cost of maintaining a gain-based ordering state used for cache replacement decisions (Section 5.2.2). This gain-based ordering can be implemented with a binary heap. If the insertion algorithm originally introduced for heapsort [214] by Williams is used, then each new object insertion to the cache implies in worst case an $O(log(n))$ time complexity cost to maintain the corresponding state for gains. Consequently, the insertion of n objects in the cache runs in worst case in $O(n \cdot log(n))$. Nevertheless, a faster method [215] by Floyd that involves repeatedly merging small heaps to form bigger ones requires only $(2 + o(1))n$ actions in the worst case, hence the corresponding complexity cost can be significantly lowered to $O(n)$. This is particularly important because apart from reducing the running time cost for inserting new objects to the cache, it keeps the cost lower for updating the gains of the already cached objects in case their expected gains according to (5.4) or (5.5) change in correspondence to changes in temporal locality, as expressed via newer short timescale measurements on both mobility prediction and requested content popularity. If this is the case, then –from a practical point of view– the added burden for keeping the gain-ordering state up-to-date with temporal locality depends on the frequency of updating (extracting, reassessing and reinserting) the gains in the heap. To reduce the burden, updates may take place with a lower frequency, e.g. per k incoming mobile requests. Nonetheless, this should come at a cost on performance, marking a trade-off between lowering the burden of updates on expected gains and increasing the actually achieved gains from prospective cache hits. Last, the (practical) impact on memory and running time can be larger if n in the above analysis refers to a number of IO chunks rather than to a number of IOs as a whole.

5.6 Conclusions

This chapter presented our novel Efficient Mobility-based Caching (EMC) distributed model along with Content Popularity and Legacy Caching model extensions. Our solution has significant design advantages over other proactive approaches, the most important of which lie in its ability (i) to address *niche* mobile demand, (ii) to dynamically *tune* the contribution of mobile requests' popularity and users' mobility prediction on cache actions, and (iii) to take on-the-fly cache decisions based on contemporary, short timescale local mobility information. By design, our approach targets less popular or personalised content that is unaddressed by other proactive approaches in literature and by CDNs, both of which are designed to serve popular content. Our model can be applied to heterogeneous wireless network environments in order to yield monetary and delay cost gains for users, along with possible positive implications on QoE.

Regarding the performance of our solution, among our most notable findings, we show that gains are good from mobility prediction against the cases of applying no, popularity-only or naïve local proactive caching in scenarios that combine different caching demand levels, video catalogues and mobility models, among other system parameter combinations. Cache decisions based only on mobility prediction appear to adapt well to temporal locality due to using short timescale information which allow to capture changes in temporal locality. This outcome on mobility prediction largely coincides with the conclusions in [54] which relate prediction accuracy to the latest mobility information in urban environments due to the higher road network complexity, traffic congestion and the variety of mobility habits and routes.

Extending our model with cache replacements can (even substantially) improve performance in terms of average gains and robustness against system parameters, yet at the cost of a computational overhead. However, our basic mobility-based model already yields an overall good performance which in certain scenarios can be very close to its extended counterparts, especially when including backhaul gains. This observation is important given the added complexity of cache replacements. In addition, we observe that the performance of mobility-based caching appears to improve with the level of popularity skewness, approaching close to the high and robust gains of the extended model with cache replacements. Also, we point out that mobility-based proactive caching yields more benefits than popularity-based caching in scenarios with low demand for mobility-based proactive caching and a large video catalogue.

Finally, unlike most of the other work in the literature which neglects the impact of CDN caching on the performance of local solutions, we try to roughly approach the aforementioned impact with respect to delay cost gains yielded by local proactive caching solutions. Our conclusion is that our basic and extended models can have more robust gains than proactive approaches targeted on popular content.

Chapter 6

Global Name Resolution in Information-Centric Networking

This chapter is dedicated to our conducted research study towards a *feasible* and globally *scalable* Information-Centric Networking Name Resolution System (ICN NRS). It is based on our published work in [161, 89, 216, 217] and corresponds to our thesis point ((3)) on page 12 and to our contribution point ((4)) on page 14. The reader is provided with the necessary background on ICN in Chapter 2 and the related literature review on Name Resolution Systems Section 3.2. The goal of the chapter is to study global name resolution in ICN in seek for a NRS that can be *feasible* in a sense that it combines good scalability and policy compliance features. To this direction, we contribute to certain design aspects as well as to the performance evaluation of the novel lookup-by-name design of the Distributed Hash Table-based Name Resolution System (DHT-NRS) proposed for the PSI ICN architecture. Moreover, we try to cover a gap between intuition and reality on the scalability properties of the DONA and CURLING ICN NRSs that exists in the literature by engaging into a meticulous performance evaluation study. The presented performance evaluation results are based on full-scale real inter-domain topology traces as well as scaled-down synthetic topologies, in an effort to quantify a series of important scalability and policy compliance performance aspects, including the distribution of name-resolution state across the Internet topology, the associated processing and signalling overheads, routing policy violations of name resolution requests and the implied level of path stretching. As a final remark, we note that the presented system design aspects in DHT-NRS as well as all of the presented performance evaluation results are the product of a close collaboration with the other authors in our aforementioned publications. For more on our contributed simulation software, please refer to on page 17.

6.1 Chapter outline

In what follows, Section 6.2 engages into a deeper technical *background* discussion in extension of the fundamental concepts on ICN name resolution that are previously outlined in Section 2.2.2 on page 21 of Chapter 2. Then, the section proceeds with our *motivation* on page 121, based on the preceding background analysis.

In Section 6.3.1, we first discuss the adaptation of *multihoming* and *peering* AS relations to the overlay routing scheme of H-Pastry, along with the rest of the design aspects of H-Pastry. Moreover, the section provides a head-to-head simulation-based performance comparison between the DHT-NRS running on top of H-Pastry and DONA on page 125. The results highlight various important system performance aspects and intrinsic properties of DHT-NRS due to its underlying DHT nature, which arguably bring into question the feasibleness of adapting DHT-NRS on a global scale, mainly, as a result of its implied routing policy violations.

In Section 6.4 we turn our interest towards investigating the possibility of adapting a route-by-name approach as a global ICN NRS, triggered by the conclusions on DHT-NRS and the preliminary evaluation results of DONA over scaled-down topologies. For that, we focus on the design of CURLING and present a detailed feasibility study against the performance of DONA on page 133 based on a thorough quantification of important aspects over a full-scale inter-AS Internet topology. Finally, we close this section and with it this chapter on page 141 by discussing the potential role of *cloud computing* for adapting a global- scale route-by-name name resolution design at the cost of a bearable added routing stretch. The concept of this discussion is that cloud facilities have the ability to offer large volumes of resources on demand, hence they can help to address any scalability and load balancing issues even in the case of small ASs with limited CapEx/OpEx capabilities .

6.2 Background and motivation

6.2.1 Background

Name resolution constitutes the basis of all other functionalities in the ICN architectural paradigm. It enables to resolve names into corresponding locations, to cache IOs closer to consumption, to adapt certain routing and security features, and so forth. Consequently, a *global* ICN NRS is a prerequisite for making the ICN paradigm appropriate to adapt as the Future Internet architecture. However, the closest existing equivalent to a global NRS, the DNS, is far from being adequate with respect to ICN requirements, as discussed in Section 3.2.1 on page 40, and despite the significant body of work that can be found in the literature, global name resolution in ICN *remains* unfortunately *open.*

The design of a global ICN NRS must adhere to some fundamental properties outlined in Table 6.1 on the next page. Property (1) is essentially a prerequisite, while the rest of the properties serve the necessary economic and performance efficiency purposes

Table 6.1: Fundamental properties of a feasible global-scale ICN NRS.

(1)	Must be able to *guarantee* resolution.
(2)	Resolution traffic must imply *minimal* or *no* routing *policy violations*.
(3)	Must imply a balanced distribution of the resolution *state* across the domains of the inter-AS topology.
(4)	Must imply a *scalable control messaging* burden both for control messages and resolution requests.
(5)	Any assumptions made, must comply at least to the current reality and future trends with respect to technological capabilities, network topology, user demand, security, trust and business interests of ISPs, etc.

of an appropriate global-scale ICN NRS. Regarding property (2), the resolution traffic will be significantly higher than today in an ICN Future Internet, thus an ICN NRS must imply minimal or (ideally) no routing policy violations. Policy violations cause *stretched* paths in comparison to the shortest BGP-compliant paths and *"valley" routing* [165, 27], i.e. situations in which an AS acts as a transit for traffic that is not destined to its customer(s) on behalf of its provider or peering ASs.

Regarding both property (3) and property (4), the current number of IOs already implies a huge resolution state size (Section 2.2.1 on page 19) and a corresponding control messaging exchange burden that is difficult to *distribute* in a *fair* and *balanced* way suiting the distinctive roles of customer, provider or sibling AS-relations, as well as the capabilities and size of the different ASs such as stub/access network on the one extreme, and Tier-1 ISPs on the other. Besides current conditions, resource requirements can be particularly higher within the context of ICN if the adapted naming scheme does not support aggregation such as in the case of flat names. In fact, most ICN architectures adapt flat, self-certifying names due to their better support for persistence and authentication [72, 65]. The remaining approaches like [218] in the case of NDN adapt a purely hierarchical name space in order to leverage from the potential benefits of prefix-based name aggregation and caching, a choice which is nonetheless debatable within the ICN context as we discuss in Section 2.2.1 on page 19.

Last, the purpose of property (5) is intentionally generic. It simply states that any design proposed should be based on realistic assumptions in order to be feasible to adapt. For instance, it is unrealistic to assume that all domains in global adaptation scenarios lie in a purely hierarchical, tree-like topology structure. Likewise, is to assume that resolution delay is tolerable, or that all domains have the same capabilities messages, or that big content providers, e.g. YouTube or Facebook, lie only at a particular tier of the inter-AS hierarchy, or that sibling domains are necessarily willing (resp., not willing) to share any type of information with each other, and so forth.

Despite being a necessity, trying to jointly integrate all of the properties together in a single design is a very difficult and multidimensional task, as discussed above, with trade-offs arising between the desired properties, as we discuss next. This reality has triggered the design of a series of overlay *lookup-by-name* and *route-by-name* (defined

Table 6.2: Summary of strengths and weaknesses of lookup-by-name and route-by-name NRS approaches, as identified in the background literature. Obliquely striked-through numbers refer to properties from Table 6.1 that fail to be met. Lookup-by-name designs fail to satisfy properties (2) and (5), whereas route-by-name prerequisite properties (3) and (4).

	Strengths	Weaknesses
Lookup-by-name	✓ Scalable with respect to both system size and state size.	• ~~(2)~~ Routing policy (BGP) violations even by increasing node degree and corresponding stretched-paths compared to policy-compliant, shortest-path routing [159, 27]. • ~~(5)~~ Increased delays due to stretched paths. • ~~(5)~~ Business relations, security and trust issues due to limited or no control over state placement. • ~~(5)~~ Applying Canon [108]: (i) increases DHT design complexity; (ii) does not address sibling relations; (iii) adds up to criticism on matters of business relations, security and trust interest.
Route-by-name	✓ Adapt fully to the underlying inter-AS topology. ✓ Fully adhere to the established BGP routing policies. ✓ Imply zero path stretch.	• ~~(3)~~,~~(4)~~ Intuitively evident scalability concerns in global adaptation scenarios: (i) extensive state replication; (ii) load imbalances, particularly at the higher levels of the inter-network topology.

in Section 2.2.2 on page 21) ICNNRSs. Overlays have the advantage of being ready to run over the already established Internet infrastructure and to leverage the more easily available computing resources of the application layer, in contrast to the more restricted ones of the network layer such as in the case of the NDN [219] architecture. However, lookup-by-name as well as route-by-name design approaches intuitively suffer from a fundamental *tradeoff* between *routing efficiency* and state *load balance*, as briefed in Table 6.2.

Lookup-by-name approaches are typically built upon some DHT overlay routing mechanism. DHTs have the advantage of *scaling* with respect to to system size, as discovery procedures can be performed in logarithmic time, as well as with respect to state size by balancing the distribution of the resolution state amongst nodes irrespective of the structure of the inter-domain topology. Only this comes at the cost of (i) highly stretched, suboptimal name-resolution paths [159], also by (ii) violating the established BGP policies [27] ("valley routing", as mentioned above), (iii) the need to update the DHT whenever information moves and, finally, (iv) having little or no control over the placement of the resolution state as, e.g., the RVP responsible for a particular IO's resolution may actually reside outside the domain boundaries of the IO's publisher. To make things worse, DHTs may still not adhere to routing policies even by increasing the average node degree as in [159].

In addition, the design of DHT-based NRSs can be too complex, raise business and/or trust issues, or be superficial. Inevitably, any DHT-based NRSs has to modify or extend the “flat” routing schemes of the original DHT designs in order to adapt the recursive hierarchy of provider-to-customer relations between the ASs of ISPs. Only adapting to the hierarchy is not as trivial as in the case of hierarchical Chord [108] and can have implications: First, because the Canon paradigm expects explicitly from different ASs with direct business relations to share crucial information regarding their own internal organisation. Only business relations between ASs are not an equivalent to trust relations, not mentioning the fact that cooperating ISPs may well be still business competitors or have conflicting interests harmed by exposing their internal organisation. Second, because the underlying topology in the Internet is not as trivial as a pure a multi-tier hierarchy of interconnected ASs with transit customer-provider links. Such an assumption as in the case of [163] is not just an oversimplification; it is unrealistic. Therefore, *sibling relations* between ASs and *multihoming* with higher level ASs have to be adapted in the hierarchical DHT without violating the Canon paradigm. This aspect is highly important as, clearly [62, 220, 221], the Internet topology is constantly increasing and evolving into a *mesh* dominated by such inter-AS relations. Unlike that, much of the work in the literature, e.g. [67], (only) superficially approaches multihoming based on indirection with the use of binding schemes.

Route-by-name NRSs, on the other, naturally adapt to the underlying inter-AS topological structure, including multihoming and peering relations. Accordingly, they *fully adhere* to the established *BGP* routing policies and, consequently, imply *no path stretch*, unlike their DHT-based lookup-by-name counterpart designs. Nonetheless, the same significant works in the literature that propose route-by-name approaches admit (based on proof of sketch concepts as in [72]) that they need to pay a *high cost* in terms of extensive *(i) state replication* and *(ii) load imbalances* regarding control messaging resolution and state distribution across the inter-AS topology, particularly at the higher levels of the underlying inter-AS topological structure.

6.2.2 Motivation

The former background describes a probable dead-end towards the design of a feasible ICN NRS. As summarised in Table 6.2, on the one hand lookup-by-name designs fail to satisfy prerequisite property (2) from Table 6.1 on page 119, whereas on the other hand route-by-name prerequisite properties (3) and (4) from the same table. Nevertheless, a closer look at the state-of-the-art in the literature discussed in Section 3.2 on page 40 shows that most performance and/or scalability conclusions drawn about the proposed ICN NRS approaches are based on (i) *rough*, (ii) *intuitive*, (iii) *unrealistic* or (iv) *oversimplified* assumptions.

The former applies also to the two most representative examples from the category of route-by-name designs, namely, DONA and CURLING. The of authors DONA engage into a discussion on its feasibility based on empirical estimations, abstractions and assumptions from older work in the literature regarding the underlying Internet topology and user requests demand. For instance, they assume that the number of registered IOs is in the scale of 10^{11} while some more recent credible estimations [4, 6] raise this number to 10^{13} and even 10^{16} [67]. This is at least 2 and up to 5 orders of magnitude greater than the authors' expectations. What is more, is that DONA lacks a detailed, quantification analysis with respect to the impact of the underlay topology and the popularity characteristics of IO requests on the distribution of the resolution state and messaging. Instead, the authors merely refer to the high load of Tier-1 ISPs, which by intuition is easy to guess.

Likewise to [72], the work of [73] provides only a discussion on the scalability properties of CURLING, according to which system scalability depends on the amount and popularity of IOs in each resolution node with the most vulnerable nodes being the ones which maintain the highest number of popular IO resolution entries, rather than the intuition that the highest burden will be suffered by Tier-1 ASs. This conclusion contradicts the proof of sketch analysis of [72] (despite the evident architectural similarities with DONA) based on arguments regarding content manipulation operations, i.e. inter-ISP business relationships, local ISP policies and specific content provider and customer preferences expressed via the special scoping and filtering functions of CURLING. Indeed, it can be easily perceived that these operations will work as natural load distribution mechanisms. However, the extend to which this is going to happen combined to the extra messaging burden cost that needs to be paid due to forwarding requests rather than exchanging state between sibling ASs as DONA does, remains an open question.

Regarding lookup-by-name approaches, most of the work in the literature uses oversimplified *tree* topologies. For instance, MDHT [67] uses a full k-ary tree without multihoming and peering links taken into consideration, while the work of [163] uses a tree overlay network that is composed out of a random selection of nodes in an inter-domain topology that has merely two levels. In another example, the authors of [90] proceed into a performance evaluation study based on observations made in different contexts and significant abstractions regarding intra-domain routing overhead and caching. And in one last example, the work of [164] neglects the routing overheads.

For all that, our conclusion is that the *impact* of the topological features of the *underlay network*, i.e. multihoming, sibling and peering inter-AS relations, is *not adequately studied*. On the one hand, it could cause great state and messaging load imbalances beyond just the level of Tier-1 ISPs in the case of route-by-name schemes such as by affecting even the smallest of ISPs with the fewest available resources. On the other hand, it could push policy violations, path stretch and a corresponding resolution delay to intolerable levels for ICN standards as argued on page 119. All above provide us with the right motivation for:

(i) Exploring the potential of a novel DHT-based NRS that can preserve the general scalability and load balancing properties of DHTs, while at the same time reduce path-stretch and routing policy violations.

(ii) Revisiting the DONA and CURLING route-by-name schemes in order to *quantify* the speculated load imbalances and overheads in global adaptation scenarios.

(iii) Discussing the potential role of *cloud computing* with respect to adapting a global-scale route-by-name NRS in ICN at the cost of a bearable added routing stretch and reasonable CapEx/OpEx investments that suit the role and capabilities of ASs.

6.3 Lookup-by-Name schemes

6.3.1 Adapting Multihoming & Peering Autonomous System relations to Hierarchical Pastry

Hierarchical Pastry (H-Pastry) [161] is a multi-level DHT scheme based on applying the *Canon* paradigm to the routing model of Pastry. The fundamentals of both Canon and Pastry are given in Section 3.2.2 on page 41. Due to Canon, H-Pastry adapts fully to the underlying hierarchical inter-domain topology based on the progressive merging of the individual DHT constructions that correspond to domains, while the choice of Pastry provides inherent advantages regarding the consideration of network proximity during the formation of the overlay, without incurring any node selection/grouping overhead as is other cases, for instance in hierarchical Chord (a.k.a. Crescendo) [108]). As a result, the routing model in H-Pastry can support inter-domain multihoming and peering relationships. Moreover, it can yield shorter overlay paths [161, 222] compared to its design ancestor Pastry, which justifies why it has been adapted as the look-up routing substrate in the design on DHT-NRS [89].

In order to adapt to the multi-level structure of an inter-network, H-Pastry employs a corresponding structure to partition the information identifier space, and the corresponding routing state. Distinct `Routing Tables`[40] identical to those of regular Pastry are maintained for each level of the inter-domain topology. The participating nodes at each level maintain information in these `Routing Tables` about other nodes that are numerically closer to certain points in the identifier space than any other node at that particular level. In this manner, the participating H-Pastry nodes recursively create levels, a.k.a. *"rings"*, that adapt to the AS-level topology of the network, as described in the following subsection.

Ring formation: state exchange with other levels

Assume a domain X that lies at level l of the inter-domain hierarchy and one of its nodes x which is already a member of the ring of X. Node x needs to exchange state with the overlay nodes that reside at the descendant domains of X. Next, x exchanges its current state with the overlay nodes at the ancestor domain X_a of X that is at level $l-1$ of the hierarchy. Moreover, it does the same thing with the nodes in the sibling –via X_a–

[40] The terms "`Routing Table`" and "`Leaf Set`" are the same as the ones introduced in Pastry's flat DHT scheme. The reader may refer to Section 3.2.2 on page 41 for a formal definition of both.

domains of X, i.e. with nodes in the descendant domains of X_a, of course excluding X itself and its own descendant domains. This short series of steps for exchanging routing state is recursively repeated with overlay nodes at the ancestor domain(s) of X at level $l-2$ and so forth.

Multihoming relations

As described in the previous subsection, the ring formation procedure is already compatible with multihoming. A domain delivers its traffic to non-descendant or peering domains via more than one higher level domains. Therefore, in case X is multihomed to, e.g. $X_a^1, X_a^2, \dots X_a^n$, the state exchange proceeds with each of the former domains X_a^i and the nodes of the sibling –via X_a^i– domains of X. Multihoming is common in the Internet topology because it provides more efficient routing, fault resilience and load balancing between ISPs. And likewise are peering agreements, due to which traffic is exchanged between peers instead of traversing higher level domains. This direct traffic exchange is desirable in the case of overlay routing as well, thus the traffic between peering domain nodes must *not* be transferred via other domains, i.e. as it is done with respect to intra-domain paths according to the Canon paradigm. However, both types of inter-domain relationships insert circles in the otherwise tree-like structure of the inter-AS topology, which *poses limits* to the *applicability* of the *Canon* paradigm.

Peering relations

To address the *issue of cycles* and manage to apply the Canon paradigm in its entireness, we follow an approach inspired by the work of [107]: we connect the peering domains the same (virtual) node (thus making them multihomed) in order to facilitate the DHT formation process which we described above. This node is only *virtually* introduced one level above the peering domains as shown in Figure 6.1, thus it does *not* correspond to any actual network entity.

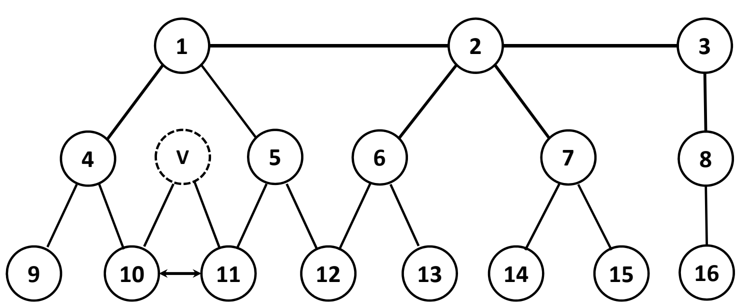


Figure 6.1: Example inter-domain topology: each node represents a distinct administrative domain. Domain 12 is *multihomed* and a *peering* relationship is established between domains 10 and 11.

Total state per node

In overall, the *total state* maintained in an overlay node x (assuming node x from above) is equal to $l+2$ pairs of a `Leaf Set` and a `Routing Table`: (i) a pair of a `Leaf Set` and a `Routing Table` for the intra-domain (regular Pastry) ring, which is referred to as the "level $l+1$ routing state"; (ii) a pair of a `Leaf Set` and a `Routing Table` for the nodes at the descendant domains of X, which is referred to as the "level l routing state'; (iii) l `Leaf Set` and `Routing Table` pairs for Pastry state about nodes at the l levels of ancestor domains of X and their corresponding customer domains, referred to as the "level l-1, l-2, ..., 0 routing state" (in accordance to the former).

Essential performance aspects

As presented in the performance evaluation and design analysis of [161], H-Pastry exhibits some very good performance and resilience properties:

- It can *lower path stretch* by 55% and 47% compared to Chord[41] [71] and its hierarchical version Crescendo [108] respectively, while also
- managing to *keep traffic within administrative domains*, hence resulting in 27% less inter-domain hops and 55% shorter intra-domain paths than flat Pastry.
- H-Pastry also *reduces valley* overlay routing *violations* per path by 56%, 31% and 36% compared to Chord, Pastryand hierarchical Chord respectively.
- Last, it exhibits *good fault resilience* properties: its routing fails in the event of a maximum of $H \times |L/2|$ concurrent node failures, whereas the corresponding number in the case of regular Pastry is $|L/2|$. grouped in H sets of $|L/2|$ adjacent IDs each, where H is the height of the domain level hierarchy and L the size of the `Leaf Set`.

6.3.2 Feasibility and Scalability aspects of the Distributed Hash Table-based Name Resolution System

This section discusses the extent to which DHT-NRS can meet with the requirements described in Table 6.1 on page 119. Our focus is put particularly on the exhibited (i) system load and (ii) routing performance aspects of DHT-NRS relative to DONA in order to quantify the inherent pros and drawbacks of DHT-NRS against an architecture that fully adapts to the substrate inter-network structure and fully respects the established routing policies. The important *system load aspects* which we study below refer to the implied memory, signalling and processing overheads, as well as to the level of load imbalances amongst the participating nodes in either of the two NRSs, whereas for *routing performance*, our analysis focuses on *route policy compliance* and the related *path-stretch*.

[41] Recall that a Chord-like overlay routing scheme is used in the ROFL ICN architecture that we briefly review in Section 2.3.3 on page 31.

Table 6.3: Control Messages in DHT-NRS.

Type	Description
`Adv`	*Advertisement*: sent by the publisher(s) of an IO to register the IO with DHT-NRS containing information that maps the ID of the advertised IO to the network location of the issuing publisher.
`Sub`	*Subscription*: issued by a subscriber to request a specific IO after its name, with further information about the network location of the subscriber.
`Ntf`	*Notification*: special notification message sent by an RVP to the publisher(s) of an IO in order to instruct them to start the delivery of the IO data.
`Pub`	*Publication*: Upon notification from the RVP, a publisher sends the IO data to the subscriber(s) using a separate forwarding mechanism.

Basic functionality in DHT-NRS

Name resolution in DHT-NRS follows the Pub-Sub paradigm of Section 2.1 on page 18, where IO publishers advertise the availability of the IOs they wish to share and the prospective IO consumers issue a subscription to declare their interest in some IOs. These publisher and subscriber entities interact via a set of Randezvous Nodes (RNs) which are deployed across the inter-domain topology and which act as special brokers responsible for matching subscriptions to corresponding publications based on the requested IO names. Each RN has a unique ID, with each AS across the Internet topology having at least one RN, while adding further RNs has a positive impact on system scalability. Moreover, a statistically unique ID is created for each IO, while both publishers and subscribers interact with DHT-NRS through their local domain RN(s) which they learn during network attachment. Consequent name-resolution requests take place based on the exchange of special control messages (see Table 6.3) along the lines of the procedure discussed in Section 2.3.1 on page 24. An Information Object Entry (IOE) record is maintained for each unique ID-to-publisher mapping by an RN designated by H-Pastry, which is denoted as the RandezVous Point (RVP) for the ID. Corresponding `Adv` and `Sub` messages are routed towards this node via the H-Pastry routing overlay. Regarding cases of multiple publishers per IOE, the RVP adapts an *anycast* mode of communication after some, e.g., load balancing or topological criteria, and sends a `Ntf` to one of the alternative IO providers. Note that the exact information included in a `Ntf` depends on the underlying forwarding mechanism and that *caching* can exploit the topological properties of H-Pastry to automatically select the (network-wise) closest publisher.

Regarding routing of resolution traffic, DHT-NRS tries to address the scalability concerns of the DHT-based efforts discussed in Section 3.2.3 on page 42 by adapting to the hierarchy of the underlying network topology. To do so, DHT-NRS is built upon the DHT overlay routing substrate of H-Pastry (Section 3.2.2 on page 42) which adheres to the Canon [108] paradigm, thus it considers the physical network proximity, the boundaries of the administrative domains and the corresponding inter-domain routing policies. However, the simulation-based evaluation results presented in Section 6.3.2

Table 6.4: Traffic mix characteristics.

Type	Data plane ratio	Control plane ratio	Object size median	Object Size Distribution	Popularity
Web	35.10%	36.40%	10.386 KB	Lognormal-Pareto	Zipf
P2P	15.85%	2.56x10^{-4}%	650.11 MB	*Sampling*	Mandelbrot-Zipf
Video	19.54%	37.04x10^{-3}%	7.6 MB	*Concatenated* Normal	Weibull
Other	29.51%	63.57%	5 KB	Normal	Zipf

denote that DHT-NRS can *not* fully eliminate some of the intrinsic problems of DHTs, manly with respect to path-stretch and BGP policy violations.

Simulation model

All performance results presented in this section refer to 5 simulation repetitions, using the simulation model that is outlined in Table 6.5 on the next page. In detail, we adapt *scaled-down inter-domain topologies* generated by the algorithm presented in [173]. The size of these topologies is manageable for evaluation purposes while they maintain the same characteristics (i.e., business relationships) as in the measured graphs. Our evaluation employs a 400 domain topology following a hierarchical model with six levels that also contains multi-homing and peering links between the domains. We deploy a population of 4400 RNs for DHT-NRS (resp., RHs for DONA) uniformly across the domains.

Regarding the *workload* used as input, we employ a mixture of various traffic types (e.g., Web, Video, P2P) described in detail in Table 6.4. We choose to use the GlobeTraff [183] traffic generator, rather than DNS traces such as [223] which would reflect the current (instead of a Future) Internet architecture, for instance by omitting requests sent directly to servers like HTTP requests. To estimate the number of resolution requests, we divide the corresponding data volume with the median[42] IO size for each traffic type in accordance to relevant studies (see [183] and references therein) and end up with the control plane traffic shown in Table 6.4. Then, we use 5 different workload instances to increase randomness, each of which corresponds to 25 GB of traffic and in an average of 2,430,379 subscription messages for 1,032,030 IOs. This size limit is imposed by the resource limitations of the OMNeT++ simulation environment. Finally, we note that each resolution request is injected from a randomly chosen access network.

Routing performance

The routing performance of the NRS affects both the latency perceived by the end-users, as well as the traffic load on the network. We express the routing performance

[42] The choice of the median rather than the mean IO size was made to avoid skewing the results due to the long-tail characteristics of some distributions such as the Pareto tail used for Web IOs.

Table 6.5: Simulation model for Section 6.3.2. The model is also used as part of the simulation model in Section 6.4.

Parameters	Values
Network topology:	Scaled-down [173] Internet topologies with 4400 DHT-NRS RNs (resp., DONA RHs) uniformly across 400 domains, organised in 6 level hierarchies along with multi-homing and peering links.
Workload:	Five synthetic traces generated with Globetraff [183] in accordance to the traffic characteristics of Table 6.4 on page 127. 2,430,379 resolution requests on average, for 1,032,030 IOs advertised by a single publisher each. All `Adv`/REGISTER and `Sub`/FIND requests issued from a randomly chosen RN/RH at an access network.

of DHT-NRS with the *stretch* metric, defined as the *ratio* of the *number* of *inter-domain hops* required for a `Sub` message to reach the RVP and the corresponding `Ntf` message to reach the RN serving the publisher of the desired IO, *over* the *hop count* required by an identical FIND message to reach the same target (the RH that issued the corresponding REGISTRATION) in DONA.

1) *DONA significantly outperforms DHT-NRS in routing efficiency at the cost of extensive IOE replication:* Figure 6.2a on page 130 shows the stretch values derived for several scenarios with different cache sizes. Stretch ranges from 2.84 without caching to 1.95 in the Infinite Cache Size (ICS) scenario. *Under current traffic patterns, DONA significantly outperforms DHT-NRS in routing efficiency.* This is only possible however because DONA extensively replicates IOEs throughout the hierarchy to guarantee the existence of the desired registration on the shortest, *policy-compliant path* towards the publisher.

2) *Caching in DHT-NRS can not deal sufficiently with path-stretching:*

   2.1) *Caching results in low hit ratios and correspondingly low path-stretching benefits:* Figure 6.2b shows that cache hit ratios are 27.17%, 31.75%, 37.76% and 53% for the 50%m, 100%m, 150%m and ICS scenarios, respectively. The corresponding savings in the resolution path lengths are 13%, 15.39%, 18.24% and 31%. Evidently, such gains are *not sufficient* to overcome nor to mitigate the problem of path-stretching.

   Regarding the low hit ratios on the one hand, caching adapts reactively to users' request patterns, thus many non-popular IOEs (including the ones which will not be requested again) get cached and quickly evicted, causing an inefficient utilisation of the available cache space. Furthermore, cache *effectiveness* for less popular IOs is known [14, 56] to increase *only logarithmically* with the size of the cache, while such niche demand is significant (Section 5.1.2 on page 74). These facts explain the *(i) lacking hit ratio increase* with each

+50%m increase on the x-axis of Graph (b) and *(ii)* the fact that the hit ratio can *not exceed* the value of *53%* even with *unrealistically* large ICS caches.

Regarding the small path length reduction on the other, evidently one reason lies in the low hit ratio achievements. Another reason lies in the fact [167, 168] that 70-80% of requests are not only niche, but also one-timers inside edge networks (Section 4.2.2 on page 49) where most of the end-user requests originate from. This implies that subscriptions are less likely to hit a nearby (or local) cache early enough (resp., directly), which would resolve the named request in just a few hops (resp., immediately) with significant path length reduction benefits.

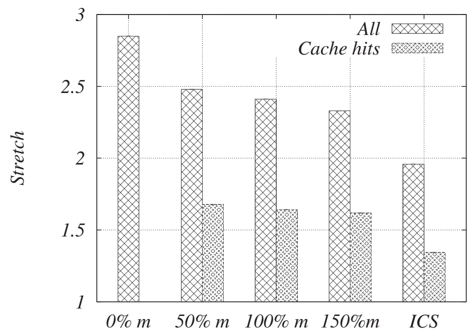
2.2) Resolution paths after cache hits are at least 34% longer than in DONA: The former performance numbers are subject to the popularity characteristics of the workload.

Therefore, we also consider the stretch value of resolution paths for which the resolution requests hit a cache in order to explore the potential benefits of caching in scenarios where IO popularity does allow higher cache hit ratios. In these cases, caching results in a direct pointer to IO publisher(s) by skipping the remainder of the resolution path to the IO's RVP. As shown in Figure 6.2a, stretch ranges from 1.34 in the best case hypothetical scenario of ICS to 1.67 in the 50%m scenario. But even this substantial reduction for popular IOs yields at least 34% longer paths compared to DONA. This is due to the intrinsic indirection of DHT-NRS, as the RVP may not reside in the shortest path from a subscriber and a publisher. However, caching along with the routing properties of H-Pastry should further shorten resolution paths by selecting the closest publisher in the case of multiple (e.g., replica) publishers per IO.

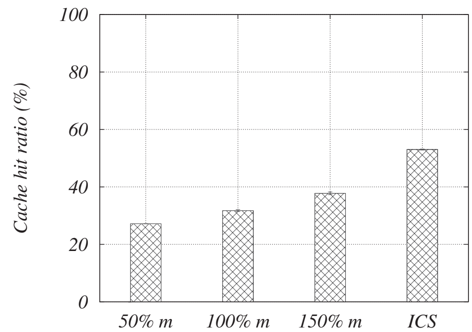
State size and distribution

In our evaluation, we use the term "state" to refer to the IOEs maintained at each node of the name-resolution system. The state size is related to the total number of IOs and determines the amount of resources required to support the operation of the architecture *wrt* memory and lookup processing load. Figure 6.2c shows the cumulative distribution function of the state size for both DHT-NRS and DONA. Note that the x axis is in log scale.

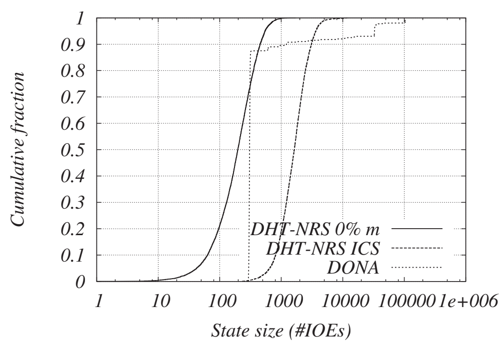
1) The state maintained in DHT-NRS RNs is considerably lower and more uniform than in the RHs of DONA: The difference between the two architectures is significant, with the size of the state maintained by RN nodes in DHT-NRS being considerably lower than the corresponding load imposed on DONA's RHs. More importantly, we see that DHT-NRS achieves a more uniform distribution of state across the participating nodes compared to DONA. For instance, 95% of the RNs



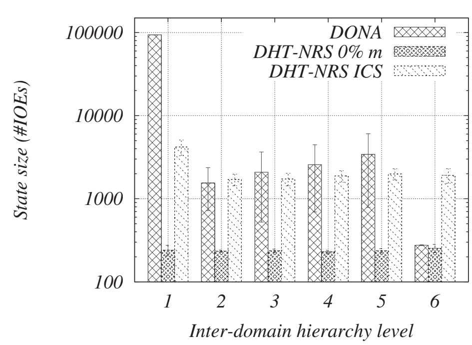
(a) Inter-domain stretch of DHT-NRS



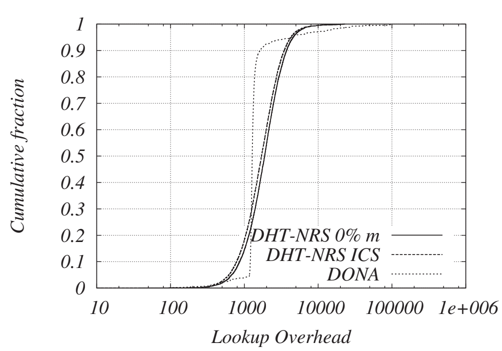
(b) Cache hit ratio



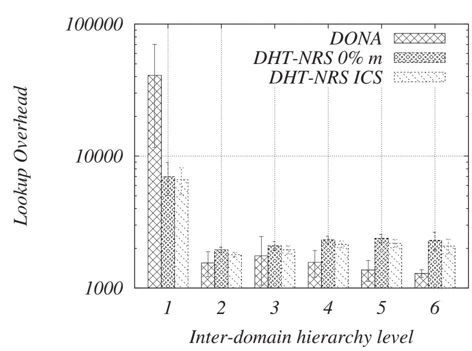
(c) Cumulative state overhead



(d) Distribution of state across hierarchy levels



(e) Cumulative lookup overhead



(f) Distribution of lookup overhead across hierarchy levels

Figure 6.2: Performance evaluation results: DHT-NRS vs. DONA. Cache size in DHT-NRS is expressed as a proportion of the median number (m) of registration entries per RH in DONA. We choose the median value because the distribution of state in DONA is considerably skewed, as denoted by Graphs (c) and (d), and the evaluation results over full-scale topologies presented latter in Section 6.4. Moreover, we examine scenarios with an Infinite Cache Size (ICS) to reflect the upper limit for the performance of caching in DHT-NRS.

in DHT-NRS maintain less than 550 IOEs (0%m scenario), while 95% of the RHs in DONA maintain up to almost 33000 IOEs. The highly skewed state distribution in DONA is due to the accumulation of registrations at higher levels of the inter-domain hierarchy, and poses a significant challenge for network operators, who would have to resort to dense deployments of RHs in order to cope with the associated resource requirements.

2) The inter-domain topology structure causes the concentration of excessive state in at the top-most levels of the hierarchy in DONA: Moreover, it is important to point out the relation of the observed state distribution skewness to the structure of the inter-domain topology. In the considered topologies, slightly less than 50% of access networks reside at level 2, meaning that a major part of the registrations is stored only at the first two levels of the hierarchy. This is demonstrated in Figure 6.2d which shows the average state size per node at each level of the hierarchy. Note that the y axis is presented in log-scale. This figure suggests that the inter-domain topology structure causes the concentration of excessive state in DONA at the top-most levels of the hierarchy. These top-level domains face disproportionate overhead compared to lower level domains, incurring the corresponding CAPEX/OPEX overheads, Note, that return to discuss this matter in further in Section 6.4, were we also show that lower-level domains can face an even worse problem regarding disproportionate state overheads due to sibling relations, making cloud-based solutions a compelling option as suggested in Section 6.4.4.

3) RHs in DONA have higher memory requirements even with respect to RN with unlimitted cache in DHT-NRS: Figure 6.2c and Figure 6.2d also show the amount and distribution of state required to avoid cache replacement (i.e., the ICS scenario). This hypothetical scenario requires a considerable amount of memory which is highly unlikely to be available in practice. However, even in this scenario, almost 8% of RHs in DONA have higher memory requirements than any RN in DHT-NRS.

4) State is distributed more evenly in DHT-NRS across the hierarchy levels: Again in Figure 6.2c and Figure 6.2d we see that in DHT-NRS state is distributed more evenly across the hierarchy levels, with the exception of level 1 RNs in the ICS scenario where the routing scheme causes a concentration of cached IOEs (see also the next section).

Processing overhead

The state size per node determines the processing overhead per IOE lookup at the RVP and the overall processing overhead is determined by the actual number of lookups performed by a node. For each architecture, we define the Lookup Overhead (LO) metric as the sum of the total number of `Sub`/FIND messages forwarded by each RN/RH respectively, and the total number of messages *terminating* at each node i.e., triggering communication with the publisher. Figure 6.2e shows the cumulative distribution function of LO for both architectures.

1) *A major fraction of DONA RHs is subject to less lookup overhead than RNs in DHT-NRS:* This is due to the considerably worse routing performance of DHT-NRS that results in `Sub` messages traversing longer networking distances and thus consuming resources at more intermediate RNs.

2) *Disproportionate overhead for the top-level domains in DONA:* However, a closer look at the LO distribution across the inter-domain hierarchy reveals a disproportionate overhead for the top-level domains in DONA (Figure 6.2f). Evidently, the fact that a major part of the access domains resides at level 2 of the hierarchy results in a corresponding resolution overhead for the top- level domains, which again calls for the use of considerable processing (as well as memory) resources.

3) *RNs at the top-most domains of the hierarchy in DHT-NRS are subject to considerably higher processing overhead compared to lower level RNs:* Interestingly, Figure 6.2f also shows that RNs at the top-most domains of the hierarchy in DHT-NRS are subject to considerably higher processing overhead compared to lower level RNs. The additional overhead is due to the forwarding of `Sub` and `Ntf` messages (ID ownership is evenly distributed across the RNs) and is attributed to the structure of the inter-domain topology and the design of H-Pastry: to adapt to the inter-domain hierarchy structure, H-Pastry causes the top level domains to be included in the first non-local ring of each of the access networks at level 2. Therefore, paths towards non-local RVPs are most likely to pass through these domains. As a substantial fraction of access networks resides at level 2, a proportional number of IOs are served from publishers at these domains, resulting in a significant part of the overall `Ntf` messages reaching RNs at these domains via the top-level domains. Similarly, H-Pastry causes the `Sub` messages originating from level 2 access domains to first traverse the top-level domains before taking a downhill direction towards the RVP.

Advertisement/registration overhead

In addition to subscriptions, signalling overhead is also generated by IO registrations. We characterize this overhead as the total number of single inter-domain hop transmissions required for the registration of a single IO to complete. Our measurements show that DHT-NRS requires on average 6.34 inter-domain transmissions per IO in the $0\%m$ scenario, while DONA requires on average 35.56 transmissions. These numbers indicate *an excessive inter-domain traffic load in the case of DONA*, attributed to the (limited) flooding method used to disseminate registration messages to the upper levels of the inter-domain hierarchy (as well as to peering domains). Multihoming plays an important role in this as it results in registration messages being transmitted to multiple domains at higher levels. In the employed topologies 56.75% of all domains are multihomed, with 2.4 providers on average. Unlike DONA, by carefully partitioning the identifier space DHT-NRS forms a structured overlay that only requires the targeted routing of `Adv` messages.

6.4 Route-by-name schemes

The feasibility weaknesses of DHT-NRS discussed in the previous sections, along with the background discussion on the drawbacks of DHT-based solutions on page 120 of Section 6.2.1, turn our interest in this section to route-by-name approaches. However, the performance evaluation results in the graphs of Section 6.2 denote that DONA, which is a route-by-name NRS, faces feasibility issues as well. Only unlike DHT-NRS, DONA *suffers* from *poor scalability* due to the impact of the the inter-domain topology structure, which causes the disproportionate concentration of state at the top-level domains of the hierarchy. On the positive side, a major fraction of DONA RHs is subject to less lookup overhead than RNs in DHT-NRS. Moreover, DONA has an intrinsic *zero path-stretch* overhead property by adapting fully to the underlay topology, which unfortunately comes with a trade-off in terms of a routing state establishment overhead that outreaches the corresponding DHT-NRS overhead by 1100% due to multi-homing and peering agreements.

The former fuel our interest in further investigating the feasibility of route-by-name designs in two ways: First, in the following subsections we focus on quantifying the performance aspects of the CURLING ICN NRS using DONA as a performance benchmark, in order to examine its scalability and, hence, its feasibility. The design in CURLING tries on the one hand to preserve the merits of DONA, while on the other to reduce the extend of its identified scalability and load balancing problems. Second, the observed resource requirement imbalances in DONA both previously in Section 6.3.2 as well as later in Section 6.4, trigger a discussion on the need for large-scale centralized deployment solutions for route-by-name designs based on cloud computing in Section 6.4.4.

6.4.1 Simulation model

As previously discussed, the name resolution performance of DONA and CURLING heavily depends on the structure of the underlying inter-domain topology. However, studies reveal a number of approximately 45K ASs and approximately 200K annotated links for the entire inter-AS graph [62], which make such a performance evaluation a technical challenge on its own. We have therefore considered two complementary approaches discussed below:

1) *Full-scale network topology scenarios:* We used the AS-level Internet graph inferred by the CAIDA BGP traces [62]. On top of this graph we developed a custom DONA/CURLING simulator which simulates the registration of IOs across the

Table 6.6: Simulation model for Section 6.4. The setup for the scaled-down scenarios is the same as the one outlined in Table 6.5 on page 128 for the evaluation purposes of Section 6.3.2.

Parameters	Values
Network topology (scaled-down scenarios):	Scaled-down [173] Internet topologies with 4400 DHT-NRS RNs (resp., DONA RHs) uniformly across 400 domains, organised in 6 level hierarchies along with multi-homing and peering links.
Network topology (full scale scenarios):	AS-level Internet graph inferred by the CAIDA BGP traces [62]: ∼45K ASs and ∼200K annotated links in the entire inter-AS graph.
Workload (scaled-down scenarios):	Five synthetic traces generated with Globetraff [183] in accordance to the traffic characteristics of Table 6.4 on page 127. 2,430,379 resolution requests on average, for 1,032,030 IOs advertised by a single publisher each. All `Adv`/REGISTER and `Sub`/FIND requests issued from a randomly chosen RN/RH at an access network.
Workload (full scale scenarios):	Each stub AS generate a single distinct REGISTER message.

domains. We neglect processing, queueing and transmission delays, to enhance the scalability of the simulator. For the same reason, our measurements consider only one RH per AS. Due to these limitations, we employ the CAIDA trace set only to study the distribution of state across the inter-domain topology.

2) Scaled-down network topology scenarios: We employ the same scaled-down inter-domain topologies as the ones we discuss in Section 6.3.2.

In both setups we consider REGISTER messages generated only by stub networks. This reflects (i) the currently dominant service/business model where content providers may connect at any level of the inter-domain hierarchy to push their content, without providing transit services, and (ii) the cases of user generated content, with users connected under stub networks of the hierarchy. With multihoming, RHs forward registrations to the RH of their randomly selected default provider AS.

Since the Internet inter-domain AS graph is far from hierarchical, due to the prevalence of multihomed ASs and peering links, in this paper we adopt the methodology proposed in [221] to classify ASs into four (Table 6.7 on the facing page) tiers based on the size of their customer *cone*, i.e. the total number of their downstream customers.

Considering the simulator scalability limitations discussed above, we again consider two different workload models, one per topology model. For the scaled-down topology, we adapted the setup as the one described in simulation model of Section 6.3.2. For the large scale CAIDA trace, we reduce the computational requirements by having each stub AS generate a single distinct REGISTER message. This convention obviously corresponds to the simplified, uniform distribution of content across the Internet. However, it provides a rough insight on the distribution of state across the hierarchy. A more

Table 6.7: AS classification into tiers based on customer *cone* size.

Stub networks	Networks that have no more than 4 customer networks. By definition, all access networks are stub networks.
Small ISPs	Internet Service Provider ASs that have a cone size between 5 and 50.
Tier-1 ASs	ASs at the highest level of the hierarchy that do not act as customers for any other another AS. All such ASs have peerings to every other Tier-1 AS, creating a mesh.
Large ISPs	This category includes the remaining ASs which have a larger cone than small ISPs but are not Tier-1 ASs.

Table 6.8: Performance evaluation metrics for Section 6.4.

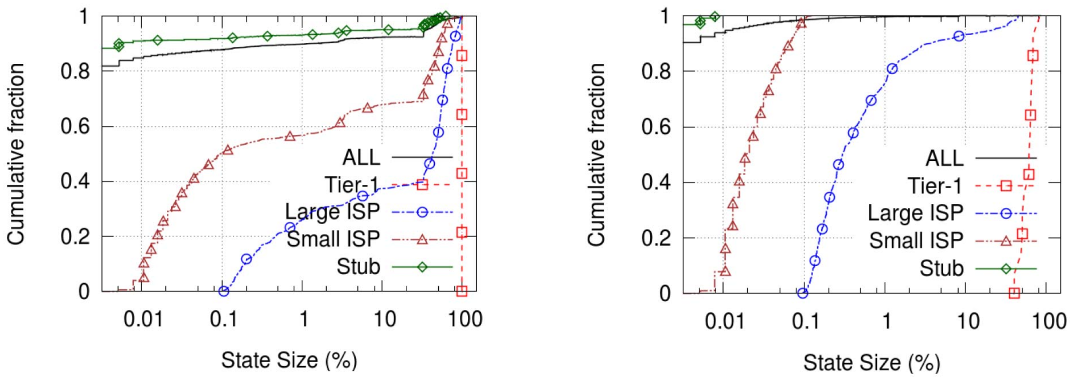
Metric type	Description
Lookup Overhead (LO):	The *LO* metric, previously defined and used on page 131, is the sum of the total number of resolutions messages forwarded by each RH and the total number of messages terminating at each node.
Resolution delay:	The number of *inter-domain hops* taken by a name resolution request until the requested IO is resolved at some NRS node.
Registration overhead:	The aggregate number of *single hop transmissions* needed for all REGISTER messages to reach their target(s) and register all IOs in the inter-network.
State size:	$m = \frac{RE_{total}}{RE_{unique}}$.

realistic evaluation would have involved an uneven IO distribution across the topology, but to the best of our knowledge, there is no such model. Note that we do not consider resolution requests in order to further reduce processing and memory requirements, as the added overhead is at least of the same order of magnitude as for registrations.

6.4.2 Performance evaluation metrics

In the following, we discuss in further some of the metrics used in our evaluation and which we outline in Table 6.8. The *size of the state* maintained by the NRS determines the resources required in terms of memory and lookup processing load. Unless otherwise stated, we assume that state is organized as in [72], namely, each IO entry implies a 40 byte identifier and a 2 byte pointer to the next RH. We evaluate state size on a *a) per RH* and on a *b) per AS tier* basis. In the former case, we attempt to quantify the overheads imposed to individual ASs, indicating the level of NRS deployability in terms of deployment costs. In the latter case, we aim express the expected burden at different levels of the AS hierarchy.

To achieve these goals, we measure the number of IOEs per AS and *normalize* it as a percentage of the overall state size (i.e., unique IOEs) throughout the inter-network.



a: DONA

b: CURLING

Figure 6.3: Distribution of state size across the AS-level topology (CAIDA 2013 traces).

Then, we quantify the level of replication in each NRS scheme by defining the *multiplier* m as the ratio of the total number of registration entries maintained throughout the inter-domain topology (RE_{total}) to the total number of unique IOs (RE_{unique}):

$$m = \frac{RE_{total}}{RE_{unique}} .$$

Regarding *resolution delay*, Likewise to the results of Section 6.3.2, we do not take into account lookup processing, queuing and transmission delays, so as to isolate the impact of state placement across the hierarchy in each considered scheme.

Finally, regarding registration overhead, the registration of IOs may result in a significant overhead on the control plane due to the vast size of the information space. We investigate this aspect by measuring the aggregate number of single hop transmissions needed for all REGISTER messages to reach their target(s) and register all IOs in the inter-network.

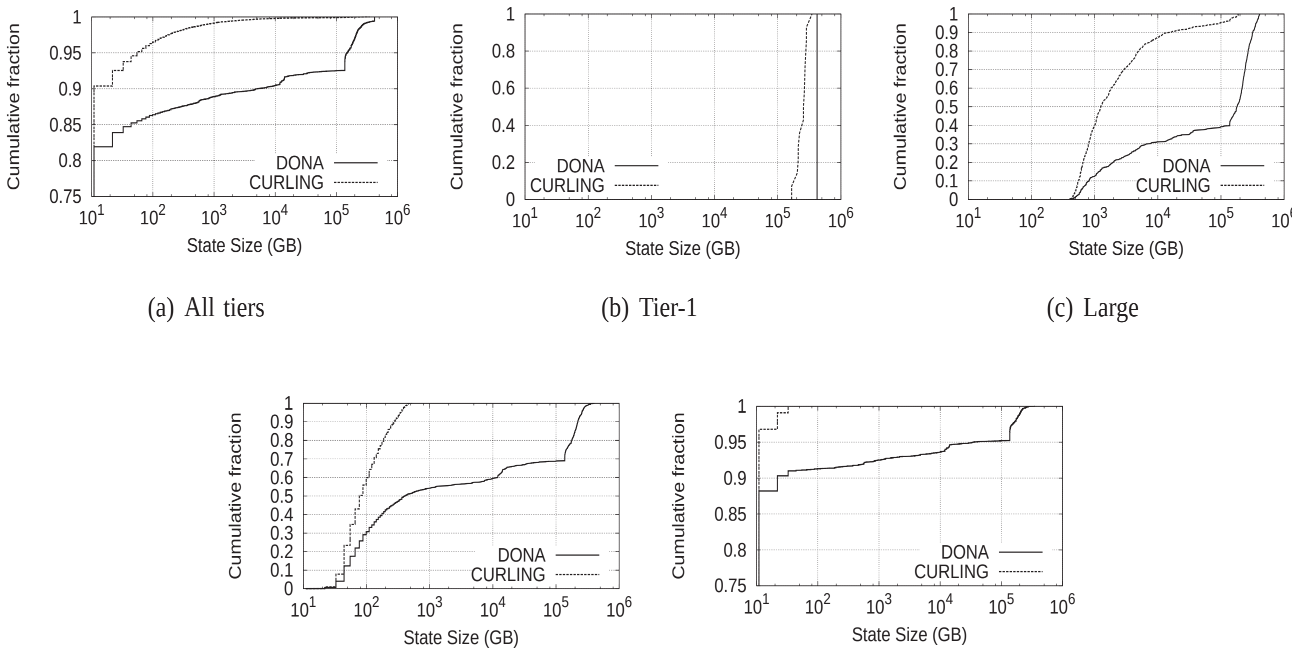
6.4.3 Performance evaluation results

Table 6.9: State size per AS as a percentage of the total state size throughout the inter-network.

	DONA		CURLING		
Type	**Average**	**Median**	**Average**	**Median**	**Avg. gain**
All Tiers	3.778	0.003	0.060	0.003	62.97
Tier-1	100.00	100.00	59.895	61.769	1.67
Large ISP	36.701	42.687	2.758	0.298	13.31
Small ISP	15.599	0.097	0.029	0.018	537.90
Stub	2.039	0.003	0.003	0.003	679.67

Full-scale topology

Figures 6.3a and 6.3b on the current page show how state is distributed to ASs across all topology tiers and per tier, for DONA and CURLING, respectively, based



(a) All tiers (b) Tier-1 (c) Large

(d) Small (e) Stub

Figure 6.4: Comparison of state distribution per tier (CAIDA 2013 traces).

on the 2013 CAIDA topology model. The y-axis shows the cumulative fraction of ASs that together maintain the fraction of the total state size indicated on the x-axis. We also provide the average and median values fractions of the total state held by each AS across all tiers and per tier in Table 6.9. The replication factors are $m_{DONA} = 1702.64$ and $m_{CURLING} = 27.34$, indicating that CURLING provides very large improvements against DONA in the state required per AS, although, as the last column of Table 6.9 shows, the gains are very dependent on the AS category: while the average x-fold improvement of CURLING over DONA is 62.97, it ranges from 1.67 for Tier-1 ASs to 679.67 for stub ASs. More than 81% of ASs in DONA and 90% in CURLING are only burdened with their own registration entries, while the remainder of the ASs (8150 and 4335 ASs in total, for DONA and CURLING, respectively) are disproportionally loaded with state, reaching even the entire available state on the network in the case of Tier-1 ASs in DONA. This is a direct consequence of the structure of the inter-domain topology: the vast majority of ASs (92.87%) belong to the Stub tier, with only a small fraction (9.17%) having more than one (and up to 4) customers that contribute their registration entries.

Figures 6.4a to 6.4b on this page present a more detailed comparison between DONA and CURLING. These plots show the cumulative distribution of state to ASs, but expressed in GB and corresponding to a total of 10^{13} IOs in the inter-network. We can see that the overall state on the network is in the order of several hundred TB (420 TB for 10^{13} IOs), and this is also the total state size handled by each Tier-1 AS in

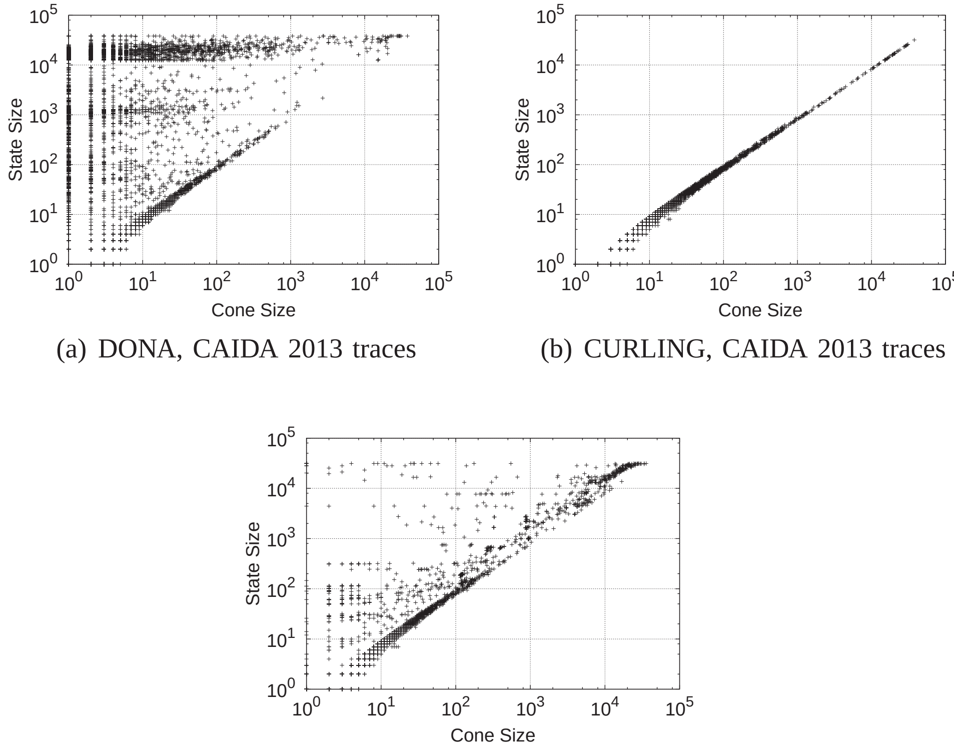
DONA. As expected, CURLING accumulates considerably less state per AS compared to DONA: in the case of Tier-1 ASs, the requirements remain in the order of TBs, but they are 40% lower. Approximately 90% of the Large ISPs would be required to maintain up to ten TB of state in CURLING, while this only holds for approximately 30% of the Large ISPs in the case of DONA. The difference between the two schemes becomes more evident at lower tiers, where DONA increases state size by up to 3 or 4 orders of magnitude. Table 6.10 translates the observed state sizes into the corresponding hardware resource requirements, expressed as the number of 16 GB RAM servers required to hold the resolution state in RAM at each AS, per tier[43]. Clearly, both schemes require the deployment of large data centers at Tier-1 to cope with the state size; fortunately, this is required at only a few ASs.

Table 6.10: Number of 16 GB RAM servers required to hold state in RAM.

	DONA		**CURLING**	
Type	**Average**	**Median**	**Average**	**Median**
Tier-1	26250	26250	15723	16215
Large ISP	9635	11206	724	79
Small ISP	4095	26	8	5
Stub	536	1	1	1

The reduced state size with CURLING, also reflected in the substantial difference between the m_{DONA} and $m_{CURLING}$ values reported above, is the direct result of not exchanging IO entries between peering domains. On closer inspection, each tier is defined by the *cone* size of the participating ASs, which in turn determines the size of the accumulated (distinct) registrations created by cone members. This state is augmented in DONA by the entries received from peering ASs, as clearly demonstrated by the graphs of Figure 6.5, which shows the relation between AS cone size and state size. The large concentration of points in the diagonal shows that the amount of state accumulated by ASs is mainly affected by the size of their cone. The scattered points above the diagonal in Graph 6.5a show the impact of peering relationships in DONA. This is most evident at the upper right part of the graph which depicts the full mesh connectivity at Tier-1. It is also important however to note the existence of several cases where ASs with a relatively small cone size maintain state of disproportionate size in DONA (upper left area of Graph 6.5a), reflecting the establishment of peering relationships with ASs at higher tiers (and therefore large cone sizes), which results in the reception of large amounts of state. In contrast, Graph 6.5b shows no points above the diagonal for CURLING. For the vast majority of ASs, CURLING results in a state overhead largely determined by their cone size. This is particularly important, since smaller ASs, residing lower in the hierarchy, are not forced into large infrastructure deployments only to resolve large content volumes residing at other domains. The points below the diagonal correspond

[43] For simplicity, we assume that all 16 GB are used to hold name resolution state and ignore indexing overheads.



(a) DONA, CAIDA 2013 traces

(b) CURLING, CAIDA 2013 traces

(c) DONA, CAIDA 2011 traces

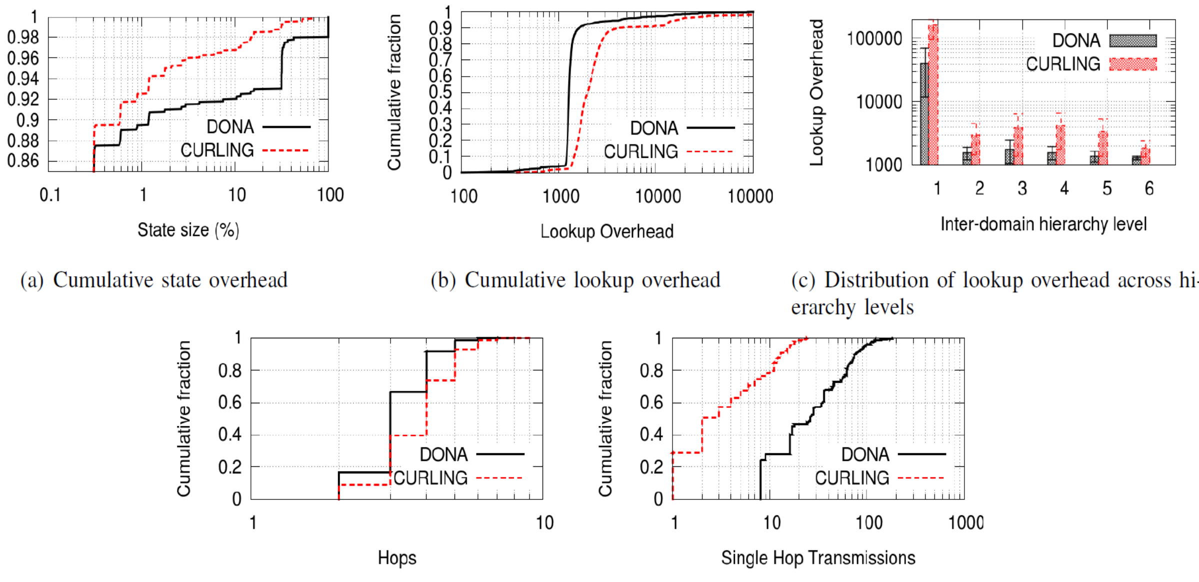
Figure 6.5: Relation between AS cone size and accumulated registration state in log scale.

to ASs whose cone also contains non Stub ASs, which do not generate registrations.

The impact of peering relationships on state overhead is of particular importance, due to the evolution of the inter-domain topology graph as a result of the increasing establishment of peering links, especially by large content providers such as Google [224]. Unfortunately, the identification of peering links remains a difficult task. As argued in [225] the followed CAIDA methodology fails to reveal up to 90% of existing peering links. For the purposes of our study, we repeated our measurements over the 2011 CAIDA traces [226] in which the reported number of peering links is 3523. This number is significantly lower compared to the 66617 peering links in the corresponding 2013 CAIDA traces [62]. The impact of the additional peering links becomes obvious by comparing Graph 6.5a and 6.5c, where we can see how the additional peering links have multiplied the points above the diagonal.

Scaled-down topology

Figure 6.6a shows the same trend for the cumulative distribution of state size per node in DONA and CURLING, as in the case of the CAIDA trace based measurements (see Figure 6.4a), confirming the validity of our scaled-down approach. There is a sharp increase in the set of ASs that present a relative state size larger than approximately 25-30% of the total state size in the case of DONA, which is also observed with the CAIDA trace, due to the exchange of state over peering links with higher tier ASs.



(a) Cumulative state overhead

(b) Cumulative lookup overhead

(c) Distribution of lookup overhead across hierarchy levels

(d) Cumulative distribution of hops per resolution request

(e) Cumulative distribution of single hop transmissions per registration

Figure 6.6: Simulation based performance results (scaled down topology).

Figure 6.6b shows the cumulative distribution of the lookup overhead per node as defined in Section 6.4.2. On average, CURLING incurs a 2.78x increase on lookup processing load compared to DONA, as resolution requests must be forwarded to the closest (according to BGP) common ancestor in the topology, even when the requested item resides in a peering AS (or one of the ASs in its cone). Figure 6.6c further shows how lookup overhead is distributed across the levels[44] of the hierarchy. In both cases, the top-most domains receive a substantially increased load than their lower level counterparts. This is because the top-most domains end up always serving requests that could not be resolved lower in the hierarchy, which is the typical case when requests and IOs reside in different areas of the inter-network. Obviously, the exchange of state over peering links (in DONA) does not suffice to balance the load among the hierarchy levels. Nevertheless, we notice that lookup overheads are consistently higher for CURLING at all levels. Since CURLING does not allow the exchange of state across peering links, FIND messages reach higher levels in the hierarchy more frequently. In the case of the top-most domains, the workload increase is 3.9x on average.

Figure 6.6d shows the cumulative distribution of hop counts for name resolution, showing that DONA requires less hops to resolve an IO identifier as RHs maintain resolution entries from peering domains, i.e FIND messages often reach their destination

[44] The level of a domain is the minimum hop- count distance to the root.

without reaching a Tier-1 domain. On average, DONA requires 3.26 hops per resolution against 3.86 hops in CURLING i.e., DONA leads to a 16% average reduction in hops. Finally, Figure 6.6e shows the cumulative distribution of the single hop transmissions for registrations in both DONA and CURLING. Registering an IO with DONA requires 35.57 single hop transmissions on average, against only 5.20 in the case of CURLING i.e., an overhead of more than 684%, due to the propagation of registrations over peering links. Following [72], we can estimate a total average/median load of 66.13/66.13 Gbps (DONA) and 39.61/40.85 Gbps (CURLING) of registration traffic at Tier-1 domains, for a total number of 10^{13} IOs with an average lifetime of two weeks and a Registration size of 1 KB. These are obviously non- negligible data rates for control plane operations, even though they only refer to a few, tier-1 ASs, and are within reach of current technology capabilities. The expected rates drop for Large ISPs, Small ISPs and Stub networks to 24.28/28.23 Gbps, 10.32 Gbps/64.15 Mbps, 1.35 Gbps/1.98 Mbps (DONA) and 1.82 Gbps/197.08 Mbps, 19.18/11.90 Mbps, 1.98/1.98 Mbps (CURLING).

6.4.4 Cloud Computing for addressing Scalability concerns

Our evaluation results on DONA indicate that DONA's requirements can only be met by exploiting cloud techniques. As outlined in Section 1.1.2 on page 7, the service model of modern clouds offers a series of advantages to customer organisations, particularly to the *smaller* ones with limited resources which can not afford the needed local CapEx and OpEx. Most notably, cloud computing offers: (i) a pay-for-use charging model, (ii) fast deployment, (iii) ubiquitous network access, (iv) availability and advanced security, (v) lower costs, (vi) better resiliency, (vii) scalability and elasticity via rapid (de)provisioning of resources, etc. Particularly *elasticity*, is an important property for deploying a system like DONA to the cloud, as it refers to the ability to address large resolution loads and to adapt quickly to workload changes (i.e., resolution flush crowds or DoS attacks) by (de)provisioning resources, hence allowing low-cost data storage solutions, along with other merits regarding disaster recovery, on-demand security controls, real time detection of system tampering and so forth.

Modern cloud data centres scale to over a 100,000 servers, each one with enough resources to host several tens of virtual machines. At the same time, storage costs are constantly decreasing [227], which allows modern data centres to store huge volumes of data as needed in the case of DONA. In this context, the conservatively calculated resource requirements of DONA discussed in Section 6.3.2 and Section 6.4.3 could be well supported by current cloud data centre capacities. However, this is only a rough indication of the technical feasibility of a DONA deployment. A series of issues emerge when attempting to gain a more concrete view on the realization of such an architecture in the context of cloud computing.

The impact of AS topology

Performance results demonstrate that the distribution of state overhead, and the associated resource requirements, are very skewed across and within tiers. With the notable exception of tier-1 ASs, a diverse set of requirements exists for the various ASs. This results in a discrepancy in the size of the corresponding investment for the deployment of RHs in private or public cloud data centres and, hence, an imbalance in the CapEx/OpEx of ASs for the realization of DONA. This imbalance is in principle determined by the structure of the AS level graph and in particular the cone size, as well as the number of peering links, of each AS. Revisiting these results, we see that for the vast majority of ASs, DONA deployment would not require a very large investment. In this respect, modern cloud data centre capabilities seem to primarily provide the required solution for tier-1 ASs. However, particular attention must be paid to the effect of peering links. As shown in Graph 6.5a on page 139, several ASs residing at lower levels of the hierarchy have significant resource requirements for the support of DONA, due to the existence of peering links with higher level ASs. Moreover, following the trend of adding peering AS relations, particularly to large content providers[45], these requirements have significantly increased compared to the results of Graph 6.5c. Not only that, but also credible studies [221] report that a significant fraction of the established peering links is not present in public BGP data, especially in lower tier ASs, where this fraction reaches up to 90% for large content providers, hence we can only speculate that the situation would be in practice even worse than what depicted in Graph 6.5a. This means that the cloud will be critical in enabling DONA even at lower hierarchy levels. At the same time, it further shows that the corresponding CapEx/OpEx is far from proportional to the size of an AS (as expressed by the customer cone size).

The role of virtualisation

Current experience from large scale indexing and search services shows that cloud services can satisfy high requirements for computing and storage resources. However, the role of the cloud in the realization of an architecture like DONA can not be restricted to the provision of large volumes of resources. By enabling multi-tenancy, virtualisation in cloud computing allows a significant reduction of the associated CapEx/OpEx of ASs. This means that the economic benefits of cloud computing could actually ease the realization of future Internet architectures like DONA. At the same time, virtualisation further enables the *efficient up/down scaling* of the allocated resources in cases of significant workload variations. Such variations can be caused by the establishment of an additional peering link between ASs, the interconnection of a new content provider or the emergence of particularly popular IOs (flash crowds).

---

[45] In a similar vein, it is important to note the potential imbalance of resource requirements when considering the distribution of content providers (publishers) in the inter-network. It has been recently shown that no more than 150 large content providers are responsible for producing 50% (or perhaps even more) of the Internet traffic (e.g., YouTube [220]).

Nevertheless, deploying DONA in the cloud comes with the cost of *routing stretch*. Recall that DONA is designed and tailored to the hierarchical inter-network substrate, and that RHs propagate resolution messages along the lines of established BGP-compliant routes. As a result, resolution in DONA does not suffer from stretch, an advantage which can cease to be valid when (partially or fully) offloading DONA's functionality to the facilities of some public Cloud Service Provider (CSP) such as [228] or [229]. Pushing incoming requests from lower level ASs to public data centres would unavoidably result in added stretch during resolution. There is therefore a need to consider different modes of deploying DONA in the cloud, in addition to the default option of relying on public cloud providers.

One alternative to public cloud services is to invest in private facilities for data centres. While public cloud services seem to be a reasonable option for small scale ASs, private facilities seem more appropriate for large ASs and Tier-1 providers which, unlike small ASs, have the ability to invest in data centres so as to avoid the costs implied by unnecessary stretch. Also, note that such ASs need to deal with massive amounts of state and requests for resolution. Nonetheless, even small scale ASs may face huge demands after deploying DONA, as some small scale ASs are peers with either ASs residing higher in the hierarchy or big content providers such as YouTube. A *hybrid cloud approach* could work as a remedy for such ASs, based on small scale investments on private data centre facilities to directly resolve (or accelerate resolution through replication or caching) "privileged" IOs. By *privileged*, we refer to any sort of content or service that has to be "specially treated" such as content from big content providers who are willing to pay for fast (i.e., stretch-less) resolution, or popular information for which caching data locally can reduce the cost of data transport through ISPs. For the remaining IOs, resolution can be offloaded to public cloud providers.

Enabling content distribution services through caching

On an another front, cloud capabilities are seen as an excellent means to facilitate DONA's caching functionality. By offering large volumes of storage, the cloud can therefore further contribute to the enhancement of the overall DONA service model, enabling network operators to couple name resolution with content distribution services. Taking a step further, these services could be enhanced by considering other features such as streaming capabilities, content pre-fetching mechanisms, etc. At the same time, the already established communication paths between different ASs, could be used for the gradual formation of inter-domain content delivery platform. This is in alignment with the currently observed market trends where network operators are getting involved in the CDN market [230].

6.5 Conclusions

This chapter was dedicated to our research study towards a *feasible* and globally *scalable* Information-Centric Networking Name Resolution System. Regarding the first part of our effort with respect to the design and evaluation of a novel Distributed Hash Table-based Name Resolution System (DHT-NRS), our findings portrayed in the graphs of Figure 6.2 on page 130 reveal that DHT-NRS may be a scalable choice, yet *not feasible one* in global adaptation scenarios. The merits of DHT-NRS regard (i) state size & distribution and (ii) the implied processing overhead. Nonetheless, route caching in DHT-NRS can not compete with the extensive replication of routing information in DONA under current traffic patterns, yielding stretched path values ranging from 1.95 to 2.84. Regardless of the comparison to DONA, such highly stretched paths signify *unacceptable routing policy violations*, leading to a dead-end because DHTs may still not adhere to routing policies even by increasing the average node degree [159]. Last, the design modifications on Pastry discussed on page 123 for the purposes of H-Pastry in global adaptation scenarios of DHT-NRS turn to be (i) complicated with respect to the creation and maintenance (joining or leaving of NRS nodes) of the routing tables referring to other levels in the domain hierarchy, and (ii) *infeasible to apply* in practice as the different ISP ASs would need to share confidential information about their internal organisation during the formation of H-Pastry rings as explained on page 123.

The feasibility weaknesses of DHT-NRS described above, along with the background discussion on the general drawbacks of DHT-based solutions provided on page 120 turn our interest to route-by-name approaches. In particular, we focus on DONA and CURLING in global adaptation scenarios and conclude that *CURLING* can be considered as *a possibly feasible ICN NRS* in Internet-wide deployment scenarios. Its requirements are far more modest than DONA's, which is encouraging, as route-by-name NRSs respect BGP policies and AS peering agreements, contrary to their lookup-by-name counterparts. To this end, we discuss in Section 6.4.4 the feasibility of leveraging resource virtualisation and scalable storage abilities of private, public or hybrid cloud facilities for the cases of larger (Tier-1/Large) and smaller ASs, respectively. Although the discussion there is more focused on DONA because it faces larger scalability issues, the discussion can be extended to the case of CURLING. However, we note that leveraging the potentials of cloud computing should come at the cost of adding stretch to routing paths, while the interest on reducing the requirements at the higher levels of the hierarchy remains paramount.

Regarding our performance findings on CURLING (using DONA as a benchmark), we have shown that CURLING (and also DONA) lead to an extensive replication of state, as the replication factor across the inter-domain topology is 27.34 (resp., 1702.64 for DONA). Additionally, the distribution of state across the topology is heavily skewed as Tier-1 ASs must maintain 59.89% (resp., 100% for DONA) of the entire state in the inter-network, leading to the requirement for the deployment of data centres at the scale of 20K servers. However, 90% (resp., more than 81% for DONA) of all ASs only hold state for local content leading to moderate deployment costs for the vast majority of ASs. In another positive finding, we show that CURLING reduces the state size 62x by *not* forwarding registrations across peering ASs, and even up to 679x for stub domains, compared to DONA. As a result, small ASs are not flooded by state from higher tier peers and CURLING deployment costs become proportional to AS (cone) sizes. Finally, CURLING pays a cost of an average 2.78x increase of lookup processing load compared to DONA. As intuitively expected, the topmost domains pay the highest penalty i.e., 297% additional lookups on average. On the positive side, CURLING reduces bandwidth requirements to a few hundred Mbps for the majority of Large ISPs.

Chapter 7

Future Research

One extension to our proactive caching models presented in Chapters 4 and 5 is to allow mobile users to obtain their requested IOs after a cache miss from another cache that has proactively fetched the IOs, provided that this action implies a smaller data transfer cost compared to the original source(s) of the IOs. This collaboration can be combined with a further extension: a two-level caching model in which the cache points, even though they reside at the same level (at the edge of the network), they form parent-to-child relations in two-level hierarchies.

Hierarchical caching can be also applied to facilitate the collaboration between ISPs or between local ISP caches lying at the edge of the network (e.g. at small cells) and content provider-controlled caches in higher levels of the hierarchy such as within the same ISP or any of its sibling/provider ISPs. Regarding the earlier case, the motivation lies in the cost savings in terms of inter-ISP data transit charges [74] and the solution can be realised with hierarchies that encompass an arbitrary number of cache levels. In the latter case, the objective of such a collaboration is to increase shared profits between ISPs and content providers, as suggested in [231]. Note, that our cost-based cache decision models are ideal for such scenarios, as they naturally integrate service profit gains from cache decisions at all levels.

Possible directions for future work also include *i)* adapting our proposed proactive caching frameworks to dense environments of small cell and Wi-Fi hotspots with *overlapping coverage*, *ii)* investigating *video streaming* and *hierarchical* cache structure applications in heterogeneous wireless networking scenarios and *iii)* extending the suggested utility models discussed in Section 4.4 on page 60 to account for wireless/wired or backhaul link congestion, so as to uniformly address both storage and capacity constraints. Regarding the first point, in particular, the content placement problem in dense small cell/Wi-Fi topologies is NP-hard [143, 144], hence a solution like the one proposed in Chapter 5 that uses congestion pricing can have advantages.

A further interesting direction would be to apply our proactive caching models for the purpose of internal caching in the mobile devices. This way, mobiles can utilise their limited internal cache space with data cached during their sojourn in the small cells or Wi-Fi hotspots in order to *i) further avoid macro-cellular data consumption* after they disconnect from the small cells or Wi-Fi hotspots, or –in a different context– to *ii)* convert mobile devices into *"mules"* carrying data in *delay tolerant*, intermittently connected *mobile ad hoc networks* likewise to the work of [232].

Finally, our proactive caching and congested resource pricing schemes can be applied in software mobility scenarios within the contexts of ICN and cloud or fog computing. These scenarios involve mobile software agents which can relocate virtually anywhere on the network, as discussed in Section 2.2.4 on page 22, rather than mere device/physical mobility. The aforementioned constrained resources can be storage or network capacity for delay critical applications that process a large amount of data, or CPU processing. Regarding ICN scenarios in particular, our proactive caching approach matches our discussion in Section 6.4.4 on page 141 for moving resolution load in the case of route-by-name ICN NRSs to the cloud. In particular, ASs with small capabilities could utilise a limited and/or inexpensive amount of local resolution resources according to our resource congestion pricing scheme, only for part of the resolution state. This can be realised in the form of autonomous actions taken within the AS, or via hierarchical actions when applied in the case of customer-to-provider AS relations. The expectation is to yield resolution cost gains in terms of path-stretching, which comes as a cost for moving state to the cloud.

Appendix A

Special Terminology, Abbreviations and Acronyms

A.1 Special Terminology

The following terms are used within a special context in this Ph.D. dissertation:

Information Object: In this dissertation we use the term “Information Object” and its corresponding acronym “IO”, as well as the terms “information”, “content” and “data-object” interchangeably to refer to *named* pieces of *information*. IOs have a very general context that embraces everything that can be offered via the Internet, from actual data-objects like files, to services and “things” within the context of the IoT.

MAXPOP: Benchmark caching strategy. Max Popularity (MAXPOP) prefetches the *topmost* popular videos at the cache points based on their long-term requests frequency. This form of caching lacks locality knowledge in space and in time, hence it helps to highlight the advantages of adapting up-to-date user mobility in or local content popularity information based on short time-scale measurements to cache actions.

NAÏVE: Benchmark caching strategy. Neglects cache-storage congestion and user mobility. Caches content to all neighbouring cache points with enough available cache space at the time of request. This “blind” form of proactive caching lacks intelligence and is used demonstrate the benefits of adapting user mobility and/or content popularity, as well as cache congestion pricing in cache decisions.

NOCACHE: Benchmark strategy. *No caching* applied for mobility support. Mobiles consume data only from the original sources and via the backhaul of their currently associated AP and the macro-cellular link (if the latter one exists).

ORACLE: Benchmark caching strategy. IOs get cached *only* to the cache points that a mobile will connect, unless the locally available buffer space is not enough. Provides a theoretical *performance upper bound* for proactive caching actions. For clarity, we not that this upper bound is *not* an optimal solution to the distributed cache problem. For such a solution, the reader must refer to the OPTIMAL benchmark cache strategy.

OPTIMAL: Benchmark cache model. Represents the optimal allocation of the distributed cache space. Can be computed *only* for flat caching scenarios, as the problem is hard (Section 4.3.2 on page 57) in hierarchical cooperative caching scenarios.

Small cell: We use the term "small cell" within the context of this dissertation to collectively refer to any type of low-power, operator-controlled BS that: (i) operates in licensed spectrum using either of the established mobile communications standards like the UMTS or the LTE; (ii) connects to some cellular wireless provider via a broadband link; and (iii) its coverage range is smaller than what is defined for *microcells* by industry standards. Based on the above and the definitions in [233], small cells are mainly *picocells* (<~200 m) and *femtocells* (<~10–12 m).

Skewed (mobility model): A skewed mobility model refers to scenarios where there is a clear user trend or pattern for moving towards a certain direction. See also the definition of SKDX% on the current page.

Skewed (content popularity): A skewed content popularity model refers to content popularity distributions where a small number of contents accounts for the majority of content requests, as opposed to the majority of contents in the (so-called) "long tail" of the popularity distribution that correspond to low request frequencies.

Uniform (mobility model): A uniform mobility model refers to scenarios where the number of mobile users moving along each direction is uniformly equal.

A.2 Abreviations

Abbreviated terms or phrases:

SKDX%	Denotes that X% of mobiles move towards the same direction in space, while the remainder (100-X)% adapts a uniform mobility model.

A.3 Acronyms

List of all acronyms used:

ADSL	Asymmetric Digital Subscriber Line
AP	Access Point
API	Application Programming Interface
ARPANET	Advanced Research Projects Agency Network
AS	Autonomous System
BGP	Border Gateway Protocol
BS	Base Station
CAIDA	Cooperative Association for Internet Data Analysis
CapEx	Capital Expenditures
CaR	Content-aware Routers
CCN	Content-Centric Networking
CC	Cached Content
CDM	Cache Decision Module
CDN	Content Delivery Network
CIP	Cellular IP
CN	Correspondent Node
CoA	Care-of Address
COMET	COntent Mediator architecture for content-aware nETworks
CR	Content Router
CRn	Closest Range
CRS	Content Resolution Server
CS	Content Store
CSP	Cloud Service Provider
CURLING	Content-Ubiquitous Resolution and deLivery Infrastructure for Next Generation services
D2D	Device-to-Device
DASH	Dynamic Adaptive Streaming over HTTP
DHT	Distributed Hash Table
DHT-NRS	Distributed Hash Table-based Name Resolution System
DNS	Domain Name System
DONA	Data-Oriented (and beyond) Network Architecture
DoS	Denial-of-Service
EPC	Efficient Mobility-based Caching
EMC-R	Efficient Mobility-based Caching model with cache Replacements
EMPC-R	Efficient Mobility and Popularity-based Caching model with cache Replacements
EPC	Efficient Proactive Caching
FHD	Full High Definition

FIB	Forwarding Information Base
FN	Forwarding Node
GDS	GreedyDual-Size
GUID	Globally Unique Identifier
HAWAII	Handoff-Aware Wireless Access Internet Infrastructure
HD	High Definition
HIP	Host Identity Protocol
HMIP	Hierarchical MIPv6
H-Pastry	Hierarchical Pastry
HTTP	Hypertext Transfer Protocol
HTTPS	HTTP Secure
HWN	Heterogeneous Wireless Network
i3	Internet Indirection Infrastructure
ICN	Information Centric Network
ICS	Infinite Cache Size
ID	IDentifier
IO	Information Object
IOE	Information Object Entry
IoT	Internet of Things
IP	Internet Protocol
ISP	Internet Service Provider
LAN	Local Area Network
LFBL	Listen First Broadcast Later
LFU	Least Frequently Used
LRU	Least Recently Used
LTE	Long-Term Evolution
MaxPop	Max Popularity
MDHT	Multi-level Distributed Hash Table
MIP	Mobile IP
NAT	Network Address Translator
NDN	Named Data Networking
NRS	Name Resolution System
OpEx	Operational Expenditures
OSPF	Open Shortest Path First
P2P	Peer-to-Peer
PIT	Pending Interests Table
PLA	Packet Level Authentication
POINT	iP Over IcN the betTer IP
PSI	Publish-Subscribe Internetworking
PSIRP	Publish-Subscribe Internet Routing Paradigm
Pub-Sub	Publish-Subscribe

PURSUIT	PUblish-SUbscribe Internet Technology
QoE	Quality-of-Experience
QoS	Quality-of-Service
RENE	Rendezvous Network
RF	Radio-Frequency
RH	Resolution Handler
RID	Rendevous ID
RN	Randezvous Node
ROFL	Routing on Flat Labels
RV	RandezVous
RVP	RandezVous Point
SID	Scope ID
SIP	Session Initiation Protocol
TCP	Transmission Control Protocol
TIMIP	Terminal Independent Mobile IP
TM	Topology Manager
TTL	Time To Live
UMTS	Universal Mobile Telecommunications System
URL	Uniform Resource Locator
VC	Video Catalogue
VDSL	Very-high-bitrate Digital Subscriber Line
VPN	Virtual Private Network
WiMax	Worldwide Interoperability for Microwave Access
WLAN	Wireless Local Area Network
WWW	World Wide Web

Bibliography

[1] J. M. McQuillan and D. C. Walden, "The ARPA Network Design Decisions," *Computer Networks*, vol. 1, pp. 243–289, 1977. [Online]. Available: http://dx.doi.org/10.1016/0376-5075(77)90014-9

[2] V. Jacobson, "If a Clean Slate is the solution what was the problem," in *Stanford Clean Slate Seminar.* Stanford, 2006.

[3] R. Braden, "Requirements for Internet Hosts – Communication Layers," IETF, Tech. Rep., 1989., Tech. Rep., 10 1989, RFC 883. [Online]. Available: https://tools.ietf.org/html/rfc1122

[4] Cisco, Cisco Visual Networking Index, "Global mobile data traffic forecast update, 2016–2020," Jun. 2016. [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.pdf

[5] A. Rowstron and P. Druschel, "Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems," in *Proceedings of the Middleware 2001.* Springer, 2001, pp. 329–350.

[6] Google, "We knew the web was big, july 2008," 2008. [Online]. Available: http://googleblog.blogspot.com/2008/07/we-knew-web-was-big.html

[7] M. D'Ambrosio, C. Dannewitz, H. Karl, and V. Vercellone, "MDHT: a hierarchical name resolution service for Information-Centric Networks," in *Proc. of the 2011 ACM SIGCOMM workshop on Information-centric networking (ICN'11).* New York, NY, USA: ACM, 2011, pp. 7–12.

[8] J. H. Saltzer, D. P. Reed, and D. D. Clark, "End-to-end arguments in system design," *ACM Transactions on Computer Systems (TOCS)*, vol. 2, no. 4, pp. 277–288, 1984.

[9] M. Al-Khalidi, N. Thomos, M. J. Reed, M. F. Al-Naday, and D. Trossen, "Network controlled mobility management using IP over ICN architecture," *CoRR*, vol. abs/1610.09011, 2016. [Online]. Available: http://arxiv.org/abs/1610.09011

[10] M. Chetty, J.-Y. Sung, and R. E. Grinter, *How smart homes learn: The evolution of the networked home and household.* Springer, 2007.

[11] R. E. Grinter, W. K. Edwards, M. W. Newman, and N. Ducheneaut, "The work to make a home network work," in *Proceedings of the ECSCW 2005.* Springer, 2005, pp. 469–488.

[12] P. Mockapetris, "DOMAIN NAMES - CONCEPTS and FACILITIES," IETF, Tech. Rep., 1983., Tech. Rep., 11 1983, rFC 882.

[13] ——, "DOMAIN NAMES - IMPLEMENTATION and SPECIFICATION," IETF, Tech. Rep., 1983., Tech. Rep., 11 1983, rFC 883.

[14] A. Ghodsi, S. Shenker, T. Koponen, A. Singla, B. Raghavan, and J. Wilcox, "Information-centric networking: seeing the forest for the trees," in *Proceedings of the 10th ACM Workshop on Hot Topics in Networks.* ACM, 2011, p. 1.

[15] "ISO/IEC 23009-1:2014 Dynamic adaptive streaming over HTTP (DASH)," http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=65274, 2014, online; accessed November 20$^{\text{th}}$, 2016.

[16] B. Cohen, "Incentives build robustness in bittorrent," in *Proceedings of the Workshop on Economics of Peer-to-Peer systems*, vol. 6, 2003, pp. 68–72.

[17] "W3 Catalog," 2016. [Online]. Available: http://scg.unibe.ch/archive/software/w3catalog/

[18] L. Page, "Method for node ranking in a linked database," Sep. 4 2001, uS Patent 6,285,999.

[19] A. Sharma, A. Venkataramani, and R. K. Sitaraman, "Distributing content simplifies isp traffic engineering," in *Proceedings of the ACM SIGMETRICS Performance Evaluation Review*, vol. 41, no. 1. ACM, 2013, pp. 229–242.

[20] E. Nygren, R. K. Sitaraman, and J. Sun, "The akamai network: a platform for high-performance Internet applications," *ACM SIGOPS Operating Systems Review*, vol. 44, no. 3, pp. 2–19, 2010.

[21] "Limelight Networks," 2016. [Online]. Available: https://www.limelight.com/

[22] "Amazon CloudFront," 2016. [Online]. Available: https://aws.amazon.com/cloudfront/

[23] "Napster," 2016. [Online]. Available: http://www.napster.com

[24] "eMule," 2016. [Online]. Available: http://www.emule-project.net/

[25] M. Ripeanu, "Peer-to-peer architecture case study: Gnutella network," in *Proceedings of the Peer-to-Peer Computing, 2001. Proceedings. First International Conference on.* IEEE, 2001, pp. 99–100.

[26] H. Schulze and K. Mochalski, "Ipoque Internet study 2008/2009," *Leipzig, Germany*, 2009.

[27] S. Seetharaman and M. Ammar, "Inter-domain policy violations in multi-hop overlay routes: Analysis and mitigation," *Computer Networks*, vol. 53, no. 1, pp. 60–80, 2009.

[28] P. Mell and T. Grance, "The NIST definition of Cloud Computing," *Computer Security Division*, 2011.

[29] J. L. García-Dorado, A. Finamore, M. Mellia, M. Meo, and M. Munafo, "Characterization of ISP traffic: Trends, user habits, and access technology impact," *Network and Service Management, IEEE Transactions on*, vol. 9, no. 2, pp. 142–155, June 2012.

[30] "Facebook," 2016. [Online]. Available: https://www.facebook.com/

[31] "Twitter," 2016. [Online]. Available: https://twitter.com/

[32] A. Finamore, M. Mellia, M. M. Munafò, R. Torres, and S. G. Rao, "YouTube everywhere: Impact of device and infrastructure synergies on user experience," in *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*. ACM, 2011, pp. 345–360.

[33] R. Torres, A. Finamore, J. R. Kim, M. Mellia, M. M. Munafo, and S. Rao, "Dissecting video server selection strategies in the youtube cdn," in *Proceedings of the 2011 31st International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2011, pp. 248–257.

[34] "YouTube," 2016. [Online]. Available: http://www.youtube.com

[35] "Netflix," 2016. [Online]. Available: https://www.netflix.com

[36] G. Tyson, Y. Elkhatib, N. Sastry, and S. Uhlig, "Demystifying porn 2.0: A look into a major adult video streaming website," in *Proceedings of the 2013 conference on Internet measurement conference*. ACM, 2013, pp. 417–426.

[37] E. Dahlman, G. Mildh, S. Parkvall, J. Peisa, J. Sachs, and Y. Selén, "5G radio access," *Ericsson Review*, vol. 6, pp. 2–7, 2014.

[38] C. Perkins, "RFC 5944: IP Mobility Support for IPv4," IETF, Tech. Rep., Tech. Rep., 2010. [Online]. Available: http://www.ietf.org/rfc/rfc5944.txt

[39] D. Johnson, C. Perkins, and J. Arkko, "RFC 3775: Mobility support in IPv6," Network Working Group, Tech. Rep., 2004. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3775.txt

[40] H. Soliman, L. Bellier, and K. E. Malki, "RFC 5380: Hierarchical mobile IPv6 mobility management (HMIPv6)," 2008. [Online]. Available: https://tools.ietf.org/html/rfc5380

[41] K. Katsaros, N. Fotiou, G. Polyzos, and G. Xylomenos, "Overlay multicast assisted mobility for future publish/subscribe networks," *Proc. ICT Mobile Summit*, 2009.

[42] R. Koodli, "Fast handovers for mobile ipv6," 2005.

[43] L. Song, D. Kotz, R. Jain, and X. He, "Evaluating Next-Cell Predictors with Extensive Wi-Fi Mobility Data," *IEEE Transactions on Mobile Computing*, vol. 5, no. 12, pp. 1633–1649, 2006. [Online]. Available: http://dx.doi.org/10.1109/TMC.2006.185

[44] "HTTP Streaming Architecture," https://developer.apple.com/library/content/documentation/NetworkingInternet/Conceptual/StreamingMediaGuide/Introduction/Introduction.html, 2016, online; accessed November 20$^{\text{th}}$, 2016.

[45] K. Evensen, A. Petlund, H. Riiser, P. Vigmostad, D. Kaspar, C. Griwodz, and P. Halvorsen, "Mobile video streaming using location-based network prediction and transparent handover," in *Network and Operating System Support for Digital Audio and Video, 21st International Workshop, NOSSDAV 2011, Vancouver, BC, Canada, June 1-3, 2011, Proceedings*, 2011, pp. 21–26. [Online]. Available: http://doi.acm.org/10.1145/1989240.1989248

[46] K. V. Katsaros, N. Fotiou, G. C. Polyzos, and G. Xylomenos, "Supporting Mobile streaming services in future publish/subscribe networks," in *2009 Wireless Telecommunications Symposium, WTS 2009, Prague, Czech Republic, April 22-24, 2009*, 2009, pp. 1–7. [Online]. Available: http://dx.doi.org/10.1109/WTS.2009.5068987

[47] N. Fotiou, K. V. Katsaros, G. C. Polyzos, M. Särelä, D. Trossen, and G. Xylomenos, "Handling mobility in future publish-subscribe information-centric networks," *Telecommunication Systems*, vol. 53, no. 3, pp. 299–314, 2013. [Online]. Available: http://dx.doi.org/10.1007/s11235-013-9699-9

[48] X. Vasilakos, V. A. Siris, and G. C. Polyzos, "Towards exploiting User-Centric Information for Proactive Caching in Mobile Networks," in *Proceedings of the 28th World Wireless Research Forum (WWRF28), Athens, Greece.* World Wireless Research Forum, 2012.

[49] X. Vasilakos and V. A. Siris, "Adapting data popularity in mobility-based proactive caching decisions for heterogeneous wireless networks," in *10th International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness (QShine)*, Aug 2014, pp. 191–192. [Online]. Available: http://dx.doi.org/10.1109/QSHINE.2014.6928687

[50] H. Falaki, R. Mahajan, S. Kandula, D. Lymberopoulos, R. Govindan, and D. Estrin, "Diversity in smartphone usage," in *Proceedings of the 8th international conference on Mobile systems, applications, and services.* ACM, 2010, pp. 179–194.

[51] I. Trestian, S. Ranjan, A. Kuzmanovic, and A. Nucci, "Measuring serendipity: connecting people, locations and interests in a mobile 3G network," in *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference.* ACM, 2009, pp. 267–279.

[52] P. Deshpande, A. Kashyap, C. Sung, and S. R. Das, "Predictive methods for improved vehicular Wi-Fi access," in *Proceedings of the 7th international conference on Mobile systems, applications, and services*. ACM, 2009, pp. 263–276.

[53] A. J. Nicholson and B. D. Noble, "Breadcrumbs: forecasting mobile connectivity," in *Proceedings of the 14th Annual International Conference on Mobile Computing and Networking, MOBICOM 2008, San Francisco, California, USA, September 14-19, 2008*, 2008, pp. 46–57. [Online]. Available: http://doi.acm.org/10.1145/1409944.1409952

[54] F. Zhang, C. Xu, Y. Zhang, K. Ramakrishnan, S. Mukherjee, R. Yates, and T. Nguyen, "EdgeBuffer: Caching and prefetching content at the edge in the MobilityFirst Future Internet architecture," in *Proceedings of the 2015 IEEE 16th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, June 2015, pp. 1–9.

[55] X. Vasilakos, V. A. Siris, and G. C. Polyzos, "Addressing Niche Demand based on joint Mobility Prediction and Content Popularity caching," *Computer Networks*, vol. 110, pp. 306 – 323, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128616303334

[56] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and Zipf-like distributions: Evidence and implications," in *Proceedings of the 18th Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 1999*, vol. 1, Mar 1999, pp. 126–134 vol.1.

[57] A. Wolman, M. Voelker, N. Sharma, N. Cardwell, A. Karlin, and H. M. Levy, "On the scale and performance of cooperative web proxy caching," in *Proceedings of the ACM SIGOPS Operating Systems Review*, vol. 33. ACM, 1999, pp. 16–31.

[58] P. Cao, J. Zhang, and K. Beach, "Active cache: caching dynamic contents on the web," *Distributed Systems Engineering*, vol. 6, no. 1, pp. 43–50, mar 1999.

[59] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon, "I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system," in *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*. ACM, 2007, pp. 1–14.

[60] ——, "Analyzing the video popularity characteristics of large-scale user generated content systems," *IEEE/ACM Transactions on Networking (TON)*, vol. 17, no. 5, pp. 1357–1370, 2009.

[61] H. Yu, D. Zheng, B. Y. Zhao, and W. Zheng, "Understanding user behavior in large-scale video-on-demand systems," in *Proceedings of the ACM SIGOPS Operating Systems Review*, no. 4. ACM, 2006, pp. 333–344.

[62] "CAIDA: Cooperative Association for Internet Data Analysis," 2016. [Online]. Available: http://www.caida.org

[63] P. T. Eugster, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec, "The many faces of Publish/Subscribe," *ACM Computing Surveys (CSUR)*, vol. 35, no. 2, pp. 114–131, 2003.

[64] D. Trossen and G. Parisis, "Designing and realizing an Information-Centric Internet," *IEEE Communications Magazine*, vol. 50, no. 7, pp. 60–67, 2012.

[65] A. Ghodsi, T. Koponen, J. Rajahalme, P. Sarolahti, and S. Shenker, "Naming in content-oriented architectures," in *Proc. of the ACM SIGCOMM ICN workshop*, New York, NY, USA, 2011, pp. 1–6.

[66] D. Smetters and V. Jacobson, "Securing network content," PARC, Tech. Rep. TR-2009-01, October 2009.

[67] C. Dannewitz, M. D'Ambrosio, and V. Vercellone, "Hierarchical DHT-based name resolution for Information-Centric Networks," *Elsevier Computer Communications*, vol. 36, no. 7, pp. 736–749, 2013.

[68] A. Narayanan and D. Oran, "NDN and IP Routing: Can it scale?" *Proposed Information-Centric Networking Research Group (ICNRG), Side meeting at IETF-82, Taipei*, November 2011.

[69] H. Liu, X. De Foy, and D. Zhang, "A multi-level dht routing framework with aggregation," in *Proceedings of the second edition of the ICN workshop on Information-centric networking*. ACM, 2012, pp. 43–48.

[70] G. Xylomenos, C. N. Ververidis, V. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros, G. C. Polyzos *et al.*, "A survey of Information-Centric Networking research," *Communications Surveys & Tutorials, IEEE*, vol. 16, no. 2, pp. 1024–1049, 2014.

[71] I. Stoica, R. Morris, D. Liben-Nowell, D. R. Karger, M. F. Kaashoek, F. Dabek, and H. Balakrishnan, "Chord: a scalable peer-to-peer lookup protocol for Internet applications," *IEEE/ACM Transactions on Networking*, vol. 11, no. 1, pp. 17–32, 2003.

[72] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica, "A data-oriented (and beyond) network architecture," in *Proc. of the 2007 ACM SIGCOMM*. New York, NY, USA: ACM, 2007, pp. 181–192.

[73] W. K. Chai, N. Wang, I. Psaras, G. Pavlou, C. Wang, G. G. De Blas, F. J. Ramon-Salguero, L. Liang, S. Spirou, A. Beben *et al.*, "Curling: Content-ubiquitous resolution and delivery infrastructure for next-generation services," *IEEE Communications Magazine*, vol. 49, no. 3, pp. 112–120, 2011.

[74] G. Xylomenos, X. Vasilakos, C. Tsilopoulos, V. A. Siris, and G. C. Polyzos, "Caching and Mobility Support in a Publish-Subscribe Internet architecture," *Communications Magazine, IEEE*, vol. 50, no. 7, pp. 52–58, 2012.

[75] K. Keeton, D. Ferrari, B. A. Mah, S. Seshan, and R. H. Katz, "Providing Connection-Oriented Network Services to Mobile Hosts." in *Proceedings of the Symposium on Mobile and Location-Independent Computing*, 1993.

[76] A. Caglayan and C. Harrison, "Agent sourcebook," 2011.

[77] V. Giannaki, X. Vasilakos, C. Stais, G. C. Polyzos, and G. Xylomenos, "Supporting mobility in a publish subscribe internetwork architecture," in *Computers and Communications (ISCC), 2011 IEEE Symposium on*, June 2011, pp. 1030–1032.

[78] K. Katsaros, G. Xylomenos, and G. C. Polyzos, "MultiCache: An overlay architecture for information-centric networking," *Computer Networks*, vol. 55, no. 4, pp. 936 – 947, 2011, special Issue on Architectures and Protocols for the Future Internet. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128610003798

[79] D. Trossen, X. Vasilakos, P. Flegkas, V. Sourlas, and G. Parisis, "Mobility work re-visited not considered harmful," in *2011 3rd International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, Oct 2011, pp. 1–8.

[80] D. Lagutin, "Redesigning Internet–the packet level authentication architecture," *Licentiate's Thesis-Helsinki University of Technology*, 2008.

[81] P. Jokela, A. Zahemszky, C. Esteve Rothenberg, S. Arianfar, and P. Nikander, "LIPSIN: line speed Publish/Subscribe inter-networking," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 4, pp. 195–206, 2009.

[82] N. Fotiou, P. Nikander, D. Trossen, and G. Polyzos, "Developing information networking further: From psirp to pursuit," in *Proceedings of the 7^{th} ICST Int'l Conf. on Broadband Commun., Networks, and Systems*, 2010.

[83] FP7 PURSUIT project. [Online]. Available: http://www.fp7-pursuit.eu/PursuitWeb/

[84] FP7 PSIRP project. [Online]. Available: http://www.psirp.org/

[85] The POINT project. [Online]. Available: https://www.point-h2020.eu/

[86] D. Trossen, G. Parisis, K. Visala, B. Gajic, J. Riihijarvi, P. Flegkas, P. Sarolahti, P. Jokela, X. Vasilakos, C. Tsilopoulos, and S. Arianfar, "Conceptual Architecture: Principles, patterns and sub-components descriptions," *Deliverable D2*, vol. 2, May 2011. [Online]. Available: http://www.fp7-pursuit.eu/PursuitWeb/

[87] D. Trossen, G. Parisis, B. Gajic, J. Riihijarvi, P. Flegkas, P. Sarolahti, P. Jokela, X. Vasilakos, C. Tsilopoulos, and S. Arianfar, "Architecture Definition, Components Descriptions and Requirements," *Deliverable D2*, vol. 3, 2009.

[88] D. Trossen, B. Tagger, G. Parisis, M. Skjegstad, N. Fotiou, X. Vasilakos, C. Stais, M. Thomas, Yiannis Al-Naday, M. Reed, V. Vassilakis, A. Bontozoglou, B. Gajic, P. Sarolahti, P. Flegkas, and V. Sourlas, "Final Updated Architecture," *Deliverable D2*, vol. 5, March 2013.

[89] K. V. Katsaros, N. Fotiou, X. Vasilakos, C. N. Ververidis, C. Tsilopoulos, G. Xylomenos, and G. C. Polyzos, “On inter-domain name resolution for Information-Centric Networks,” in *Proc. of the IFIP TC 6 Conference on Networking.* Berlin, Heidelberg: Springer-Verlag, 2012, pp. 13–26.

[90] R. Jarno, M. Särelä, K. Visala, and J. Riihijärvi, “On name-based inter-domain routing,” *Elsevier Computer Networks*, vol. 55, pp. 975–986, 2011.

[91] B. H. Bloom, “Space/time trade-offs in hash coding with allowable errors,” *Communications of the ACM*, vol. 13, no. 7, pp. 422–426, July 1970.

[92] V. Sourlas, P. Flegkas, L. Gkatzikis, and L. Tassiulas, “Autonomic cache management in Information-Centric Networks,” in *Proceedings of the 2012 IEEE Network Operations and Management Symposium*, April 2012, pp. 121–129.

[93] V. A. Siris, X. Vasilakos, and G. C. Polyzos, “A Selective Neighbor Caching Approach for Supporting Mobility in Publish/Subscribe Networks,” in *Proc. of ERCIM Workshop on eMobility. Held in conjunction with WWIC*, 2011, p. 63.

[94] X. Vasilakos, V. A. Siris, G. C. Polyzos, and M. Pomonis, “Proactive Selective Neighbor Caching for enhancing Mobility Support in Information-Centric Networks,” in *Proceedings of the 2^{nd} edition of the ICN workshop on Information-Centric Networking.* ACM, 2012, pp. 61–66.

[95] V. Sourlas, P. Flegkas, G. S. Paschos, D. Katsaros, and L. Tassiulas, “Storage planning and replica assignment in Content-Centric Publish/Subscribe networks,” *Computer Networks*, vol. 55, no. 18, pp. 4021–4032, 2011.

[96] N. Fotiou, K. Katsaros, G. C. Polyzos, M. Särelä, D. Trossen, and G. Xylomenos, “Handling mobility in future Publish-Subscribe Information-Centric Networks,” *Telecommunication Systems*, vol. 53, no. 3, pp. 299–314, 2013.

[97] V. A. Siris, X. Vasilakos, and D. Dimopoulos, “Exploiting mobility prediction for mobility & popularity caching and DASH adaptation,” in *Proceedings of the IEEE 17^{th} International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, June 2016, pp. 1–8. [Online]. Available: http://dx.doi.org/10.1109/WoWMoM.2016.7523523

[98] V. A. Siris, X. Vasilakos, and G. C. Polyzos, “Efficient Proactive Caching for supporting Seamless Mobility,” in *Proceedings of the 15^{th} IEEE International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, June 2014, pp. 1–6.

[99] NSF Named Data Networking (NDN) project. [Online]. Available: http://www.named-data.net/

[100] Content Centric Networking (CCN) project. [Online]. Available: http://blogs.parc.com/ccnx/

[101] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proceedings of the 5th international conference on Emerging networking experiments and technologies.* ACM, 2009, pp. 1–12.

[102] M. Meisel, V. Pappas, and L. Zhang, "Ad hoc networking via named data," in *Proceedings of the ACM MobiArch*, 2010.

[103] D. Smetters and V. Jacobson, "Securing network content," PARC, Tech. Rep. TR-2009-01, October 2009.

[104] F. Bronzino, K. Nagaraja, I. Seskar, and D. Raychaudhuri, "Network service abstractions for a mobility-centric future Internet architecture," in *Proceedings of the 8th ACM international workshop on Mobility in the evolving Internet architecture.* ACM, 2013, pp. 5–10.

[105] A. Baid, T.Vu, and D. Raychaudhuri, "Comparing alternative approaches for networking of named objects in the future Internet," in *Proceedings of the IEEE Workshop on Emerging Design Choices in Name-Oriented Networking (NOMEN)*, 2012.

[106] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana, "Internet indirection infrastructure," in *Proceedings of the ACM SIGCOMM*, 2002, pp. 73–86.

[107] M. Caesar, T. Condie, J. Kannan, K. Lakshminarayanan, and I. Stoica, "ROFL: routing on flat labels," *SIGCOMM Computer Communications Review*, vol. 36, no. 4, pp. 363–374, 2006.

[108] P. Ganesan, K. Gummadi, and H. Garcia-Molina, "Canon in G Major: Designing DHTs with Hierarchical Structure," in *Proc. of the 2004 ICDCS*, 2004, pp. 263–272.

[109] V. Almeida, A. Bestavros, M. Crovella, and A. De Oliveira, "Characterizing reference locality in the www," in *Parallel and Distributed Information Systems, 1996., Fourth International Conference on.* IEEE, 1996, pp. 92–103.

[110] A. Carzaniga and A. L. Wolf, "Content-based networking: A new communication infrastructure," in *Proceedings of the Workshop on Infrastruture for Mobile and Wireless Systems.* Springer, 2001, pp. 59–68.

[111] G. Cugola, E. Di Nitto, and A. Fuggetta, "The jedi event-based infrastructure and its application to the development of the opss wfms," *IEEE transactions on Software Engineering*, vol. 27, no. 9, pp. 827–850, 2001.

[112] J. E. Smith and W.-C. Hsu, "Prefetching in supercomputer instruction caches," in *Proceedings of the 1992 ACM/IEEE Conference on Supercomputing*, ser. Supercomputing '92. Los Alamitos, CA, USA: IEEE Computer Society Press, 1992, pp. 588–597. [Online]. Available: http://dl.acm.org/citation.cfm?id=147877.148093

[113] V. N. Padmanabhan and J. C. Mogul, "Using predictive prefetching to improve world wide web latency," *ACM SIGCOMM Computer Communication Review*, vol. 26, no. 3, pp. 22–36, 1996.

[114] E. P. Markatos, C. E. Chronaki *et al.*, "A top-10 approach to prefetching on the web," in *Proceedings of INET*, vol. 98, 1998, pp. 276–290.

[115] D. Duchamp, "Prefetching Hyperlinks," in *2nd USENIX Symposium on Internet Technologies and Systems, USITS'99, Boulder, Colorado, USA, October 11-14, 1999*, 1999. [Online]. Available: http://www.usenix.org/publications/library/proceedings/usits99/duchamp.html

[116] J. Xu, J. Liu, B. Li, and X. Jia, "Caching and prefetching for web content distribution," *Computing in Science Engineering*, vol. 6, no. 4, pp. 54–59, July 2004.

[117] E. Cohen and H. Kaplan, "Proactive caching of dns records: Addressing a performance bottleneck," *Computer Networks*, vol. 41, no. 6, pp. 707–726, 2003.

[118] H. Hu, J. Xu, W. S. Wong, B. Zheng, D. L. Lee, and W.-C. Lee, "Proactive caching for spatial queries in mobile environments," in *Proceedings of the 21st International Conference on Data Engineering (ICDE 2005)*. IEEE, 2005, pp. 403–414.

[119] A. Gaddah and T. Kunz, "Extending mobility to publish/subscribe systems using a pro-active caching approach," *Mobile Information Systems*, vol. 6, no. 4, pp. 293–324, 2010. [Online]. Available: http://dx.doi.org/10.3233/MIS-2010-0105

[120] I. Burcea, H. Jacobsen, E. de Lara, V. Muthusamy, and M. Petrovic, "Disconnected Operation in Publish/Subscribe Middleware," in *5th IEEE International Conference on Mobile Data Management (MDM 2004), 19-22 January 2004, Berkeley, CA, USA*, 2004, p. 39. [Online]. Available: http://dx.doi.org/10.1109/MDM.2004.1263041

[121] M. Caporuscio, A. Carzaniga, and A. L. Wolf, "Design and Evaluation of a Support Service for Mobile Publish/Subscribe Applications," *IEEE Transactions on Software Engineering*, vol. 29, no. 12, pp. 1059–1071, December 2003.

[122] L. Fiege, F. C. Gärtner, O. Kasten, and A. Zeidler, "Supporting Mobility in Content-Based Publish/Subscribe Middleware," in *Middleware 2003, ACM/IFIP/USENIX International Middleware Conference, Rio de Janeiro, Brazil, June 16-20, 2003, Proceedings*, 2003, pp. 103–122. [Online]. Available: http://dx.doi.org/10.1007/3-540-44892-6_6

[123] V. Sourlas, G. S. Paschos, P. Flegkas, and L. Tassiulas, "Mobility Support Through Caching in Content-Based Publish/Subscribe Networks," in *10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, CCGrid 2010, 17-20 May 2010, Melbourne, Victoria, Australia*, 2010, pp. 715–720. [Online]. Available: http://dx.doi.org/10.1109/CCGRID.2010.22

[124] J. Wang, J. Cao, J. Li, and J. Wu, "MHH: A Novel Protocol for Mobility Management in Publish/Subscribe Systems," in *Proc. of IEEE Int'l Conference on Parallel Processing (ICPP)*, 2007, p. 54.

[125] U. Farooq, E. Parsons, and S. Majumdar, "Performance of Publish/Subscribe Middleware in Mobile Wireless Networks," in *Proc. of International Workshop on Software and Performance (WOSP)*, 2004, pp. 295–305.

[126] A. Mishra, M.-h. Shin, and W. Arbaush, "Context caching using neighbor graphs for fast handoffs in a wireless network," in *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 2004*, vol. 1. IEEE, 2004.

[127] S. Pack, H. Jung, T. Kwon, and Y. Choi, "SNC: A Selective Neighbor Caching Scheme for Fast Handoff in IEEE 802.11 Wireless Networks," *ACM Mobile Computing and Communications Review*, vol. 9, no. 4, pp. 39–49, October 2005.

[128] M.-H. Chiu and M. A. Bassiouni, "Predictive Schemes for Handoff Prioritization in Cellular Networks Based on Mobile Positioning," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 510–522, March 2000.

[129] W.-S. Soh and H. S. Kim, "QoS provisioning in cellular networks based on mobility prediction techniques," *IEEE Communications Magazine*, vol. 41, no. 1, pp. 86–92, January 2003.

[130] S. K. Dandapat, S. Pradhan, N. Ganguly, and R. Roy Choudhury, "Sprinkler: distributed content storage for just-in-time streaming," in *Proceeding of the 2013 workshop on Cellular networks: operations, challenges, and future design*. ACM, 2013, pp. 19–24.

[131] M. H. Cheung and J. Huang, "Optimal Delayed Wi-Fi Offloading," in *Proceedings of the 11th International Symposium on Modeling Optimization in Mobile, Ad Hoc Wireless Networks (WiOpt)*.

[132] E. Baştuğ, J.-L. Guénégo, and M. Debbah, "Proactive small cell networks," in *Proceedings of the 2013 20th International Conference on Telecommunications (ICT)*, May 2013, pp. 1–5.

[133] E. Baştuğ, M. Bennis, and M. Debbah, "Living on the edge: The role of proactive caching in 5G wireless networks," *Communications Magazine, IEEE*, vol. 52, no. 8, pp. 82–89, Aug 2014.

[134] E. Bastug, M. Bennis, and M. Debbah, "Proactive caching in 5G small cell networks," 2015.

[135] S. Jin and A. Bestavros, "Popularity-aware greedy dual-size web proxy caching algorithms," in *Proceedings of the 20th international conference on Distributed computing systems*. IEEE, 2000, pp. 254–261.

[136] ——, "Greedydual web caching algorithm: exploiting the two sources of temporal locality in web request streams," *Computer Communications*, vol. 24, no. 2, pp. 174–183, 2001.

[137] D. Applegate, A. Archer, V. Gopalakrishnan, S. Lee, and K. K. Ramakrishnan, "Optimal content placement for a large-scale vod system," vol. 24, no. 4, 2016, pp. 2114–2127. [Online]. Available: http://dx.doi.org/10.1109/TNET.2015.2461599

[138] V. Gonçalves, N. Walravens, and P. Ballon, ""how about an app store?" enablers and constraints in platform strategies for mobile network operators," in *Proceedings of the 9th International Conference on Mobile Business and 2010 Ninth Global Mobility Roundtable (ICMB-GMR).* IEEE, 2010, pp. 66–73.

[139] F. Malandrino, M. Kurant, A. Markopoulou, C. Westphal, and U. C. Kozat, "Proactive seeding for information cascades in cellular networks," in *INFOCOM, 2012 Proceedings IEEE.* IEEE, 2012, pp. 1719–1727.

[140] A. C. Güngör and D. Gündüz, "Proactive wireless caching at mobile user devices for energy efficiency," in *2015 International Symposium on Wireless Communication Systems (ISWCS)*, Aug 2015, pp. 186–190.

[141] E. Bastug, K. Hamidouche, W. Saad, and M. Debbah, "Centrality-based caching for mobile wireless networks," in *Proceedings of the 1st KuVS Workshop on Anticipatory Networks*, 2014.

[142] K. Hamidouche, W. Saad, and M. Debbah, "Many-to-many matching games for proactive social-caching in wireless small cell networks," in *Proceedings of the 12th International Symposium on Modelling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt).* IEEE, 2014, pp. 569–574.

[143] N. Golrezaei, K. Shanmugam, A. Dimakis, A. Molisch, and G. Caire, "Femtocaching: Wireless video content delivery through distributed caching helpers," in *INFOCOM, 2012 Proceedings IEEE*, March 2012, pp. 1107–1115.

[144] N. Golrezaei, A. Molisch, A. Dimakis, and G. Caire, "Femtocaching and device-to-device collaboration: A new architecture for wireless video distribution," *Communications Magazine, IEEE*, vol. 51, no. 4, pp. 142–149, April 2013.

[145] I. Baev, R. Rajaraman, and C. Swamy, "Approximation algorithms for data placement problems," *SIAM Journal on Computing*, vol. 38, no. 4, pp. 1411–1429, 2008.

[146] I. D. Baev and R. Rajaraman, "Approximation algorithms for data placement in arbitrary networks," in *Proceedings of the 12th annual ACM-SIAM symposium on Discrete algorithms.* Society for Industrial and Applied Mathematics, 2001, pp. 661–670.

[147] Y. Guan, Y. Xiao, H. Feng, C. Shen, and L. J. C. Jr., "Mobicacher: Mobility-aware content caching in small-cell networks," in *IEEE Global Communications Conference, GLOBECOM 2014, Austin, TX, USA, December 8-12, 2014*, 2014, pp. 4537–4542. [Online]. Available: http://dx.doi.org/10.1109/GLOCOM.2014.7037523

[148] A. S. Gomes, T. Braun, and E. Monteiro, "Enhanced caching strategies at the edge of LTE mobile networks," in *2016 IFIP Networking Conference, Networking*

2016 and Workshops, Vienna, Austria, May 17-19, 2016, 2016, pp. 341–349. [Online]. Available: http://dx.doi.org/10.1109/IFIPNetworking.2016.7497245

[149] A. S. Gomes, B. Sousa, D. Palma, V. Fonseca, Z. Zhao, E. Monteiro, T. Braun, P. Simoes, and L. Cordeiro, "Edge caching with mobility prediction in virtualized lte mobile networks," *Future Generation Computer Systems*, vol. 70, pp. 148–162, 2017.

[150] K. Poularakis, G. Iosifidis, V. Sourlas, and L. Tassiulas, "Multicast-aware caching for small cell networks," in *2014 IEEE Wireless Communications and Networking Conference (WCNC)*, April 2014, pp. 2300–2305.

[151] K. Poularakis and L. Tassiulas, "Code, cache and deliver on the move: A novel caching paradigm in hyper-dense small-cell networks," *IEEE Transactions on Mobile Computing*, vol. PP, no. 99, pp. 1–1, 2016.

[152] S. Borst, V. Gupta, and A. Walid, "Distributed Caching Algorithms for Content Distribution Networks," in *INFOCOM, 2010 Proceedings IEEE*, March 2010, pp. 1–9.

[153] S. T. ul Hassan, M. Bennis, P. H. J. Nardelli, and M. Latva-aho, "Caching in wireless small cell networks: A storage-bandwidth tradeoff," *IEEE Communications Letters*, vol. 20, no. 6, pp. 1175–1178, June 2016.

[154] K. Kanai, T. Muto, J. Katto, S. Yamamura, T. Furutono, T. Saito, H. Mikami, K. Kusachi, T. Tsuda, W. Kameyama, Y. J. Park, and T. Sato, "Proactive content caching for mobile video utilizing transportation systems and evaluation through field experiments," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 8, pp. 2102–2114, Aug 2016.

[155] G. Zheng and V. Friderikos, "Optimal proactive cache management in mobile networks," in *2016 IEEE International Conference on Communications (ICC)*, May 2016, pp. 1–6.

[156] S. Sevilla, P. Mahadevan, and J. Garcia-Luna-Aceves, "iDNS: Enabling information centric networking through The DNS," in *Proc. of the IEEE INFOCOM Workshop on Name Oriented Mobility*, 2014, pp. 476–481.

[157] M. A. Kyle Schomp and M. Rabinovich, "DNS Resolvers Considered Harmful ," in *Proc. of the ACM Workshop on Hot Topics in Networks (Hotnets)*, 2014.

[158] V. Ramasubramanian and E. G. Sirer, "The Design and Implementation of a Next Generation Name Service for the Internet," in *Proc. of the ACM SIGCOMM*, 2004, pp. 331–342.

[159] M. Walfish, H. Balakrishnan, and S. Shenker, "Untangling the web from DNS," in *Proceedings of the 1st Conference on Symposium on Networked Systems Design and Implementation - Volume 1*, ser. NSDI'04. Berkeley, CA, USA: USENIX Association, 2004, pp. 17–17. [Online]. Available: http://dl.acm.org/citation.cfm?id=1251175.1251192

[160] P. Vixie, “What DNS is not.” *Commun. ACM*, vol. 52, no. 12, pp. 43–47, 2009.

[161] N. Fotiou, K. V. Katsaros, X. Vasilakos, C. Tsilopoulos, C. N. Ververidis, G. Xylomenos, and G. C. Polyzos, “H-Pastry: An Adaptive Multi-level Overlay Inter-Network,” Athens University of Economics and Business, Athens, Greece, Tech. Rep. 2011-MMLAB-TR-002, 2011. [Online]. Available: http://mm.aueb.gr/technicalreports/2011-MMLAB-TR-003.pdf

[162] N. J. A. Harvey, M. B. Jones, S. Saroiu, M. Theimer, and A. Wolman, “SkipNet: A Scalable Overlay Network with Practical Locality Properties,” in *Proc. of the USENIX Symposium on Internet Technologies and Systems (USITS)*, 2003, pp. 9–9.

[163] J. Choi, J. Han, E. Cho, H. Kim, T. Kwon, and Y. Choi, “Performance comparison of content-oriented networking alternatives: A tree versus a distributed hash table,” in *Proc. of the IEEE Conference on Local Computer Networks (LCN)*, 2009, pp. 253–256.

[164] S. Varjonen, T. Heer, K. Rimey, and A. V. Gurtov, “Secure Resolution of End-Host Identifiers for Mobile Clients,” in *Proceedings of the Global Communications Conference, GLOBECOM 2011, 5-9 December 2011, Houston, Texas, USA*, 2011, pp. 1–6. [Online]. Available: http://dx.doi.org/10.1109/GLOCOM.2011.6134003

[165] L. Gao, “On inferring autonomous system relationships in the internet,” *IEEE/ACM Transactions on Networking (ToN)*, vol. 9, no. 6, pp. 733–745, 2001.

[166] V. A. Siris, X. Vasilakos, and G. C. Polyzos, “Efficient Proactive Caching for Supporting Seamless Mobility,” vol. abs/1404.4754, 2014. [Online]. Available: http://arxiv.org/abs/1404.4754

[167] N. Carlsson and D. L. Eager, “Ephemeral Content Popularity at the Edge and Implications for On-Demand Caching,” *IEEE Transactions on Parallel and Distributed Systems*, 2016.

[168] B. M. Maggs and R. K. Sitaraman, “Algorithmic Nuggets in Content Delivery,” *Computer Communication Review*, vol. 45, no. 3, pp. 52–66, 2015. [Online]. Available: http://doi.acm.org/10.1145/2805789.2805800

[169] S. Traverso, M. Ahmed, M. Garetto, P. Giaccone, E. Leonardi, and S. Niccolini, “Temporal locality in today’s content caching: why it matters and how to model it,” *Computer Communication Review*, vol. 43, no. 5, pp. 5–12, 2013. [Online]. Available: http://doi.acm.org/10.1145/2541468.2541470

[170] S. Martello and P. Toth, *Knapsack Problems: Algorithms and Computer Implementations*. New York, NY, USA: John Wiley & Sons, Inc., 1990.

[171] R. Andonov, V. Poirriez, and S. V. Rajopadhye, “Unbounded knapsack problem: Dynamic programming revisited,” *European Journal of Operational Research*, vol. 123, no. 2, pp. 394–407, 2000. [Online]. Available: http://dx.doi.org/10.1016/S0377-2217(99)00265-9

[172] M. R. Korupolu, C. G. Plaxton, and R. Rajaraman, "Placement algorithms for hierarchical cooperative caching," *Journal of Algorithms*, vol. 38, no. 1, pp. 260–302, 2001.

[173] X. Dimitropoulos, D. Krioukov, A. Vahdat, and G. Riley, "Graph annotations in modeling complex network topologies," *ACM Transactions on Modeling and CoMuter Simulation*, vol. 19, November 2009.

[174] A. Dhamdhere and C. Dovrolis, "Twelve Years in the Evolution of the Internet Ecosystem," *Networking, IEEE/ACM Transactions on*, vol. 19, no. 5, pp. 1420–1433, Oct 2011.

[175] V. Chandrasekhar, J. Andrews, and A. Gatherer, "Femtocell networks: a survey," *Communications Magazine, IEEE*, vol. 46, no. 9, pp. 59–67, September 2008.

[176] J. Märkendahl and O. Mäkitalo, "A comparative study of deployment options, capacity and cost structure for macrocellular and femtocell networks," in *Proceedings of the Personal, Indoor and Mobile Radio Communications Workshops (PIMRC Workshops), 2010 IEEE 21st International Symposium on, year=2010*, Sept, pp. 145–150.

[177] Cisco, Cisco Visual Networking Index, "Global mobile data traffic forecast update, 2013–2018," Feb. 2014.

[178] Intel Corporation, "Rethinking the Small Cell Business Model," 2012. [Online]. Available: http://www.intel.com/content/dam/www/public/us/en/documents/white-papers/communications-small-cell-study.pdf

[179] ALTOBRIDGE, "Data-at-the-edge (date)," 2014. [Online]. Available: http://www.altobridge.com/data-at-the-edge

[180] Wireless LAN Working Group for 802.11, "IEEE Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," 2012. [Online]. Available: standards.ieee.org/getieee802/download/802.11-2012.pdf

[181] QUALCOMM Incorporated, "IEEE802.11ac: The next evolution of Wi-Fi™ standards," 2012. [Online]. Available: http://www.qualcomm.com/media/documents/ieee80211ac-next-evolution-wi-fi

[182] J. Jun, P. Peddabachagari, and M. Sichitiu, "Theoretical maximum throughput of IEEE 802.11 and its applications," in *Proceedings of the Network Computing and Applications, 2003. NCA 2003. Second IEEE International Symposium on*, April 2003, pp. 249–256.

[183] K. Katsaros, G. Xylomenos, and G. Polyzos, "GlobeTraff: a traffic workload generator for the performance evaluation of Future Internet architectures," in *Proceedings of the New Technologies, Mobility and Security (NTMS), 2012 5th International Conference on*, May 2012, pp. 1–5.

[184] B. Krishnamurthy, C. Wills, and Y. Zhang, “On the use and performance of content distribution networks,” in *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*. ACM, 2001, pp. 169–182.

[185] D. Beaver, S. Kumar, H. C. Li, J. Sobel, P. Vajgel *et al.*, “Finding a needle in haystack: Facebook’s photo storage.” in *OSDI*, vol. 10, 2010, pp. 1–8.

[186] F. Larumbe and A. Mathur, “Under the hood: Broadcasting live video to millions,” Facebook Code, 2015. [Online]. Available: https://code.facebook.com/posts/1653074404941839/under-the-hood-broadcasting-live-video-to-millions/

[187] comScore, “The U.S. Mobile App Report,” comScore, 2014. [Online]. Available: http://www.comscore.com/Insights/Presentations-and-Whitepapers/2014/The-US-Mobile-App-Report

[188] H. Li, H. Wang, J. Liu, and K. Xu, “Video requests from online social networks: Characterization, analysis and generation,” in *INFOCOM, 2013 Proceedings IEEE*. IEEE, 2013, pp. 50–54.

[189] X. Cheng, C. Dale, and J. Liu, “Statistics and social network of YouTube videos,” in *Proceedings of the 16th International Workshop on Quality of Service (IWQoS 2008)*. IEEE, 2008, pp. 229–238.

[190] M. Zink, K. Suh, Y. Gu, and J. Kurose, “Characteristics of YouTube network traffic at a campus network–measurements, models, and implications,” *Computer Networks*, vol. 53, no. 4, pp. 501–514, 2009.

[191] L. E. Chatzieleftheriou, M. Karaliopoulos, and I. Koutsopoulos, “Caching-aware Recommendations: Nudging User Preferences towards better Caching Performance,” in *INFOCOM, 2017 Proceedings IEEE (to appear)*, 2017.

[192] G. P. Perrucci, F. H. Fitzek, and J. Widmer, “Survey on energy consumption entities on the smartphone platform,” in *Proceedings of the Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd*. IEEE, 2011, pp. 1–6.

[193] M. Paolini, “The economics of small cells and Wi-Fi offload,” 2012.

[194] E. C. Efstathiou, P. A. Frangoudis, and G. C. Polyzos, “Stimulating Participation in Wireless Community Networks,” in *Proceedings of the 2006 IEEE INFOCOM*, 2006, pp. 1–13.

[195] E. Efstathiou, P. Frangoudis, and G. Polyzos, “Controlled Wi-Fi Sharing in Cities: A Decentralized Approach Relying on Indirect Reciprocity,” *Mobile Computing, IEEE Transactions on*, vol. 9, no. 8, pp. 1147–1160, Aug 2010.

[196] P. A. Frangoudis, G. C. Polyzos, and V. P. Kemerlis, “Wireless community networks: an alternative approach for nomadic broadband network access,” *Communications Magazine, IEEE*, vol. 49, no. 5, pp. 206–213, 2011.

[197] P. Kirwan, “Open Access: How Fon wants the world to be a giant Wi-Fi hotspot,” 2011. [Online]. Available: http://www.wired.co.uk/magazine/archive/2011/09/features/open-access-fon

[198] Fon Wireless Ltd, Feb. 2015. [Online]. Available: https://corp.fon.com

[199] L. Gao, G. Iosifidis, J. Huang, and L. Tassiulas, “Economics of mobile data offloading,” in *INFOCOM, 2013 Proceedings IEEE*, April 2013, pp. 3303–3308.

[200] D. Dimopoulos, C. Boursinos, and V. A. Siris, “Multi-Source Mobile Video Streaming: Load Balancing, Fault Tolerance, and Offloading with Prefetching,” *Proceedings of the 9th Intl Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities (Tridentcom)*, 2014.

[201] K. V. Katsaros, V. Sourlas, I. Psaras, S. Rene, and G. Pavlou, “Information-centric connectivity,” *arXiv preprint arXiv:1602.00566*, 2016.

[202] K. Poularakis, G. Iosifidis, A. Argyriou, and L. Tassiulas, “Video delivery over heterogeneous cellular networks: Optimizing cost and performance,” in *2014 IEEE Conference on Computer Communications, INFOCOM 2014, Toronto, Canada, April 27 - May 2, 2014*, 2014, pp. 1078–1086. [Online]. Available: http://dx.doi.org/10.1109/INFOCOM.2014.6848038

[203] X. Cheng, C. Dale, and J. Liu, “Understanding the characteristics of Internet short video sharing: YouTube as a case study,” *arXiv preprint arXiv:0707.3670*, 2007.

[204] “Ircache home.” [Online]. Available: http://ita.ee.lbl.gov/html/traces.html

[205] M. Piorkowski, N. Sarafijanovic-Djukic, and M. Grossglauser, “CRAWDAD dataset epfl/mobility (v. 2009-02-24),” Downloaded from http://crawdad.org/epfl/mobility/20090224, Feb. 2009.

[206] “WiGLE - Wireless Geographic Logging Engine.” [Online]. Available: https://wigle.net/

[207] S. T. Kouyoumdjieva, Ólafur Ragnar Helgason, and G. Karlsson, “CRAWDAD dataset kth/walkers (v. 2014-05-05),” Downloaded from http://crawdad.org/kth/walkers/20140505, May 2014.

[208] D. Lymberopoulos, O. Riva, K. Strauss, A. Mittal, and A. Ntoulas, “Pocketweb: instant web browsing for mobile devices,” in *Proceedings of the ACM SIGARCH Computer Architecture News*, vol. 40. ACM, 2012, pp. 1–12.

[209] B. D. Higgins, J. Flinn, T. J. Giuli, B. Noble, C. Peplin, and D. Watson, “Informed mobile prefetching,” in *Proceedings of the 10th international conference on Mobile systems, applications, and services.* ACM, 2012, pp. 155–168.

[210] J. Ghosh, M. J. Beal, H. Q. Ngo, and C. Qiao, “On profiling mobility and predicting locations of wireless users,” in *Proceedings of the 2nd international workshop on Multi-hop ad hoc networks: from theory to reality.* ACM, 2006, pp. 55–62.

[211] P. S. Prasad and P. Agrawal, "Mobility prediction for wireless network resource management," in *Proceedings of the 41^{st} Southeastern Symposium on System Theory (SSST'09)*. IEEE, 2009, pp. 98–102.

[212] Z. R. Zaidi and B. L. Mark, "Real-time mobility tracking algorithms for cellular networks based on kalman filtering," *Mobile Computing, IEEE Transactions on*, vol. 4, no. 2, pp. 195–208, 2005.

[213] L. Mihaylova, D. Angelova, S. Honary, D. R. Bull, C. N. Canagarajah, and B. Ristic, "Mobility tracking in cellular networks using particle filtering," *Wireless Communications, IEEE Transactions on*, vol. 6, no. 10, pp. 3589–3599, 2007.

[214] J. W. J. Williams, "Algorithm 232 – Heapsort," *Communications of the ACM*, vol. 7, no. 6, pp. 347–348, 1964. [Online]. Available: http://doi:10.1145/512274.512284

[215] R. W. Floyd, "Algorithm 245: Treesort," *Commun. ACM*, vol. 7, no. 12, p. 701, 1964. [Online]. Available: http://doi.acm.org/10.1145/355588.365103

[216] X. Vasilakos, K. V. Katsaros, and G. Xylomenos, "Cloud Computing for Global Name-Resolution in Information-centric Networks," in *Second Symposium on Network Cloud Computing and Applications, NCCA 2012, London, United Kingdom, December 3-4, 2012*, 2012, pp. 88–94. [Online]. Available: http://dx.doi.org/10.1109/NCCA.2012.25

[217] K. V. Katsaros, X. Vasilakos, T. Okwii, G. Xylomenos, G. Pavlou, and G. C. Polyzos, "On the inter-domain scalability of route-by-name information-centric network architectures," in *Proceedings of the IFIP Networking Conference (IFIP Networking)*, May 2015, pp. 1–9.

[218] L. Zhang, D. Estrin, J. Burke, V. Jacobson, J. Thornton, D. Smetters, B. Zhang, G. Tsudik, D. Massey, C. Papadopoulos *et al.*, "Named Data Networking (NDN) project," Tech. report ndn-0001, PARC, Tech. Rep., 2010.

[219] D. Perino and M. Varvello, "A Reality Check for Content Centric Networking," in *Proc. of the ACM SIGCOMM Workshop on Information-centric Networking (ICN)*, pp. 44–49.

[220] C. Labovitz, S. Iekel-Johnson, D. McPherson, J. Oberheide, and F. Jahanian, "Internet inter-domain traffic," in *Proceedings of the ACM SIGCOMM 2010 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, New Delhi, India, August 30 -September 3, 2010*, 2010, pp. 75–86. [Online]. Available: http://doi.acm.org/10.1145/1851182.1851194

[221] R. V. Oliveira, D. Pei, W. Willinger, B. Zhang, and L. Zhang, "The (in)completeness of the observed internet AS-level structure," *IEEE/ACM Trans. Netw.*, vol. 18, no. 1, pp. 109–122, 2010. [Online]. Available: http://doi.acm.org/10.1145/1816288.1816297

[222] N. Fotiou, K. V. Katsaros, G. Xylomenos, and G. C. Polyzos, "H-Pastry: An inter-domain topology aware overlay for the support of name-resolution services in the Future Internet," *Computer Communications*, vol. 62, pp. 13–22, 2015.

[223] C. D. Cranor, E. Gansner, B. Krishnamurthy, and O. Spatscheck, "Characterizing large DNS traces using graphs," in *Proc. of the ACM SIGCOMM Workshop on Internet Measurement*, 2001, pp. 55–67.

[224] S. M. Besen and M. A. Israel, "The evolution of Internet interconnection from hierarchy to "Mesh": Implications for government regulation," *Information Economics and Policy*, vol. 25, no. 4, pp. 235 – 245, 2013.

[225] K. Chen, D. R. Choffnes, R. Potharaju, Y. Chen, F. E. Bustamante, D. Pei, and Y. Zhao, "Where the sidewalk ends: extending the internet as graph using traceroutes from P2P users," in *Proc. of the 2009 ACM CoNEXT*. New York, NY, USA: ACM, 2009, pp. 217–228.

[226] CAIDA, "The IPv4 Routed /24 AS Links Dataset," January 2011. [Online]. Available: http://www.caida.org/data/active/ipv4_routed_topology_aslinks_dataset.xml

[227] M. Komorowski, "A history of storage cost (update)," 2014. [Online]. Available: http://www.mkomo.com/cost-per-gigabyte-update

[228] "Amazon Elastic Compute Cloud (EC2)," 2016. [Online]. Available: https://aws.amazon.com/ec2/

[229] "Microsoft Azure," 2016. [Online]. Available: https://azure.microsoft.com

[230] J. Wulf, R. Zarnekow, T. Hau, and W. Brenner, "Carrier activities in the CDN market - An exploratory analysis and strategic implications," in *Proc. of the 2010 International Conference on Intelligence in Next Generation Networks (ICIN)*, oct. 2010, pp. 1 –6.

[231] V. Douros, S. Elayoubi, E. Altman, and Y. Hayel, "Caching Games between Content Providers and Internet Service Providers," *Elsevier Performance Evaluation (PEVA) journal (to appear)*, 2016.

[232] M. Musolesi and C. Mascolo, "Car: Context-aware adaptive routing for delay-tolerant mobile networks," *IEEE Transactions on Mobile Computing*, vol. 8, no. 2, pp. 246–260, Feb 2009.

[233] Fujitsu Network Communications Inc., "High-capacity indoor wireless solutions: Picocell or femtocell?" 2013. [Online]. Available: http://www.fujitsu.com/us/Images/High-Capacity-Indoor-Wireless.pdf

[234] G. Lucarelli, X. Vasilakos, and I. Androutsopoulos, "Named Entity Recognition in Greek Texts with an Ensemble of SVMs and Active Learning," *International Journal on Artificial Intelligence Tools*, vol. 16, no. 6, pp. 1015–1045, 2007. [Online]. Available: http://dx.doi.org/10.1142/S0218213007003680

[235] X. Vasilakos, J. Sacha, and G. Pierre, "Decentralized As-Soon-As-Possible Grid Scheduling: A Feasibility Study," in *Proceedings of the 19th International Conference on Computer Communications and Networks, IEEE ICCCN 2010, Zürich, Switzerland, August 2-5, 2010*, 2010, pp. 1–6. [Online]. Available: http://dx.doi.org/10.1109/ICCCN.2010.5560056

[236] M. Handley, "Why the Internet only just works," *BT Technology Journal*, vol. 24, no. 3, pp. 119–129, July 2006.

[237] E. Borgia, R. Bruno, and A. Passarella, "MobCCN: a CCN-compliant protocol for data collection with opportunistic contacts in IoT environments," in *Proceedings of the Eleventh ACM Workshop on Challenged Networks, CHANTS@MobiCom 2016, New York City, New York, USA, October 3-7, 2016*, 2016, pp. 63–68. [Online]. Available: http://doi.acm.org/10.1145/2979683.2979695

[238] X. Huang and N. Ansari, "Content caching and distribution in smart grid enabled wireless networks," *CoRR*, vol. abs/1603.05701, 2016. [Online]. Available: http://arxiv.org/abs/1603.05701