UNIVERSITY OF CALIFORNIA
SANTA CRUZ

SCALABILITY AND PERFORMANCE ANALYSIS OF WIRELESS NETWORKS AND INFORMATION-CENTRIC NETWORKS

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

DOCTOR OF PHILOSOPHY

in

COMPUTER ENGINEERING

by

Ali Dabirmoghaddam

September 2017

The Dissertation of Ali Dabirmoghaddam
is approved:

Dr. J. J. Garcia-Luna-Aceves, Chair

Dr. Katia Obraczka

Dr. Brad Smith

Tyrus Miller
Vice Provost and Dean of Graduate Studies

Table of Contents

List of Figures

List of Tables

List of Algorithms

Abstract

Scalability and Performance Analysis of Wireless Networks and Information-Centric Networks

by

Ali Dabirmoghaddam

Computer networks are complex constructs consisting of many entities that simultaneously interact with one another. Understanding various aspects of such complex systems and their scaling properties is therefore a challenging task. This thesis studies the performance of computer networks through several layers of abstraction—namely communication, social and information—and investigates the interplay between these layers at a large scale and from an analytical point of view. The results derived from this analysis are crucial in early identifying of design issues and potential weaknesses of large-scale networks such as the Internet for which performing simulations is prohibitive.

The first part of this dissertation studies how the spatial diversity of social contacts affect the scalability of communication networks and identifies classes of social models that let computer networks properly scale. From this analysis, it is established that scalability is achieved under social models in which social contacts are statistically concentrated within a confined geographical region around each node. We shall recognize that the true distribution of social contacts in real networks does not generally meet this requirement, imposing a scalability gap on today's networks.

The second part of the dissertation studies information-centric networking (ICN); a framework that utilizes distributed content caching that can be used to bridge the foregoing scalability gap. Three dominant methods of distributed content caching—namely uniform-, optimal- and edge-caching—are compared. We

shall see that caching only at the edge of the network outperforms uniform-caching (the de facto standard of ICN) in terms of end-to-end latency while offering slightly inferior results compared to the more complex optimal-caching strategy. This result is further augmented by the observation that higher degrees of reference locality in space and/or time improve the performance of edge-caching, making it a viable alternative to optimal solution.

Finally, an analysis of the forwarding plane of ICN is presented. In the absence of host addressing, ICN routers are required to keep track of all requests (Interests) passing through them in Pending Interest Tables (PITs) that are needed to deliver data back to the requesters as well as to enable optimization mechanisms such as Interest aggregation. Through careful analysis of the PIT size distribution and the probability of Interest aggregation at PIT, we shall see not only are the true benefits from the stateful forwarding plane of ICN much smaller than anticipated, but also they come at the high expense of maintaining very large PITs. These results reveal that the forwarding plane of ICN must be rethought; an important finding that provokes the investigation of a stateless forwarding plane for future ICNs.

For Elahe

Acknowledgments

I would like to express my sincerest appreciation to my research advisor, J. J. Garcia-Luna-Aceves, for his tremendous guidance and encouragement throughout my studies at the University of California, Santa Cruz. JJ's support and supervision was instrumental in the development of many ideas discussed in this dissertation. Having the privilege to work with such distinguished and dedicated faculty is indeed an opportunity for which I shall always remain grateful.

I wish to thank Katia Obraczka and Brad Smith who agreed to take time to read my dissertation and serve in my committee despite their busy schedule. I would also like to extend my gratitude to Emily Gregg for all her great help.

I enjoyed the company of many friends and colleagues, my fellow "COCO's", Turhan Karadeniz, Ashok Masilamani, Spencer Sevilla, Ehsan Hemmati and James Mathewson, with whom I had the pleasure of sharing office space in Computer Communication Research Group. I am also thankful to my many other friends in Baskin School of Engineering, Hossein Talebi, Mohsen Kiskani, Elnaz Ebrahimi, Sina Hassani, Ehsan Amid, Holakou Rahmanian, Hossein Daraei and my friends outside UCSC campus, Ali Abedi, Ali Rahmani, Koushan Ahmadian, Mostafa Dehghan and Fayaz Farshchi for their continuous friendship and support.

I take great pleasure in thanking my family: my mom who wanted me a medical doctor, but I could not do better than a doctoral in engineering, my dad whom I lost so early and I desperately miss, my dear brother and sister; I cannot thank them enough for all they did for me. Last, and certainly not the least, I owe an immense amount of thanks to my wife Elahe, without her constant support, encouragement and love this work would certainly be impossible. She has always been a tremendous source of hope and love for me. To her I dedicate this dissertation.

Chapter 1

Introduction

Computer networks can be conceived as complex compositions of multiple interacting constructs. On a set of computer hosts, a *communication network* defines the topology graph comprising the hosts, a.k.a. *nodes*, and communication links (wireless or wired) that physically connect them together. A *social network*, on the other hand, defines the quality and the frequency of inter-node communications. Furthermore, an *information network* determines the distribution and the flow of information among nodes.

A computer network can be viewed as a superposition of such tightly intertwined layers that are defined on the same set of nodes. The time-varying nature of the network as well as other dynamics such as node mobility make understanding different aspects of the resulting construct very complicated.

Due to such complexities, the performance of computer networks is usually studied under simplifying assumptions and with narrow focuses to mitigate some of these inherent intricacies. Abstracting away the interplay between different layers of a computer network is one such treatment. This explains why separate tracks of research deal with characterizing behaviors of computer networks from specific and unidimensional viewpoints of communication, social, or information. The

interaction among the foregoing components, however, has long been overlooked. Neglecting important aspects of the actual system, such overly simplified models can only produce implications that are limited in scope and hardly applicable to real-world scenarios.

The main objective of this thesis is to devise an analytical framework to explore the interaction among various abstractions of networked systems, namely communication, social and information. To that end, Chapter 2 of this thesis aims to improve our understanding of the relationships between the communication and social networks. Particularly, it studies how the geographical diversity of social connections can affect the scalability of large-scale computer networks. In particular, it identifies classes of social relations that let a wireless multihop network grow in size while the throughput share per node decays at a slow rate. It is interesting to see from this model that there exists classes of social relationships where the expected distance between the social contacts is always upperbounded by a constant irrespective of the network size. In that situation, the maximum throughput per node decays at a slow rate of $\mathcal{O}(1/\log n)$ where n is the number of nodes in the network. When contrasted against the well-known throughput decay rate of $\mathcal{O}(1/\sqrt{n \log n})$ [1] in which the effect of the social model is completely neglected (*i.e.*, nodes communicate uniformly and at random throughout the network), the foregoing bound seems very promising.

In real networks, however, social relationships are inherent and social contacts cannot be dictated how to interact to ensure scalability in the underlying communication graph. Given this remark, Chapter 3 studies how information-centric networking (ICN) can help enhance scalability by bringing the right information closer to the consumers through careful content caching. Within this scope, a fundamental question answered in Chapter 3 is as follows. *Given a fixed budget of*

caching storage, what is the optimal way to allocate that budget across all nodes such that the users' perceived end-to-end latency is minimized? This problem is formalized as an optimization problem in Chapter 3 and solutions to certain examples are presented. The results from this model are then compared against two other methods of cache allocation, namely *uniform-caching* and *edge-caching*. The former is the commonplace approach assumed by the majority of ICN proposals and advocates ubiquitous caching of everything everywhere in the network. The latter, however, is a less known practice that promotes caching content only near the end-users. Interestingly, the results from Chapter 3 shows that edge-caching outperforms the commonly used uniform-caching approach in a wide range of scenarios studied and is only slightly inferior to the optimal solution although the resulting solutions are fundamentally different. Furthermore, Chapter 3 studies the effect of reference locality through a novel algorithm for generating synthesized references that are localized in space and/or time. The findings from this study shows that the more localized the references, the less noticeable is the difference between edge- and optimal caching. These insights are significant in that finding the optimal distribution of caching storage on a network as complex and as dynamic as today's Internet is quite impossible. Besides, distributed content caching requires significant modifications to the core network infrastructure which may not be well-justified if the resulting gain is not substantial. In contrast, leveraging caches only at the network edge enables an incremental deployment of ICNs with "less pain and most of the gain" [2].

The information-centric networking architecture attempts to offer scalability on several fronts. In addition to pervasive content caching studied in Chapter 3, a second component of ICN is the location-agnostic content addressing by name. In fact, a fundamental design principle of ICN is name-based routing of content ob-

jects (in lieu of the conventional model that is based on host addressing using IP). Accordingly, the ICN routers construct routing tables that route to content names rather than addresses where they are located. In the absence of host addressing, the routers must keep track of all user queries (a.k.a. Interests) forwarded as well as their ingress interfaces to be able to deliver right data to the requesters on the way back. This information is populated in a data structure known as the Pending Interest Table (PIT). Because of this design, it is commonly said that the forwarding plane of ICN is *stateful*. An optimization mechanism foreseen in the design of the forwarding plane of many ICN proposals is *Interest aggregation*. Thanks to the highly skewed distribution of the users' queries (driven by the power-law nature of content objects popularity distribution), an ICN router may aggregate similar Interests for the same content object in the PIT while an Interest for that object has already been forwarded and the router is awaiting to receive the content. This is of course an intuitive design that can potentially reduce the load on the network core by suppressing the forwarding of identical Interests as well as improve the end-to-end latency. The process of aggregating Interests, nonetheless, comes at a cost. Chapter 4 quantifies the costs and benefits associated with this process through a characterization of the stateful forwarding plane of ICN. The likelihood of Interest aggregation at each router and the number of transmissions saved through the aggregation process are studied in Chapter 4. Furthermore, the PIT size distribution is also analyzed. The results from Chapter 4 shows that on the one hand, under stationary traffic less than 20% of Interests are subject to aggregation, the majority of which happens close to the producers. This calls into question most anticipated benefits of Interest aggregation in terms of reducing end-to-end latency and improving bandwidth utilization. On the other hand, the results of this analysis also show that the PIT size grows at an almost exponential

rate every level deeper in the network. This makes the PIT maintenance and operation at line speed a challenging task that yields little earnings in return. The bottom line is that not only does not the stateful forwarding plane of ICN help improve scalability, but also puts and extra burden on the routers to bookkeep humongous tables of pending Interests. This leads to the necessary conclusion that the forwarding plane of ICN must be rethought. In particular, the results from Chapter 4 shows that content caching automatically suppresses a great number of Interests requesting similar content. Therefore, if storing the forwarding state per Interest is not needed for other reasons, a stateless forwarding plane is indeed a more reasonable approach to deploy ICNs at Internet scale.

Chapter 5 concludes the thesis and briefly points out some interesting areas for future research.

Chapter 2

Scalability of Wireless Multihop Networks

In this chapter, an analytical framework is presented to investigate the interplay between a communication graph and an overlay of social relationships. We focus on geographical distance as a key element that interrelates the concept of packet forwarding in a communication network with the dynamics of interpersonal relations on the corresponding social graph. We identify classes of social relationships that let the ensuing system scale—*i.e.*, accommodate a large number of users given only finite amount of resources. We establish that geographically concentrated communication patterns are indispensable to network scalability. We further examine the impact of such proximity-driven interaction patterns on the throughput scaling of wireless networks, and show that when social communications are geographically localized, the maximum per-node throughput scales approximately as $1/\log n$, which is significantly better than the well-known bound of $1/\sqrt{n \log n}$ for the uniform communication model. These results are published and presented in IEEE SECON 2013 [3], IEEE NetSciCom 2014 [4] and IEEE IPCCC 2014 [5].

2.1 Introduction

Computer networks can be conceptually organized into several distinct layers that, though logically separate, are operationally interconnected. Within this framework, such constructs are often referred to as *complex networks*. In a complex network *the communication layer* represents the physical communication infrastructure, computing servers, and clients. A *social layer* defines the communication patterns among end users collaborating with one another through applications running on end systems (host computers). An *information layer* captures the distribution and relationships among information objects throughout the network.

The reciprocal interactions among the communication, social, and information layers of complex networks have an undeniable impact on performance. However, due to the complexity of characterizing complex networks, prior work has focused on the performance of networks from unidimensional viewpoints of communication, social, or information. Examples of studies on communication networks neglecting the latent social relationships are [1, 6, 7, 8, 9]. In contrast, several interaction patterns and social paradigms [10, 11, 12] are independently studied while the restrictions imposed by realistic underlying communication networks are neglected. Unfortunately, neglecting the interaction among the layers of a complex network renders overly simplified models with implications that are limited in scope and cannot be extended to more sophisticated real-world scenarios.

In this chapter, we present an analytical framework to investigate the interplay among the communication and social layers of complex networks. Particularly, we study how the spatial diversity of social connections affects the scalability of a wireless network. Section 2.2 provides a formal description of my model. We focus on *proximity-driven social models* according to which social relations are established with respect to the geographical vicinity of nodes. In this model, nodes

are inclined to communicate with parties that are geographically closer to them more often than with ones at farther distances. This behavior is characterized with a clustering parameter α, such that nodes show higher tendency to communicate within their proximal neighborhood for larger values of α. The relevance of this model to real-world social behaviors of people has been widely studied and verified in both online and offline domains [13, 14, 15, 16].

When examining the scaling limits, it is often desirable to assume idealistic conditions. While making the problem more tractable, the results of the analysis based on these assumptions reflect the nominal performance of the system. The routing algorithm plays an integral part in the performance of multi-hop networks. By virtue of this, it is desired to adopt a routing algorithm that exhibits optimal performance. Nonetheless, there does not exist a clear consensus about a general description of optimal routing in wireless networks. In spite of the fact that the optimality criterion is principally application-dependent, the research community has tacitly adopted the shortest path routing as the reference model. Any quantitative analysis of the shortest path routing requires comprehensive knowledge of the underlying network topology. In the context of random graphs, due to the involved uncertainties, the existing literature has inevitably resorted to approximate solutions.

Particularly, prior work on the scaling limits of wireless networks has relied on coarse approximations of the way in which information is forwarded in a network. Examples of these approximations include routing along the straight line (see for example [1, 17]) and grid-based routing (see [7, 18, 19] for instance). Although these approaches make the models easier to evaluate, they can hardly represent the complexities of the routing process in real networks. Furthermore, while the resulting models can safely be applied to networks of sufficiently high density,

they cannot directly be used for the analysis of sparser networks. To address these shortcomings, Sections 2.3 and 2.4 discuss a framework for characterizing routing dynamics in random networks more accurately. We focus in particular on *the geographic greedy forwarding* (see for example GPSR [20] and GOAFR$^+$ [21]), because it can be used to represent the routing process in both dense and extended network models. Besides, greedy forwarding methods are known to scale well as the network grows in size [22]; an important property to consider when studying the scaling properties of random networks.

Section 2.5 examines scalability conditions, and Section 2.6 provides upper-bounds on the throughput capacity of wireless networks under various classes of social communication and the greedy forwarding scheme we introduced. The two extremes of this analysis are (1) the conventional uniform communication model ($\alpha = 0$) in which nodes choose their destinations uniformly and at random, and (2) the geographically concentrated interaction models ($\alpha > 3$). For the former case, as the number of nodes (n) goes to infinity, we retrieve the well-known upper-bound of $\mathcal{O}(1/\sqrt{n \log n})$ (see [1, 7, 17]), while for the latter case, we show that a maximum throughput of no better than $\mathcal{O}(1/\log n)$ can be expected.

Our framework identifies two primary obstacles against the scaling of throughput in wireless multi-hop networks; namely *bandwidth depletion*, and *inordinate relaying load*. Bandwidth depletion is related to the communication layer of the network, and results from the node transmission range having to be sufficiently small to minimize destructive interference [23] with other receivers' signal, and large enough to prevent partitioning of network into isolated clusters. As a result, a *critical transmission radius* [24, 25], denoted by $r(n)$, has to be used to minimize interference while maintaining network connectivity at the same time. In a dense network model, $r(n)$ must shrink as the number of nodes increases

(see [24]); conversely, $r(n)$ has to expand (see [25]) with the number of nodes in the case of extended network model. In either case, the limiting value of $r(n)$ makes the available bandwidth per node gradually diminish to zero.

The problem of inordinate relaying load has its roots in the social aspect of inter-node interactions and deals with the unlimited accumulation of relaying traffic as more nodes join the network. The situation is especially aggravated if there is no preference as to how nodes choose their corresponding destination(s). Accordingly, on the average, nodes get to communicate with the farthest nodes in the network just as frequently as they do with their immediate neighbors. In the limit, this gives rise to the formation of infinite length paths and thereby, a throughput share that asymptotically converges to zero. We demonstrate that this problem can be avoided if an inherent tendency exists to favor social contacts that are geographically closer. This probabilistic communication model concentrates social interactions within logical clusters of certain radius around each node. For a specific range of clustering exponent, *i.e.*, $\alpha > 3$, we show that the expected radius of such a cluster becomes finite. In that case, though the problem of inordinate load is remedied, the best throughput scaling becomes of order $\mathcal{O}(1/\log n)$, because of the bandwidth depletion problem.

Section 2.7 provides an overview of related work, and Section 2.8 concludes the chapter and discusses some avenues for future research. In particular, we will establish that while hosts cannot be brought physically closer to one another in the network, the content that they share can. The results of future chapters, in particular, indicate that caching of information near the consumers of such information can be used to emulate localized communication patterns that render better scaling of networks.

To summarize, the key contributions in this chapter are as follows.

- Presenting an analytical framework to capture the interplay between a communication network and an overlay of social relationships.
- Providing a new perspective on the characterization of wireless networks through decoupling the function of social interactions from the natural limitations of the physics of wireless communication.
- Exploring the impact of geographical diversity of social interactions on the scalability and throughput enhancement of wireless multi-hop networks.

2.2 Model and Assumptions

The term *scalable* is often used to describe systems capable of handling a large number of users without incurring significant loss in performance. In this section, a more objective definition of scalability will be presented. We introduce a cost metric that reflects the average amount of resources needed to accommodate a typical user. In the context of communication networks, a reasonable cost measure is the average number of times a packet needs to be transmitted until delivery to its intended destination. There are three key factors that influence this measure, namely, *topological factors*, such as the physical connectivity among nodes and the number of hops separating a source-destination pair on the communication graph; *social factors*, such as the governing patterns according to which nodes interact with one another, and how a source node chooses its destinations; and *unrestrained factors* related to the physical layer effects or network load dynamics in general; factors such as interference, fading, noise and congestion which might result in loss of packets and incur re-transmissions.

In this section, we merely focus on topological and social factors which can be modeled under a minimal and general set of assumptions discussed below.

2.2.1 The Connectivity Graph Model

For the underlying network topology, let us consider a Random Geometric Graph (RGG). Thanks to their simplicity and generality, RGG's have become a de facto standard in the research community to represent the underlying topology of wireless networks. A definition of RGG is provided in the following for future reference.

Definition 2.1. *$\mathcal{G}(\mathcal{X}; r)$ represents a random geometric graph in which $\mathcal{X}$ is a point process on $\mathbb{R}^k$ that describes the distribution of nodes. Further, an undirected edge connects every pair u and v iff $\|\mathcal{X}_u - \mathcal{X}_v\| \leq r$ for a given $r \in \mathbb{R}^+$.*

Here, $\|.\|$ is a norm of choice on $\mathbb{R}^k$. For simplicity, the Euclidean norm will be used in this chapter. We consider a Poisson point process (P.P.P.), $\mathcal{X}$, to describe the geographical distribution of nodes' in the network. The physical connectivity between nodes is defined according to a Boolean model that assumes nodes as being connected if and only if they are within a distance r from one another.

For simplicity, We assume that nodes are distributed on the surface of a sphere. This assumption has been commonly used to alleviate the network edge effect (see [1, 17] for example). It has been shown [1] that similar results can be derived when nodes are distributed on the plane though through much more tedious and unwieldy computations.

Two distinct models are usually considered when studying asymptotic behaviors of RGG's: the *extended model*, in which the node density is fixed but the network dimensions go to infinity; and the *dense model*, in which the network dimensions are fixed but the node density goes to infinity. In this chapter, a general framework will be presented that can be used to analyze the scaling properties of both models. We shall use *random network* in arguments that can equally be applied to both extended and dense models.

2.2.2 The Social Model

The social model describes the quality and frequency of inter-node communications in the network, *i.e.*, how sources choose their destinations. In this chapter, we consider a proximity-driven social model defined as follows.

Definition 2.2. *A communication network follows a proximity-driven social model if the probability of every node u and v communicating with each other is inversely proportional to $\|\mathcal{X}_u - \mathcal{X}_v\|^\alpha$ for some arbitrary but fixed exponent $\alpha \in \mathbb{R}_0^+$.*

Definition 2.2 implies a social model that is power-law distributed with distance. For a specific realization of the network, the probability of node u choosing v as destination, $P_u(v)$, is obtained as follows according to this definition.

$$P_u(v) = \frac{d(u,v)^{-\alpha}}{\sum_{w \neq u} d(u,w)^{-\alpha}}\,, \tag{2.1}$$

where $d(u,v) = \|\mathcal{X}_u - \mathcal{X}_v\|$. The denominator of (2.1) is in fact a normalizing constant (for that specific realization).

According to Eq. (2.1), the closer two nodes are geographically, the more likely they are to communicate; except for the case of $\alpha = 0$ that results in a uniform communication model in which a source node is equally likely to choose any other node as its destination, irrespective of their distance. At the other extreme, when $\alpha \to \infty$, every node communicates with its closest neighbor almost surely. In fact, different ranges of α correspond to distinct classes of social relationships with identical scaling behaviors. Identifying such social classes is a primary objective of this chapter.

Let us denote the number of nodes in the network by n. We want to obtain a probability distribution for the event of having a social contact at any given physical distance. For the sake of generality, we normalize the distance metric with respect to the critical transmission range $r(n)$. This allows our social model to be

equally applicable to either of the dense or extended network models. Particularly, for the case of a dense network in which the geometric diameter of the network is fixed and the critical transmission range approaches zero, this adjustment allows for having social contacts that are spaced infinitely far away.

Let $X_{\min} \leq x \leq X_{\max}(n)$ be such a range-adjusted distance measure, where $X_{\min}$ and $X_{\max}(n)$ respectively denote the minimum and maximum range-adjusted distances between two social contacts. Without loss of generality, we assume that nodes' social contacts are at least $r(n)$ distance away from them; hence, $X_{\min} \triangleq 1$. Also, $X_{\max}(n) \triangleq d/r(n)$, where d is the geometric diameter of the network (*i.e.*, the longest possible physical distance between any pair of nodes).

Define $F_\alpha(x) = \Pr\{\text{having a social contact at distance } \leq x\, r(n)\}$. According to Definition 2.2, assume that such density function is a power-law on distance. Also, by Poisson approximation, we know that the number of potential social contacts at any distance x is linearly proportional to x. Therefore, we define the corresponding p.d.f. as follows.

$$f_\alpha(x) = C_\alpha\, x\, r(n) \cdot \big(x\, r(n)\big)^{-\alpha} = C_\alpha \left(x\, r(n)\right)^{1-\alpha},$$

in which C_α is a constant independent of x. To obtain the value of C_α note that

$$1 = \int_1^{X_{\max}(n)} f_\alpha(x)\, \mathrm{d}x = \begin{cases} \dfrac{C_\alpha}{r(n)} \times \log x \Big|_1^{X_{\max}(n)} & \text{if } \alpha = 2\,, \\ \dfrac{C_\alpha\, r(n)^{1-\alpha}}{2-\alpha} \times x^{2-\alpha} \Big|_1^{X_{\max}(n)} & \text{if } \alpha \neq 2\,. \end{cases}$$

Solving for C_α yields

$$C_\alpha = \begin{cases} \dfrac{r(n)}{\log X_{\max}(n)} & \text{if } \alpha = 2\,, \\ \dfrac{(2-\alpha)\, r(n)^{\alpha-1}}{X_{\max}^{2-\alpha}(n) - 1} & \text{otherwise}\,. \end{cases} \tag{2.2}$$

The p.d.f. $f_\alpha(x)$ provides a description of our proximity-driven social model. We shall use this model to define our cost measure as discussed in the following subsection.

2.2.3 The Expected Social Path Length

Recall that the cost that every packet imposes on the network is measured by the expected number of times it has to be transmitted in the network until delivered. Knowing the average number of hops each packet travels while observing the underlying social relations, we define the expected social path length as follows.

Definition 2.3. *The expected social path length (*ESPL*) is the expected number of hops, $\bar{h}(x)$, separating a source-destination pair on a proximity-driven social network identified by $f_\alpha(x)$ and is computed as*

$$\mathbb{E}[\mathscr{L}_\alpha] = \int_1^{X_{\max}(n)} f_\alpha(x)\, \bar{h}(x)\, \mathrm{d}x\,. \tag{2.3}$$

Definition 2.3 exploits the notion of geographical distance to connect the process of routing on the physical graph of the network with the nature of social interactions. In view of that, ESPL is, in fact, a cost measure reflecting the amount of resources that every node consumes on average, respecting both topological and social considerations.

Evidently, ESPL is a non-decreasing function of the network size. Nevertheless, the network cannot sustain a continuously increasing load forever as more nodes join in. Hence, we present the following definition for the class of social relations that allow the underlying communication network scale appropriately without significant loss in performance.

Definition 2.4. *A communication network with proximity-driven social relations is scalable if* $\mathbb{E}[\mathscr{L}_\alpha] < \infty$ *when* $n \to \infty$.

Based on Definition 2.4, a necessary condition for scalability is that the network performs, on average, a finite number of transmissions per packet irrespective of the size of the network. In the sequel, we address the impact of different values of α on the growth of ESPL as the network grows larger. To that end, the next section introduces a methodology to compute the average number of hops, $\bar{h}(x)$, that a routing algorithm takes over any given distance x.

2.3 Progressive Walks

It follows from Definition 2.3 that an accurate evaluation of ESPL depends at least in part on the performance of the routing algorithm used in the network. It is commonplace to characterize the system performance under idealistic conditions leading to nominal upper-bounds on the achievable performance limits. As such, the underlying routing algorithm is assumed to be optimal for modeling purposes. Although finding optimal paths on deterministic graphs is algorithmically straightforward, in the context of random graphs, this turns out to be a highly challenging problem. Most of this complexity stems from the random nature of the underlying topology. In essence, an optimal routing algorithm requires global and exact information about the network structure and state, which is virtually non-existent when speaking of RGG's.

Several approximations of optimal routing have been studied in the literature, such as routing along the straight line [1, 17], and grid-based routing [7, 18]. These approximations are often sufficiently accurate when studying a random dense network, but are less useful for the analysis of extended networks with finite density.

Moreover, the internal mechanisms of such routing schemes are remarkably different from how distributed routing algorithms work in real networks.

Despite the theoretical difficulties in the analysis of optimal routing in random configurations, more tractable solutions with near-optimal performance can still be conceived. One such routing strategy is known as *greedy (geographical) forwarding* in which intermediate relays attempt to push the packet some distance closer to the destination. With this policy, even though the global structure of the routes need not be necessarily optimized, a sub-optimal path can still be found by making locally optimized decisions when choosing subsequent relays along the path.

Various criteria for optimizing local decisions have been studied in the literature leading to different variations of the geographical forwarding method. We abstract away such functional details by introducing the notion of *progressive walk* that captures the essence of greedy forwarding.

Definition 2.5. *We say a walk* $\langle s, \ldots, t \rangle$ *on* $\mathcal{G}(\mathcal{X}; r)$ *is a progressive walk from* s *to* t *and denote it with* $s \rightsquigarrow t$ *iff* $\|\mathcal{X}_u - \mathcal{X}_t\| \geq \|\mathcal{X}_v - \mathcal{X}_t\|$ *for all ordered pairs* (u, v) *on* $s \rightsquigarrow t$.

For a given source-destination pair, a greedy forwarding algorithm attempts to output a progressive walk on the communication graph. The expected number of hops on a greedy route is equivalent to the expected length of the corresponding walk. To succeed, a greedy forwarding algorithm requires that a physical path does exist for the intended source-destination pair. However, the algorithm may not possibly succeed even though such a path does exist.

The existence of a walk is trivial when the expected length of the walk is a parameter of interest. However, the ability of a greedy forwarding algorithm to always be able to find a progressive walk with high probability (w.h.p.) need not be true because of possible dead-ends, and hence this assumption has to be made

to simplify the given problem. Later, however, we will relax this assumption by slightly modifying the routing algorithm to circumvent dead-ends, should they be encountered. Let us first have a closer look at the core mechanism of the greedy routing algorithm, *i.e.*, progressive forwarding, through the following definition. Here, $\mathcal{B}(\mathcal{X}_c; r)$ denotes the ball of radius r centered at $\mathcal{X}_c$.

Definition 2.6. *The hand-off region of a relay u for a final destination t is $\mathcal{H}_t(u) \triangleq \mathcal{B}(\mathcal{X}_u; r) \cap \mathcal{B}(\mathcal{X}_t; x)$, where $x = \|\mathcal{X}_u - \mathcal{X}_t\|$. Further, we say that node v is a potential next-hop for $u \rightsquigarrow t$ iff $\mathcal{X}_v \in \mathcal{H}_t(u)$.*

According to Definition 2.5, the hand-off region defines the subset of nodes that can be considered by a relay node as potential next hops to further the packet towards its destination. The convergence of the progressive walk relies upon having at least one potential next-hop in each and every hand-off region along the walk. If the packet comes at a relay with a void hand-off region, *i.e.*, a dead-end, the progressive walk stalls as no further progress towards destination is possible. For the time being, we assume that the greedy algorithm converges w.h.p.

2.3.1 Greedy Forwarding with Almost Sure Convergence

The key element of a progressive walk according to Definition 2.5 is to progressively reduce the remaining distance to the destination along the walk. In fact, at every stage of the walk, the packet is pushed some distance closer to the destination in the Euclidean space. In view of this, a progressive walk can be perceived as a drifted random walk on the communication graph. The distance traveled by the packet at every hop is a random variable determined by the process specifying the topology of the communication graph as well as the optimization criteria of the greedy routing algorithm. Exploiting results from the theory of martingales,

Theorem 2.1 provides a useful model that describes the relationship between the physical distance and the average hop-count on a RGG, under certain conditions when a greedy forwarding algorithm is considered.

Assume that node s sends a packet to t through multiple intermediate hops employing a geographical greedy forwarding algorithm. Let ξ_δ be a random variable denoting the progress towards destination if a transmission at distance δ from destination takes place. The following theorem provides bounds on the expected number of hops under a greedy forwarding algorithm given an initial physical distance x.

Theorem 2.1. *Consider a source s and a destination t at distance $x = \|\mathcal{X}_s - \mathcal{X}_t\| > r$ in a RGG $\mathcal{G}(\mathcal{X}; r)$. Provided that ξ_δ's are independent, and the routing algorithm converges w.h.p.,*

$$\lim_{\delta \to r^+} \mathbb{E}[\xi_\delta] < \frac{x}{\bar{h}(x)} < \lim_{\delta \to \infty} \mathbb{E}[\xi_\delta]\,.$$

Proof. Let $S_\delta(t) = \sum_{i=1}^{t} \xi_\delta(i)$ be a random walk where $\xi_\delta(i)$ is a stochastic process with respect to i representing the progress towards destination when at distance δ from it. In fact, S_δ is a progressive walk that assumes all relays have similar-sized hand-off regions as if they are all at distance δ from destination.

Let $T_\delta = \inf\{t : S_\delta(t) \geq x\}$ be the first time $S_\delta(t)$ hits the target distance x. Note that $0 \leq \xi_\delta(i) \leq r$ and $\mathbb{E}[\xi_\delta] > 0$; thus, $P(T_\delta < \infty) = 1$. Also, $\{t < T_\delta\} = \{S_\delta(1), \ldots, S_\delta(t) < x\}$ which is clearly independent of $S_\delta(t')$ for $t' > T_\delta$. Therefore, T_δ is a stopping time with respect to $S_\delta(t)$.

Fix a δ such that $r < \delta < x$, and consider a relay at distance δ from destination. The measure of hand-off region is a monotonically increasing function of δ (see Figure 2.1). Therefore,

$$\lim_{\delta \to r^+} \mathbb{E}[\xi_\delta] < \mathbb{E}[\xi_\delta] < \lim_{\delta \to \infty} \mathbb{E}[\xi_\delta] \quad \text{for all } \delta > r\,. \tag{2.4}$$

Now, consider the process $M_\delta(t) = S_\delta(t) - t\,\mathbb{E}[\xi_\delta]$. Note that,

$$\begin{aligned}\mathbb{E}[M_\delta(t)] &= \mathbb{E}\Big[S_\delta(t) - t\,\mathbb{E}[\xi_\delta]\Big] = \mathbb{E}\Big[\sum_{i=1}^{t}\xi_\delta(i) - t\,\mathbb{E}[\xi_\delta]\Big] \\ &= \mathbb{E}\Big[\sum_{i=1}^{t}\Big(\xi_\delta(i) - \mathbb{E}[\xi_\delta]\Big)\Big] = \sum_{i=1}^{t}\mathbb{E}\Big[\xi_\delta(i) - \mathbb{E}[\xi_\delta]\Big] \\ &= \sum_{i=1}^{t}\Big(\mathbb{E}[\xi_\delta] - \mathbb{E}[\xi_\delta]\Big) = 0 < \infty\,.\end{aligned}$$

Also, $\mathbb{E}[M_\delta(t+1) - M_\delta(t)] = \mathbb{E}[M_\delta(t+1)] - \mathbb{E}[M_\delta(t)] = 0$. Therefore, $M_\delta(t)$ is a martingale with respect to ξ_δ. According to the optional stopping theorem, $M_\delta(T_\delta \wedge t)$ is also a martingale with respect to ξ_δ, where $(T_\delta \wedge t)$ is the minimum of T_δ and t. Hence,

$$\begin{aligned}\mathbb{E}[M_\delta(T_\delta)] &= \mathbb{E}\Big[S_\delta(T_\delta) - T_\delta\,\mathbb{E}[\xi_\delta]\Big] \\ &= \mathbb{E}[S_\delta(T_\delta)] - \mathbb{E}[T_\delta]\cdot\mathbb{E}[\xi_\delta] = 0\,,\end{aligned}$$

which yields

$$\mathbb{E}[S_\delta(T_\delta)] = \mathbb{E}[T_\delta]\cdot\mathbb{E}[\xi_\delta]\,. \tag{2.5}$$

Now, consider the process $S(t) = \sum_{i=1}^{t}\xi_y(t)$, where $y = \max\big(x - S(t-1), r\big)$ and $S(0) = 0$. Let $T = \min\{t : S(t) \geq x\}$ be a stopping time. From Equation (2.4), for all $y > r$ we have that

$$\begin{aligned}&\lim_{\delta\to r^+}\mathbb{E}[\xi_\delta]\cdot\mathbb{E}[T] < \mathbb{E}[\xi_y]\cdot\mathbb{E}[T] < \lim_{\delta\to\infty}\mathbb{E}[\xi_\delta]\cdot\mathbb{E}[T] \quad \Rightarrow \\ &\lim_{\delta\to r^+}\mathbb{E}[\xi_\delta]\cdot\mathbb{E}[T] < \quad \mathbb{E}[S(T)] \quad < \lim_{\delta\to\infty}\mathbb{E}[\xi_\delta]\cdot\mathbb{E}[T] \quad \Rightarrow \\ &\lim_{\delta\to r^+}\mathbb{E}[\xi_\delta)] < \quad \frac{\mathbb{E}[S(T)]}{\mathbb{E}[T]} \quad < \lim_{\delta\to\infty}\mathbb{E}[\xi_\delta]\,.\end{aligned}$$

Having $\mathbb{E}[S(T)] = x$ and noting that $\mathbb{E}[T] = \bar{h}(x)$ is in fact the average number of hops over distance x, we obtain that

$$\lim_{\delta\to r^+}\mathbb{E}[\xi_\delta] < \frac{x}{\bar{h}(x)} < \lim_{\delta\to\infty}\mathbb{E}[\xi_\delta]\,,$$

which completes the proof. □

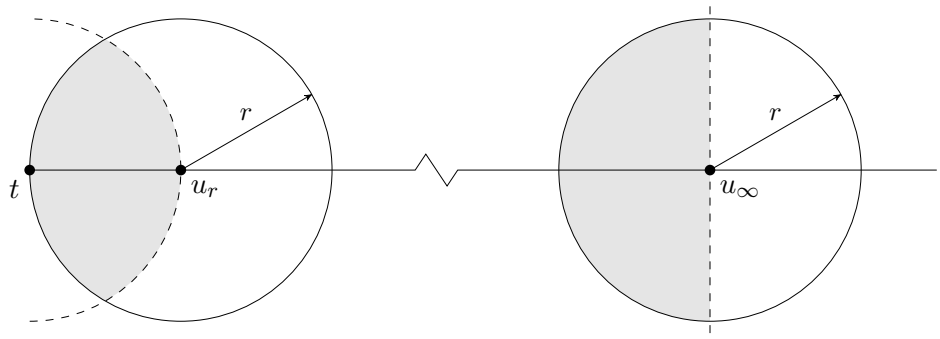

Figure 2.1: A 2-D illustration of how the hand-off region, *i.e.*, shaded area, shrinks as the remaining distance to the destination is reduced. Here t is the destination, and u_r and u_∞ represent relays at distances r and ∞ from destination, respectively.

As mentioned earlier, a problem that limits the accuracy of the given bounds in Theorem 2.1 is the assumption that the routing algorithm converges w.h.p. This assumption might be true when studying dense networks, but it is not applicable to networks of finite node density in which a dead-end might be encountered. In the following, we extend the case studied in Theorem 2.1 to account for such possibilities as well.

2.3.2 Greedy Forwarding with Backtracking

We analyze a modified greedy forwarding algorithm which works as follows. At every stage t of the walk, the packet either makes a progress of $+\xi(t)$ towards destination with probability p, or backtracks for a random step size of $-\xi(t)$ with probability $1-p$ in the event of encountering a dead-end. Considering the underlying P.P.P., the probability p is then

$$p = 1 - \exp\big(-\rho\,|\mathcal{H}(\cdot)|\big)\,,$$

where ρ is the intensity of the P.P.P., and $|\mathcal{H}(\cdot)|$ denotes the Lebesgue measure of the hand-off region. The corresponding random walk, hence, is formalized as follows.

$$S(t) = \begin{cases} S(t-1) + \xi(t-1) & \text{with probability } p\,, \\ S(t-1) - \xi(t-1) & \text{with probability } 1-p\,, \end{cases}$$

and $S(0) = 0$. Therefore, $\mathbb{E}[S(t)] = S(t-1) + (2p-1)\,\xi(t-1)$. Consider the process $M(t) = S(t) - t\,(2p-1)\,\mathbb{E}\xi$ for $t > 0$. We first verify that $M(t)$ is a martingale.

$$\begin{aligned} &\mathbb{E}[M(t+1) \,|\, M(t)] \\ &\quad = \mathbb{E}\big[S(t+1) - (t+1)(2p-1)\mathbb{E}\xi \,\big|\, S(t) - t(2p-1)\mathbb{E}\xi\big] \\ &\quad = \mathbb{E}\big[S(t) + (2p-1)\xi(t) - (t+1)(2p-1)\mathbb{E}\xi \,\big|\, \cdot\,\big] \\ &\quad = S(t) - t\,(2p-1)\,\mathbb{E}\xi = M(t)\,. \end{aligned}$$

Define a stopping time $T = \inf\{t : S(t) \geq D\}$. By the optional stopping theorem, $\mathbb{E}[M(T)] = \mathbb{E}[M(0)] = 0$. Thus,

$$\begin{aligned} \mathbb{E}[S(T)] = (2p-1)\,\mathbb{E}[T]\,\mathbb{E}[\xi] \qquad &\Rightarrow \\ x = (2p-1)\,\mathbb{E}[T]\,\mathbb{E}[\xi] \qquad &\Rightarrow \\ \mathbb{E}[T] = \frac{x}{(2p-1)\,\mathbb{E}[\xi]}\,. \qquad &(2.6) \end{aligned}$$

The natural constraint of $\mathbb{E}[T] > 0$ requires that $p > 1/2$ in order for Eq. (2.6) to make sense. As $p \to \frac{1}{2}^+$, $\mathbb{E}[T]$ diverges, which is an intuitive behavior. Also, when $p = 1$, (2.6) simplifies to (2.5) which is also expected.

The bounds given in Theorem 2.1 are expressed in terms of the expected progress the greedy forwarding algorithm makes per hop when at a limiting distance of ∞ or r from destination. As seen from Figure 2.1, in 2-D space, the hand-off region shrinks from a half-disk at the former distance to a shape resembling a biconvex lens at the latter. Aside from the size of the hand-off region,

the expected progress per hop does also depend on the forwarding policy of the greedy algorithm, *i.e.*, the criteria by which the next relay is chosen from within the set of potential next-hops. In the next section, we examine the tightness of the suggested bounds in Theorem 2.1.

2.4 Expected Progress Per Hop

Several policies for next hop selection have been proposed in the context of greedy routing algorithms. A widely used policy is to always choose the next-hop with the least remaining distance (LRD) to the destination. Even though this strategy does not guarantee that the packet would necessarily travel the fewest number of hops, it ensures the maximum possible progress towards destination at every hop.

An issue with the LRD policy is that it violates the required condition on the independence of per-hop progresses. To clarify, observe that the hand-off regions of subsequent hops are not necessarily disjoint. For instance, in Figure 2.2, the hand-off region of node u overlaps that of node v on $v \rightsquigarrow t$ in the crosshatched region. Therefore, if node v is chosen as next-hop for $u \rightsquigarrow t$ under LRD, then v cannot logically have a potential next-hop in the crosshatched region. This implies that when LRD is used as the forwarding policy, the information from the past history of the walk can, in fact, affect the future decisions.

In order to use Theorem 2.1, we must make sure that the adopted forwarding policy does not violate the independence of succeeding progresses as described above. One such compliant policy is random greedy forwarding (RGF) by which a current relay forwards the packet to a randomly chosen next hop. Such a next-hop could clearly be located anywhere within its hand-off region, and its election as the next relay does not impose any restriction on the location of subsequent hops.

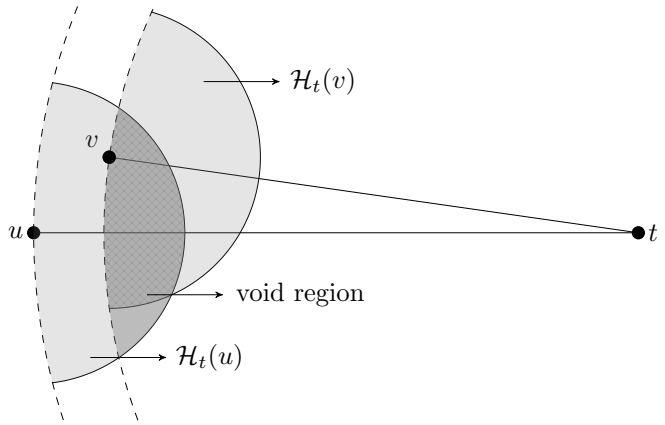

Figure 2.2: Succeeding hand-off regions may overlap. Here, the darker shaded region is empty, and part of it, *i.e.*, the crosshatched area, overlaps the hand-off region of v on $v \rightsquigarrow t$.

As such, RGF satisfies the requirements of Theorem 2.1.

In the following, we quantify the expected progress per hop under RGF policy. It is noteworthy that although RGF is not an optimal forwarding strategy, it can still serve as a lower-bound for more aggressive policies such as LRD.

2.4.1 A Lower-bound on Per-Hop Progress

Consider the case when the source and destination are located at a distance $\delta + \epsilon$ for a small positive $\epsilon \to 0$. In that case, the hand-off region for the source can be approximated by a symmetrical biconvex lens, as illustrated in the left-hand-side of Figure 2.1. For the moment, assume $r = 1$ and define the boundaries of the hand-off region as follows.

$$|\omega| = \begin{cases} \sqrt{1-(1-\delta)^2} = \sqrt{2\delta - \delta^2} & \text{if } 0 \leq \delta \leq \frac{1}{2}\,, \\ \sqrt{1-\delta^2} & \text{if } \frac{1}{2} \leq \delta \leq 1\,. \end{cases}$$

Due to the symmetry of the region, the enclosed area can be calculated as

$$A(r) = 4 \cdot \int_0^{\frac{1}{2}} \int_0^{\sqrt{2\delta-\delta^2}} \mathrm{d}\omega \, \mathrm{d}\delta = 4 \cdot \int_0^{\frac{1}{2}} \sqrt{2\delta - \delta^2} \, \mathrm{d}\delta\,.$$

Since the next-hop can be located anywhere within the hand-off region with equal probability, $\mathbb{E}[\xi(r)]$ is the expected distance from the relay over the region which can be calculated as follows.

$$\mathbb{E}[\xi(r)] = \frac{2}{A(r)}\left(\int_0^{\frac{1}{2}} \int_0^{\sqrt{2\delta-\delta^2}} \sqrt{\delta^2+\omega^2}\, \mathrm{d}\omega\, \mathrm{d}\delta + \int_{\frac{1}{2}}^{1} \int_0^{\sqrt{1-\delta^2}} \sqrt{\delta^2+\omega^2}\, \mathrm{d}\omega\, \mathrm{d}\delta \right).$$

Using numerical methods and noting that $\mathbb{E}[\xi(r)]$ is linear in r, for a general case, we obtain that

$$\lim_{\delta \to r^+} \mathbb{E}[\xi(\delta)] \approx 0.643\, r\,. \tag{2.7}$$

2.4.2 An Upper-bound on Per-Hop Progress

Consider the right-hand side of Figure 2.1. The hand-off region is defined as follows.

$$|\omega| \leq \sqrt{1-\delta^2} \quad \text{for } 0 \leq \delta \leq 1\,.$$

The area of the hand-off region is clearly $A(\infty) = \pi/2$. Hence,

$$\mathbb{E}[\xi(\infty)] = \frac{2}{A(\infty)}\left(\int_0^{1} \int_0^{\sqrt{1-\delta^2}} \sqrt{\delta^2+\omega^2}\, \mathrm{d}\omega\, \mathrm{d}\delta \right) = \frac{2}{3}\,.$$

By analogy to the previous case, we obtain that

$$\lim_{\delta \to \infty} \mathbb{E}[\xi(\delta)] \approx 0.667\, r\,. \tag{2.8}$$

From Theorem 2.1 and Equations (2.7) and (2.8), we obtain that, under a routing with RGF policy, the average hop count over any given distance $x \gg r$ is bounded as

$$1.50 \left(\frac{x}{r}\right) < \bar{h}(x) < 1.56 \left(\frac{x}{r}\right). \tag{2.9}$$

Note that x/r is the theoretical lower-bound on the number of hops under any routing scheme, which, of course, can almost never be attained on a RGG.

2.5 Scalability Analysis

We now examine the scalability conditions of random networks under proximity-driven social models, taking advantage of the mathematical models developed in previous sections. The following theorem identifies a large family of social relationships that allow a communication network to scale.

Theorem 2.2. *Under a proximity-driven social model identified by the power-law p.d.f. $f_\alpha(\cdot)$, a random network exhibits scalability if $\alpha > 3$.*

Proof. Recall from Definition 2.4 that the required condition on scalability is to maintain $\mathbb{E}[\mathcal{L}_\alpha] < \infty$ when the network size grows infinitely large. From Eq. (2.9) and for a range-adjusted distance measure x we established that $C'_{\min}\, x < \bar{h}(x) < C'_{\max}\, x$ for constants $C'_{\min}, C'_{\max} > 0$ independent of x. Plugging $C'_{\max}$ into Eq. (2.3) and expanding yields

$$\begin{aligned}
\mathbb{E}[\mathscr{L}_\alpha] &< \int_1^{X_{\max}(n)} C_\alpha \left(x\, r(n)\right)^{1-\alpha} \cdot C'_{\max}\, x \,\mathrm{d}x \\
&= C'_{\max}\, C_\alpha\, r(n)^{1-\alpha} \int_1^{X_{\max}(n)} x^{\,2-\alpha}\,\mathrm{d}x \\
&= \begin{cases} C'_{\max}\, C_\alpha\, r(n)^{-2}\, \log X_{\max}(n) & \text{if } \alpha = 3\,, \\ \dfrac{C'_{\max}\, C_\alpha}{3-\alpha}\, r(n)^{1-\alpha}\Big(X_{\max}^{3-\alpha}(n) - 1\Big) & \text{otherwise}\,. \end{cases}
\end{aligned}$$

Replacing C_α from Eq. (2.2) results in the following upper-bounds on Espl.

$$\mathbb{E}[\mathscr{L}_\alpha] < \begin{cases} C'_{\max}\, \dfrac{X_{\max}(n) - 1}{\log X_{\max}(n)} & \text{if } \alpha = 2\,, \\ C'_{\max}\, \dfrac{X_{\max}(n) \log X_{\max}(n)}{X_{\max}(n) - 1} & \text{if } \alpha = 3\,, \\ C'_{\max}\, \dfrac{\alpha - 2}{\alpha - 3} \times \dfrac{X_{\max}^{3-\alpha}(n) - 1}{X_{\max}^{2-\alpha}(n) - 1} & \text{otherwise}\,. \end{cases}$$

Likewise, replacing $C'_{\max}$ by $C'_{\min}$ results in similar lower-bounds. Note that $\lim_{n\to\infty} X_{\max}(n) = \infty$. Therefore, we can express Espl in terms of $X_{\max}(n)$

as follows.

$$\mathbb{E}[\mathscr{L}_\alpha] = \begin{cases} \Theta\big(X_{\max}(n)\big) & \text{if } 0 \leq \alpha < 2\,, \\ \Theta\left(\dfrac{X_{\max}(n)}{\log X_{\max}(n)}\right) & \text{if } \alpha = 2\,, \\ \Theta\big(X_{\max}^{3-\alpha}(n)\big) & \text{if } 2 < \alpha < 3\,, \\ \Theta\big(\log X_{\max}(n)\big) & \text{if } \alpha = 3\,, \\ \Theta(1) & \text{if } 3 < \alpha\,. \end{cases} \tag{2.10}$$

As seen from Eq. (2.10), ESPL becomes a constant independent of n only when $\alpha > 3$ and the theorem follows. □

Figure 2.3 provides a pictorial view of the relations provided in Eq. (2.10). The figure illustrates how ESPL grows against an increasingly growing network diameter ($X_{\max}(n)$) for various degrees of clustering (α). A linear relationship is evident for $0 \leq \alpha < 2$. When $\alpha = 3$, ESPL is still growing but the growth rate is very slow with respect to the network geometric diameter. For any $\alpha > 3$, ESPL becomes constant and decreases with increasing α to a limiting value of 1 as $\alpha \to \infty$. At that point, each node only communicates to its closest neighbor (that is one hop away) almost surely.

An alternative view of the relationship between ESPL and the clustering exponent α is depicted in Figure 2.4. As seen, $\alpha > 3$ is the scalable region where ESPL demonstrates a stable behavior. As soon as α drops below the threshold of 3, ESPL demonstrates a rapid initial growth which gradually flattens out around $\alpha = 2$. The monotonic relationship between ESPL and $X_{\max}(n)$ is evident for $0 \leq \alpha < 2$.

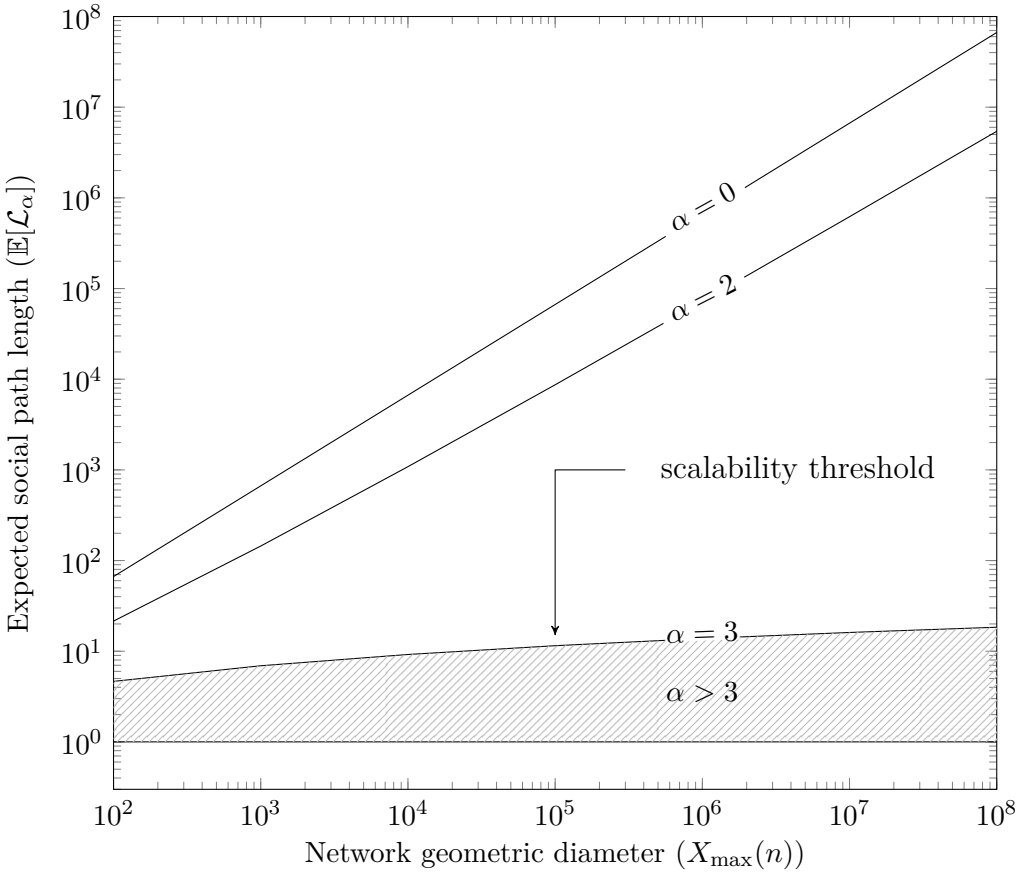

Figure 2.3: The growth rate of ESPL for various clustering exponents (α) as the geometric diameter of the network increases. $\alpha > 3$ is the scalability threshold and the shaded region highlights networks with bounded average social path lengths.

2.6 Upper-bounds on Throughput

Our scalability analysis can be extended to a characterization of the throughput capacity of random networks when various social interaction models are applied. The analysis presented below is in fact a generalization of the upper-bound throughput calculation in [1]. Assume a network consisting of n nodes each capable of transmitting W bits per second. Each node chooses a social contact at random according to some proximity-driven social model with parameter α as

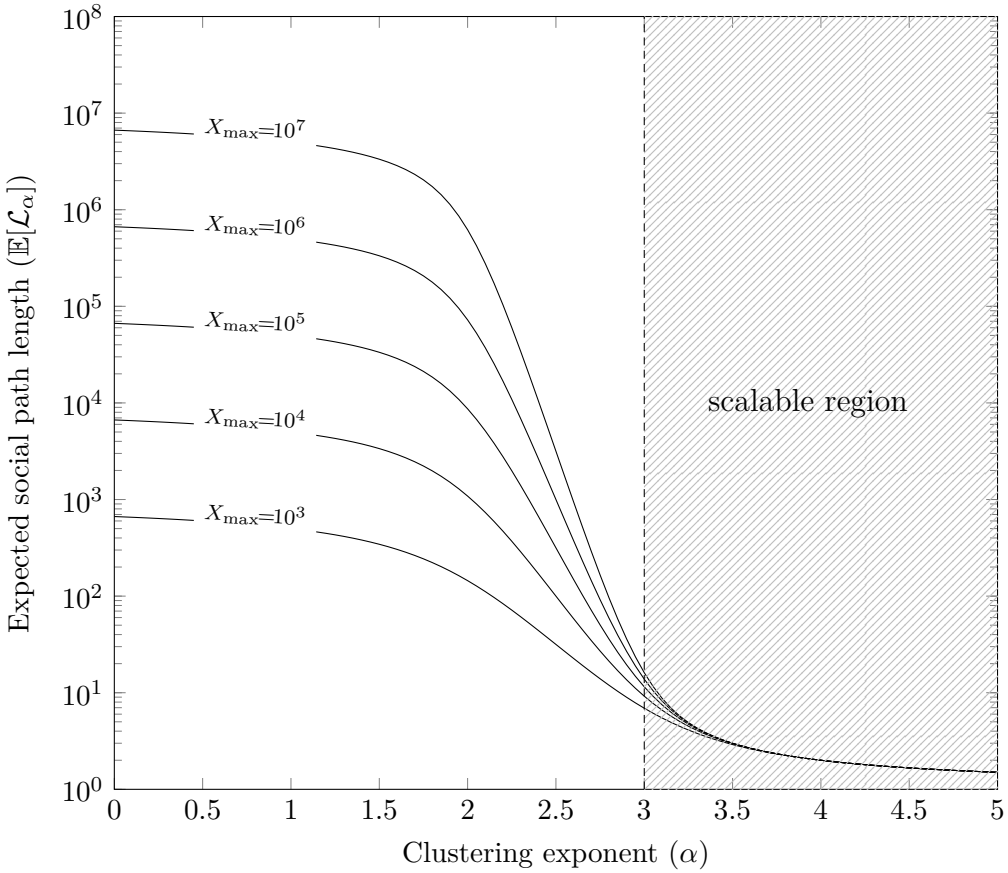

Figure 2.4: The decaying rate of ESPL versus an increasing clustering exponent (α). The shaded area, *i.e.*, $\alpha > 3$, illustrates the scalable region in which a finite expected social path length is attained.

described in Section 2.2.2. Let $R_\alpha(n)$ be the rate at which each node transmits (including both original and relaying traffic) and each packet goes through a path consisting of $\mathbb{E}[\mathcal{L}_\alpha]$ hops on average. The network, hence, carries a total of $n\, R_\alpha(n)\, \mathbb{E}[\mathcal{L}_\alpha]$ bits per second.

Due to the shared nature of the wireless medium, some distributed medium access control protocol must be in place to avoid multiple access interference. For that purpose, a simple TDMA scheme similar to the *protocol model* in [1] can be conceived. According to this model, a transmission from node u to v is considered

successful if (1) $\|\mathcal{X}_u - \mathcal{X}_v\| \leq r(n)$, and (2) $\|\mathcal{X}_v - \mathcal{X}_w\| \geq (1+\Delta)\, r(n)$ for every node w transmitting simultaneously with u over the same (sub)channel.

Lemma 2.1. *The maximum number of simultaneous transmissions the network can handle grows as* $\Theta\big(X^2_{\max}(n)\big)$.

Proof. The proof shares a similar logic with the proof of Lemma 5.4 in [1]. By requirement (2) of the protocol model and the triangle inequality, simultaneous receivers on the same (sub)-channel must be at least $\Delta\, r(n)$ distance away from one another (See Figure 2.5). Thus, the surface of the network can be covered with disjoint disks of radius $\Delta\, r(n)/2$ centered at each receiver. The area of each such disk is $\Theta\big(r(n)^2\big)$. The total area of the network is $\Theta\big(d^2\big)$. The maximum number of simultaneous receivers is thus $\Theta\big(d^2/r(n)^2\big) \equiv \Theta\big(X^2_{\max}(n)\big)$. Every receiver corresponds to an identical transmitter and hence, the lemma follows. □

From Lemma 2.1, it follows that the maximum accumulative traffic in the network cannot grow faster than $\Theta\big(X^2_{\max}(n)\big)$. In symbols,

$$n\, R_\alpha(n)\, \mathbb{E}[\mathcal{L}_\alpha] \leq \Theta\big(X^2_{\max}(n)\big)\,. \tag{2.11}$$

We use this result to derive the theoretical maximum throughput capacity per node.

Lemma 2.2. *In a random network,* $X_{\max}(n) = \Theta\Big(\sqrt{\frac{n}{\log n}}\Big)$.

Proof. In Section 2.2, we defined $X_{\max}(n) \triangleq d/r(n)$. We prove the lemma for the cases of dense and extended networks separately, referring to some known results from the random networks literature.

The case of dense network: The critical transmission range to ensure connectivity in dense graphs is derived by Gupta and Kumar [24] as $r(n) = \Theta\Big(\sqrt{\frac{\log n}{n}}\Big)$. Noting that $d = \Theta(1)$ for dense networks, the result follows immediately.

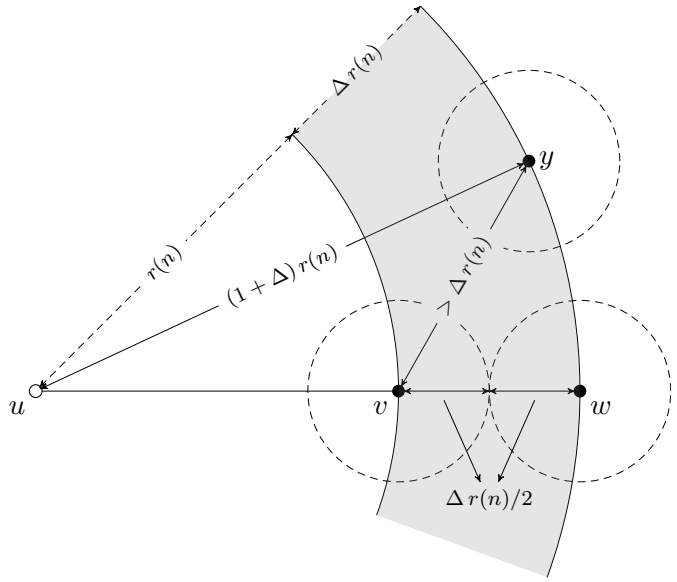

Figure 2.5: To avoid multiple access interference, the protocol model demands concurrent receivers over the same (sub)channel to maintain a distance of at least $(1 + \Delta)\, r(n)$ from an irrelevant active transmitter. Here, u is transmitting to v. The shaded region (cropped to save space), with a width of $\Delta\, r(n)$, is the guard zone in which no other node can simultaneously receive. w and y are at distance $(1+\Delta)\, r(n)$ from u and thus, can be simultaneous receivers over the same (sub)channel without being affected by u's signal. By triangle inequality, such simultaneous receivers cannot be closer than $\Delta\, r(n)$ to v. Therefore, imaginary disks of radius $\Delta\, r(n)/2$ centered at all simultaneous receivers are disjoint.

The case of extended network: Santi and Blough [25] derive the critical transmission range for extended networks as $r(d) = \Theta(\sqrt{\log d})$, where d is the geometric diameter of the network. For an extended network, $d = \Theta(\sqrt{n})$ and thus, $r(n) = \Theta(\sqrt{\log n})$. The lemma follows. □

Using Lemma 2.2 and the bounds obtained in Eq. (2.10), we derive the theoretical upper-bounds on the per-node throughput.

Theorem 2.3. *In a random network and under a proximity-based social model with clustering exponent α, the theoretical maximum per-node throughput $R_\alpha(n)$ is bounded above as*

$$R_\alpha(n) = \begin{cases} \mathcal{O}\left(\frac{1}{\sqrt{n\,\log n}}\right) & \text{if } 0 \le \alpha < 2\,, \\ \mathcal{O}\left(\sqrt{\frac{\log n}{n}}\right) & \text{if } \alpha = 2\,, \\ \mathcal{O}\left(\left(\frac{1}{n}\right)\left(\frac{n}{\log n}\right)^{\frac{\alpha-1}{2}}\right) & \text{if } 2 < \alpha < 3\,, \\ \mathcal{O}\left(\frac{1}{\log^2 n}\right) & \text{if } \alpha = 3\,, \\ \mathcal{O}\left(\frac{1}{\log n}\right) & \text{if } 3 < \alpha\,. \end{cases} \tag{2.12}$$

Proof. From Lemma 2.2 and Eq. (2.11), it is immediate that

$$R_\alpha(n) \le \Theta\big(\frac{1}{\log n\,\mathbb{E}[\mathcal{L}_\alpha]}\big)\,.$$

The theorem follows by replacing $\mathbb{E}[\mathcal{L}_\alpha]$ with corresponding bounds given in Eq. (2.10). □

Figure 2.6 illustrates the bounds derived by Theorem 2.6. The drastic decay in per-node throughput can be observed when $0 \le \alpha < 2$. The shaded region depicts the transition into a slowly decaying order of $1/\log n$ when α exceeds a threshold of 3.

The throughput order of $1/\sqrt{n\,\log n}$ was first derived in the pioneering paper by Gupta and Kumar [1] under the assumption of uniform communication model when nodes choose their destinations randomly and uniformly, *i.e.*, $\alpha = 0$. The result in Theorem 2.3 is consistent with this well-known result. In fact, Theorem 2.3 makes a stronger statement that this bound indeed holds for a larger

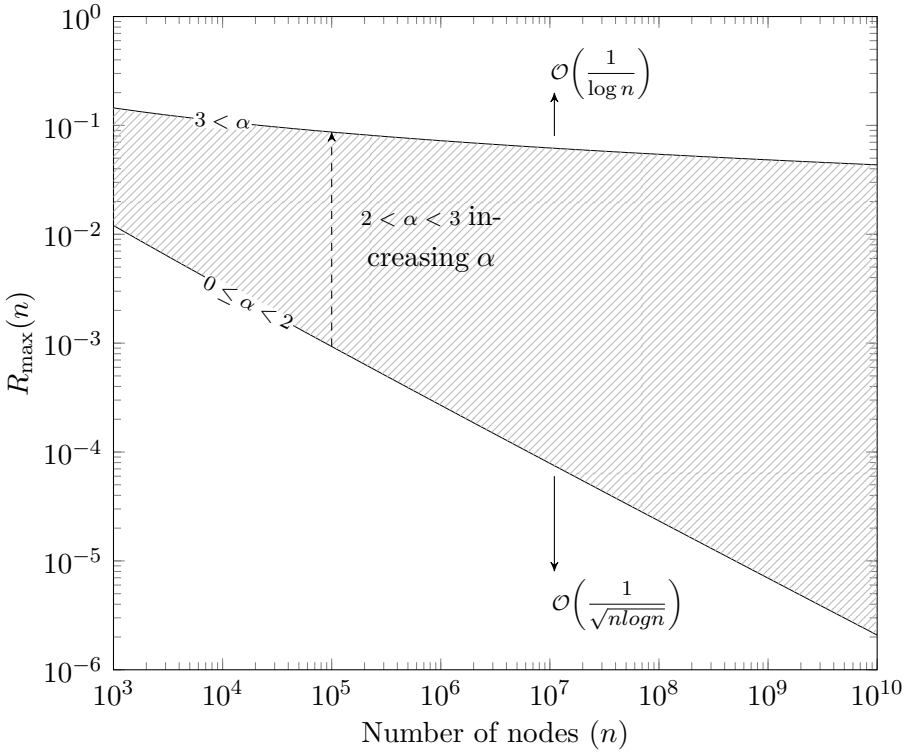

Figure 2.6: An illustration of how the throughput capacity decays versus the network size when various classes of social relationships are in effect. Curves are to only exhibit the approximate decaying rates and should not be interpreted as being accurate.

range of α between 0 and 2. With increasing α, the order throughput improves and eventually gets bounded above by $1/\log n$ when α exceeds 3.

This suggests that the maximum throughput scaling to be expected is of order $\mathcal{O}(1/\log n)$ for $\alpha > 3$ when the expected social path length remains constant. In that case, even though every packet is transmitted only a finite number of times along its path to the designated destination, still, a constant throughput per node cannot be attained. The fundamental reason for this limitation is the restrictions imposed by the MAC layer and the connectivity requirement. The area consumed by every transmission is quadratic in the nodes' transmission radius and this

prohibits other nodes within the interference range from being able to receive their intended packets at the same time. On the other hand, in order to maintain a giant connected component in both the extended and dense network models, the nodes' critical transmission radius must be adjusted such that an increasing number of nodes are covered within the one-hop neighborhood as the network grows larger. This natural requirement inevitably contributes to larger interfering groups of nodes and hence, increasingly limits the available capacity per node.

Though even highly localized interaction patterns cannot guarantee a decaying rate of better than $\mathcal{O}(1/\log n)$ in per-node throughput, in practice, this can still serve as a reasonable upper-bound should it be realized. Theorem 2.3 does not provide insights as to whether or not this bound is achievable. Nonetheless, a constructive approach to study the feasibility of this bound, similar to the lower-bound analysis in [1], can be developed. This analysis is left as a subject of future research.

2.7 Related Work

Originated by the seminal paper of Gupta and Kumar [1] and followed by a handful of subsequent work (*e.g.*, [7, 8, 9, 17, 26]) in the past decade, it was revealed that the asymptotic per-node throughput in wireless multi-hop networks rapidly decays as the network grows in size. This unfavorable behavior was primarily attributed to the effect of interference caused by the shared nature of wireless medium that could only be avoided at the expense of some sort of spatial or temporal sacrifice of bandwidth. Shortly after, a flurry of research discussed potential mechanisms to mitigate the wireless channel impairments and thereby, improve the throughput capacity through leveraging mobility [6, 17], hybridization [27, 28], directional antennas [29, 30], or cooperative transmission [31, 32].

All these proposals have correctly identified a root cause of throughput deterioration in wireless networks by focusing on the limitations due to the physics of wireless communication. Nonetheless, there also exists a social aspect to networking which has received much less attention. A social model defines the patterns according to which nodes interact with one another. For this side of the problem, the mainstream literature has generally resorted to a naïve uniform interaction model—see for example [1, 6, 7, 17, 19]. As it turns out, not only does such a simplistic model fail to reflect a realistic picture of interaction patterns in the network, but it also yields an overly pessimistic view of the network scaling limits. A remarkable body of research [13, 14, 15, 16] has been undertaken to explore geographical dynamics of social interactions in real networks, but no notable piece of work has been devoted to investigating the impact of applying realistic social models on the scalability and throughput capacity of communication networks. A primary objective of this chapter was to provide additional insight in the exploration of this cross-domain problem.

2.8 Conclusions and Outlook

We investigated how geographical diversity of social interactions can affect the scalability of communication networks. Particularly, we identified a threshold on the spatial diversity of social interactions beyond which the majority of inter-node communications become statistically concentrated within a finite neighborhood around nodes. We showed that this phenomenon enables the underlying communication graph to scale as the number of nodes in the network increases.

We further examined how the upper-bound on the throughput capacity can be improved if the social interactions among nodes are fueled by geographical proximity. According to our analysis, an upper-bound of $\mathcal{O}(1/\log n)$ can be

expected on the maximum per-node throughput if communication patterns are highly concentrated. Although more promising than the well-established bound of $\mathcal{O}(1/\sqrt{n \log n})$ under uniform communication model, the foregoing result does not yet guarantee the existence of networking mechanisms to realize this bound. In fact, the feasibility of this limit depends largely on the agreement of my social model description with the actual communication patterns among nodes in real networks.

The spatial diversity of social contacts may not naturally meet the necessary localization requirements as needed by my model. Even in that case, similar bounds may still be achievable by employing mechanisms such as in-network content replication and caching to bring content closer to consumers. I believe this is an interesting topic for future research.

Chapter 3

Distributed Content Caching in Information-Centric Networks

In this chapter, a formal framework is presented for the characterization of cache allocation models in Information-Centric Networks (ICN). The framework is used to compare the performance of optimal caching everywhere in an ICN with opportunistic caching of content only near its consumers. This comparison is made using the independent reference model adopted in all prior studies, as well as a new model that captures non-stationary reference locality in space and time. The results obtained analytically and from simulations show that optimal caching throughout an ICN and opportunistic caching at the edge routers of an ICN perform comparably the same. In addition, caching content opportunistically only near its consumers is shown to outperform the traditional on-path caching approach assumed in most ICN architectures in an unstructured network with arbitrary topology represented as a random geometric graph. The content of this chapter is published and presented in ACM ICN 2014 [33].

3.1 Introduction

Over years, the traffic workload on the Internet has evolved from endpoint-based message passing towards more of the distribution and sharing of (often multimedia) content. This shift has given rise to many content-oriented networking paradigms such as web caches, content delivery networks (CDN) and peer-to-peer (P2P) systems that are built on top of the existing TCP/IP stack. These designs are often application-/service-specific and hence are costly and inefficient. In recent past, several Information-Centric Networking (ICN) architectures [34, 35, 36] have been introduced to redesign the current architecture of the Internet to accommodate content-oriented applications and services. Decoupling the content from Internet addresses is fundamental to information-centric networking and using this principle, ICN architectures seek to provide the necessary foundation for scalable and cost-efficient content distribution. A key design component of many ICN architectures is the *universal in-network caching* of named data objects in an opportunistic fashion. The *universality* implies that content caching should be done *everywhere* and for *everything* in the network. The former requires all ICN routers to equally contribute in the network-wide process of caching, while the latter necessitates the ICN routers to cache all kinds of traffic they handle, irrespective of the popularity of the content or its geographical relevance. This approach is used to attain such performance benefits as reduced response time, efficient content distribution, and improved disruption tolerance.

As the review of prior work in Section 3.2 points out, even though universal in-network caching is assumed in many ICN architectures, there has been no quantitative analysis justifying this choice compared to opportunistic caching of content near its consumers. The main objective of this chapter is to provide a formal framework for the characterization of the performance of optimal in-

network caching in ICN's, as well as opportunistic in-network caching at the edge of ICN's—*i.e.*, close to the end-users.

Section 3.3 uses the *Independent Reference Model* (IRM), which assumes that references to content objects occur independently, to study the benefits of using universal caching compared to a simple policy of caching only at the edge of the network. The results of this analysis, supported by extensive event-driven simulations over a wide range of configurations, indicate that the optimal caching approach based on universal caching provides only marginal benefits over the simple policy of caching only at the edge of the ICN. Although empirical studies [2] in the past have shown similar results, this chapter attempts to explain this important finding from a mathematical point of view.

Section 3.4 addresses the impact of *locality of references* (*i.e.*, content requests) on the performance of caching in an ICN. This work is inspired in part by the results delineated by Traverso *et al.* [37] on the temporal locality of content references. I introduce a novel view of reference locality that captures both spatial and temporal aspects. The reference locality highlights the property that a request for a content object is likely to trigger subsequent requests for the same object from same geographical neighborhood (*i.e.*, spatial locality) in near future (*i.e.*, temporal locality). In other words, object references are localized in both space and time. Most prior work (*e.g.*, [38, 39, 40, 41]) neglects the existence of such dependencies by simply resorting to the IRM model.

Exploiting the notion of cluster point processes [42], I present a general method to produce synthetic traces of object references while maintaining their locality properties. The process of generating such non-stationary traces complies with the intuitive perception of spread of epidemics on today's social networks. An information object first attracts attention in a certain geographical region. Information

consumers start sharing the content with their social contacts. A subpopulation of their contacts who find the content interesting re-share it and this process is repeated so long as the content retains its informational value in the network. I leverage the fact that this description closely matches that of a self-exciting Hawkes process [43] and present a new algorithm to produce a synthetic trace in which, while the collective popularity profile of objects follows the commonly observed Zipf distribution [44, 45], the occurrences of object-specific references over time and space are locally clustered when observed on a smaller scale. Based on this, I introduce a convenient measure to quantify the clustering degree of references on a scale from 0 to 1. I call this measure the *localization factor*, which can be used to cover the entire spectrum of reference patterns, from IRM (when it equals 0) to highly localized (as it goes to 1).

Armed with these new tools, I extend the comparison of universal in-network caching with simple caching at the edge of an ICN for traces not necessarily conforming to the IRM assumption. The results from my model in conjunction with event-driven simulations show that, while the optimal caching naturally drifts towards the edge as the caching budget increases, higher degrees of reference locality can further accelerate this transition. According to my findings, a 35% difference between edge- vs. optimal caching under the IRM assumption decreases to only 8% with a locality factor of 0.9.

Section 3.5 addresses the problem of caching in an unstructured ICN modeled using a random geometric graph. Given that the optimal solution for universal in-network caching cannot be obtained for this complex case, ICN architectures have adopted caching of content along the paths taken by content objects from producers or caches to consumers. This heuristic method is recently popularized under the name "Transparent En-Route Caching" (TERC) [34]. My simulation

results using ndnSIM [46] demonstrate that opportunistic caching at the edge of an ICN outperforms TERC in virtually all circumstances. While this result may be surprising at first, it can be explained with the insight gained by the model. TERC forces routers to store excessive amounts of content that induces much more content replacement along paths, while edge-caching tends to store more what is of interest to consumers near the routers.

This chapter does not advocate specific mechanisms or ICN architectures. However, it provides new tools (e.g., the generation of meaningful synthetic traces) to analyze novel caching approaches in the future, and insight that has been missing to date on the caching schemes adopted in ICN architectures. In particular, given that universal in-network caching is not needed to attain efficiency, and given that edge-caching performs so well, new approaches should be developed that better integrate content routing and congestion control with content caching near consumers. Architecturally, our results indicate that deploying different types of routers in ICN's—some without caching functionality—would be far more cost effective. In the words of Fayazbakhsh *et al.* [2], content caching "at the edge" of ICN's renders "less pain, most of the gain."

3.2 Related Work

3.2.1 Caching

Although isolated caches have been studied extensively in the past (*e.g.*, [47, 48]), many aspects of interconnected networks of caches are not yet fully understood. Such systems, often described as "cache networks", are formed by serializing multiple cache modules such that the aggregate miss stream of multiple downstream caches constitutes the input to an upper level cache. Cache net-

works first became a subject of interest as a means to improve the performance of the Web [49, 50], and work on ICN architectures has renewed interest in this topic [39, 41, 51, 52, 53, 54, 55, 56, 57].

Understanding the full dimensions of networks of caches is naturally much more complicated than that of individual caches when operating in isolation. Many existing methods developed for analyzing the performance of isolated caches are based on algorithms that themselves are computationally expensive. For simplicity, these methods often introduce certain approximations that come at the inevitable cost of inaccuracy. Despite being negligible in the analysis of individual caches, these errors can aggregate and propagate through the system and produce a cascading effect when used in analysis of networks of caches.

A highly accurate approximation of least recently used (LRU) caching was introduced by Che *et al.* [49]. This analysis was recently revived in a seminal work by Fricker *et al.* [40] and shown to be applicable to a much wider range of scenarios beyond the specific conditions that Che *et al.* had initially anticipated. In the following, we briefly review this method which we shall refer to as "CHE-APRX"—abbreviated form of Che-approximation—hereinafter.

Consider a system comprising a total of N information objects and a LRU cache with capacity C. The requests for an object n come at the cache forming a Poisson process with rate $q(n)$. In fact, $q(n)$ signifies the popularity of object n in the system—*i.e.*, the proportion of total requests that belongs to n. The more popular an object n, the higher $q(n)$ as compared with other objects.

Che *et al.* define the *characteristic time* of a cache of size C, denoted by t_C, as the time it takes the cache to be filled with unique objects subject to the request rates $q(\cdot)$ under the IRM assumption, and show that t_C is indeed the unique root

that solves the following equation for t:

$$C = \sum_{i=1}^{N} \left(1 - e^{-q(i)\,t}\right). \tag{3.1}$$

Knowing t_C, the miss probability $m(n)$ for an object n, according to Che-aprx, is derived as:

$$m(n) \approx e^{-q(n)\,t_C}. \tag{3.2}$$

As mentioned earlier, Che-aprx is proved to be very accurate and highly versatile. However, there are two important restrictions in this approximation.

1) Equal-sized objects. All information objects in Che-aprx are of equal size—more precisely, unit-size such that the cache is able to store at most C objects. This assumption might seem far from reality at first, though becomes more plausible if objects are assumed to be segmented into equal-sized chunks, as required by many existing ICN proposals. It has also been shown [40] that Che-aprx can readily be extended to also account for variable-sized objects. This, however, makes the derivations more unwieldy with little, if any, extra benefit to the purpose of our analysis. Hence, we choose to keep this assumption in place.

2) Independent object references. Che-aprx assumes that the requests for information objects—a.k.a. references—arrive at the cache according to an i.i.d. process, independent of the past history of the requests and following a distribution determined by $q(\cdot)$ function. This assumption—generally referred to as the *Independent Reference Model* (IRM)—is fairly standard to many similar analyses for tractability and in order to calculate stationary cache hit/miss rates.

Although the IRM assumption is convenient, it is too simplistic in the context of cache networks, where object references exhibit strong correlation in both space and time domains. Consider for example a new song, while listed among the top hits of the month, may be highly popular for a certain period, but gradually

loses popularity as newer hits are released. Furthermore, if the song is in a specific language, it may be well-received in certain regions of the world where that language is widely spoken, while attracting little attention in other regions. The first example reflects the *temporal locality* of references, in contrast to the *spatial locality* highlighted by the second example. The IRM assumption neglects such localities in space and time by assuming that content popularity is stationary.

This shortcoming of the IRM model might not be critical for the case of a single cache that is local to a specific region, though of high importance when it comes to a large network of interconnected caches with a mix of traffic being exchanged among nodes. The extent to which this can influence the performance of a cache network is not previously studied and is a topic to be investigated in the present chapter.

3.2.2 Architectures and Systems

In-network caching of named content is a cornerstone of many ICN architectures [34, 35, 36]. This consideration is so pervasive that many research papers (*e.g.*, [34, 54, 55, 57, 58]) use the notion of "cache network" as an abstraction to describe "content-centric networks".

The caching approach used in the vast majority of existing ICN proposals (*e.g.*, [59, 60, 61]) is the transparent en-route caching (TERC) [34] by which *all* ICN routers in the network participate in the process of content caching in conjunction with their primitive function of relaying the content objects downstream. This naïve method of caching, however, has been subject of many debates and criticisms [2, 56, 57]. To reduce caching redundancy, more complex varieties of this policy, such as probabilistic in-network caching (ProbCache) [53] and opportunistic caching using reinforced counters [62] have recently been introduced.

A handful of attempts in the past few years have been made to investigate most efficient methods of caching in ICN, both empirically [2, 37, 63] and analytically [39, 53, 56]. The results from some of these studies, however, are somewhat inconsistent and indecisive. For instance, the authors of [64] argue that caching at the *core* of the network can be more effective, as opposed to [2, 52] who advocate caching closer to the network *edge*.

Amidst this flurry of research, some researchers [56, 57] believe that the best cache placement strategy is greatly influenced by factors such as network topology; hence, there does not exist a unified strategy to be generally adopted. In contrast, other work [65] reports that the impact of topology on the performance of caching is limited.

Many notable analytical works [38, 39, 41, 54, 57, 66] focusing on the characterization of caching generally suffer from the limitations imposed by the IRM assumption. This assumption is so tightly coupled with the existing models of caching that Kurose writes [34]: "[The IRM] assumption is as fundamental for cache modeling as the memoryless assumption of exponential packet/circuit interarrival times ... are for modeling packet- and circuit-switched networks ..." This is indeed the case; however, just as exponential interarrival times, the IRM model is only a simplifying assumption to make problems tractable; there is no evidence showing that real-world traffic adheres to the IRM assumption [67, 68].

Recently, Traverso *et al.* [37] have addressed the importance of temporal reference locality on the performance of today's caching networks. Their work leverages the concept of Poisson shot noise processes as a convenient tool to model and analyze temporal reference locality. They further show that adopting the IRM assumption results in an overly pessimistic view of caching performance.

3.3 Hierarchical Caching Model

Given a hierarchical network of caches, we ask the question: how much should each layer of the hierarchy of caches contribute in the caching process in order to get the most out of a constrained total caching budget? We frame this question as an optimization problem and present solutions based on non-linear integer programming.

Consider a hierarchy of LRU caches in the form of a tree with its root acting as the content source. We assume that the source stores permanent copies of all the content objects in the system. Alternatively, the source can be considered as a collection of all possible content hosts that are logically collapsed into one single entity as the root of the tree in our model.

The tree comprises $L + 2$ levels. The content subscribers (*i.e.*, users or information consumers) are at the 0^{th} level, while the content source is at level $L + 1$. Subsequently, there exist L levels of nodes with caching capabilities between users and the content source which are sequentially labeled from bottom (level 1) to the top (level L).

The caching paradigm we seek to optimize is called "on-path caching" which works as follows. When a request for an object is raised at level 0, it is forwarded along the (unique) path of intermediate caches towards the root until a cache hit occurs. If all cache accesses are missed along the path, the request will be fulfilled by fetching a copy of the object directly from the source (root). Once located, the object is transferred on the reverse path back to the requester and a local copy is also stored on each and every cache along the path.

For simplicity, in the following analysis we assume that the hierarchy is a complete k-ary tree. Under the IRM assumption and given that the cache states are independent (which is reasonable if k is not very small), the rate at which

requests for an object n arrive at a particular cache at level ℓ can recursively be formalized as:

$$q_\ell(n) = \begin{cases} q(n) & \ell = 0\,, \\ k\, q_{\ell-1}(n)\, m_{\ell-1}(n) & 0 < \ell \leq L\,, \end{cases} \tag{3.3}$$

where $m_{\ell-1}(n)$ is the miss probability of object n at a cache of level $\ell - 1$, which can itself be calculated directly using CHE-APRX (*i.e.*, Equations (3.1) and (3.2)).

3.3.1 The Expected Time To Access Content

A parameter of interest is the *expected time to access* (ETTA) an object n, which we denote by $\tau(n)$. This is defined as the expected duration between the time a user sends a request for an object n until a copy is located in the system (either at an intermediate cache or finally at the original source). We measure this duration in terms of the number of hops between the user and the closest replica of the content along the path to the source. The following theorem gives a closed-form for calculating ETTA in terms of the miss rates of the intermediate caches.

Theorem 3.1. *Consider a tree structure with $L + 2$ levels where the users are at level 0 and the content source at level $L + 1$. Employing an on-path caching strategy as described before, the expected time to access an object n is obtained as:*

$$\tau(n) = 1 + \sum_{i=1}^{L} \prod_{j=1}^{i} m_j(n)\,, \tag{3.4}$$

where $m_j(n)$ is the miss probability of content n at a cache of level j on the path from the user towards the root of the tree.

Proof. For an object n, the problem can be modeled by a discrete-time Markov chain whose states are the levels of the caching hierarchy plus an additional state

of H denoting a cache hit (See Figure 3.1). Every state ℓ, $0 < \ell < L+1$ transits into either the following state $\ell+1$ or H with probabilities $m_\ell(n)$ and $h_\ell(n) = 1 - m_\ell(n)$, respectively. States $L+1$ and H transit into state H with probability 1.

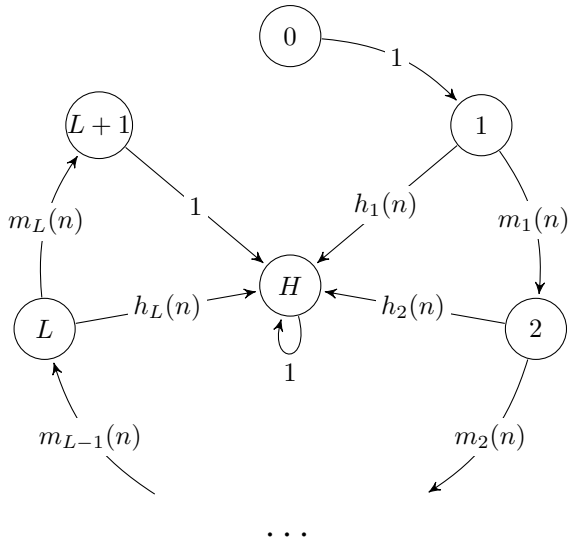

Figure 3.1: The Markov chain representing the process of locating an object on a cache hierarchy. State 0 is where the users' requests are generated. State H denotes the state of a cache hit and other states correspond to the levels in the hierarchy from bottom to the top. State $L+1$ is the root of the tree where the content source is located. $m_i(n)$ and $h_i(n) = 1 - m_i(n)$ are the miss and hit probabilities at a cache of level i.

Define $T_H \triangleq \inf\{t \geq 1 : X_t = H\}$ as the stopping time denoting when state H is visited for the first time. Also, the expected time to visit H as $\tau_0(n) \triangleq \mathbb{E}[T_H \,|\, X_0 = 0]$, where X_0 is a random variable denoting the initial state. Note that $\tau_0(n)$ counts the expected number of transitions to visit state H which can be expressed recursively as:

$$\tau_0(n) = 1 + \tau_1(n)$$

$$= 1 + 1 + m_1(n)\,\tau_2(n) + \big(1 - m_1(n)\big)\,\tau_H(n)\,.$$

Also, it is clear that $\tau_H(n) = 0$ for every object n, because visiting state H implies that the content is already located. Similarly, $\tau_2(n) = 1 + m_2(n)\,\tau_3(n)$. By induction on the index i of $\tau_i(n)$, it is easy to verify that

$$\tau_0(n) = 2 + \sum_{i=1}^{L} \prod_{j=1}^{i} m_j(n)\,.$$

In essence, $\tau_0(n)$ serves to count the expected number of steps it takes to locate the object in the hierarchy of caches. However, due to the presence of the additional state H, the real number of steps is one less than what $\tau_0(n)$ counts. Representing the actual expected value by $\tau(n)$, therefore, $\tau(n) = \tau_0(n) - 1$ and Eq. (3.4) follows. □

A slightly modified version of the foregoing result has also been used in [69] as a measure of "virtual round-trip time" to access contents of various popularity classes.

3.3.2 Optimal Cache Allocation in Tree Structure

Definition 3.1. (The optimal cache allocation problem)
Given a fixed total cache budget C, find the optimal breakdown of the caching budget across different levels of the tree that minimizes the overall expected time to access subject to a given content popularity profile $q(\cdot)$.

Under the IRM assumption, for a k-ary tree with L cache levels, we formulate this problem as a non-linear integer programming as follows:

$$\boldsymbol{c}^* = \underset{\boldsymbol{c}}{\operatorname{argmin}} \sum_{n=1}^{N} q(n)\,\tau(n;\boldsymbol{c})$$

$$\text{s.t.} \quad \sum_{\ell=1}^{L} c(\ell)\, k^{(L-\ell)} = C\,, \quad \text{and}$$

$$c(\ell) \geq 0 \text{ and integer } \forall \ell \in \{1, \ldots, L\}\,, \tag{3.5}$$

where $\boldsymbol{c}^* \in \mathbb{N}^L$ is the vector of optimal cache sizes on the tree in which $c^*(\ell)$ denotes the optimal capacity of an *individual* cache at the ℓ^{th} level.

3.3.3 Numerical Results

I collected some numerical results on the problem described above utilizing the active-set algorithm of the optimization toolbox in MATLAB. As the underlying topology, I considered k-ary tree structures with height of 7. The requesters are the leaves (*i.e.*, level 0) and the source (storing a permanent copy of all objects) is at the root (*i.e.*, level 6). The 5 intermediate levels—which we call ℓ_1 to ℓ_5 caches—are cache routers with LRU replacement policy.

All content objects in the system are ranked based on their global popularity—*i.e.*, the overall frequency of requests for that object throughout the system. For these simulations, I used one million data objects whose popularities follow a Zipf distribution with exponent 1. References to these objects are Poisson distributed with rates proportional to their popularities. For the time being, we make sure that the references conform to the IRM assumption and that identical objects have the same popularity among all users. I shall later explain how the IRM assumption can be relaxed by focusing on more general classes of traffic with non-stationary behavior in time and space.

Figure 3.2 demonstrates the optimal breakdown of the caching budget across various levels of the tree hierarchy for complete trees of degree 2 to 5. Bars show what fraction of the caching budget is allocated to various levels for any given total budget. The darker the color, the lower the cache level in the hierarchy.

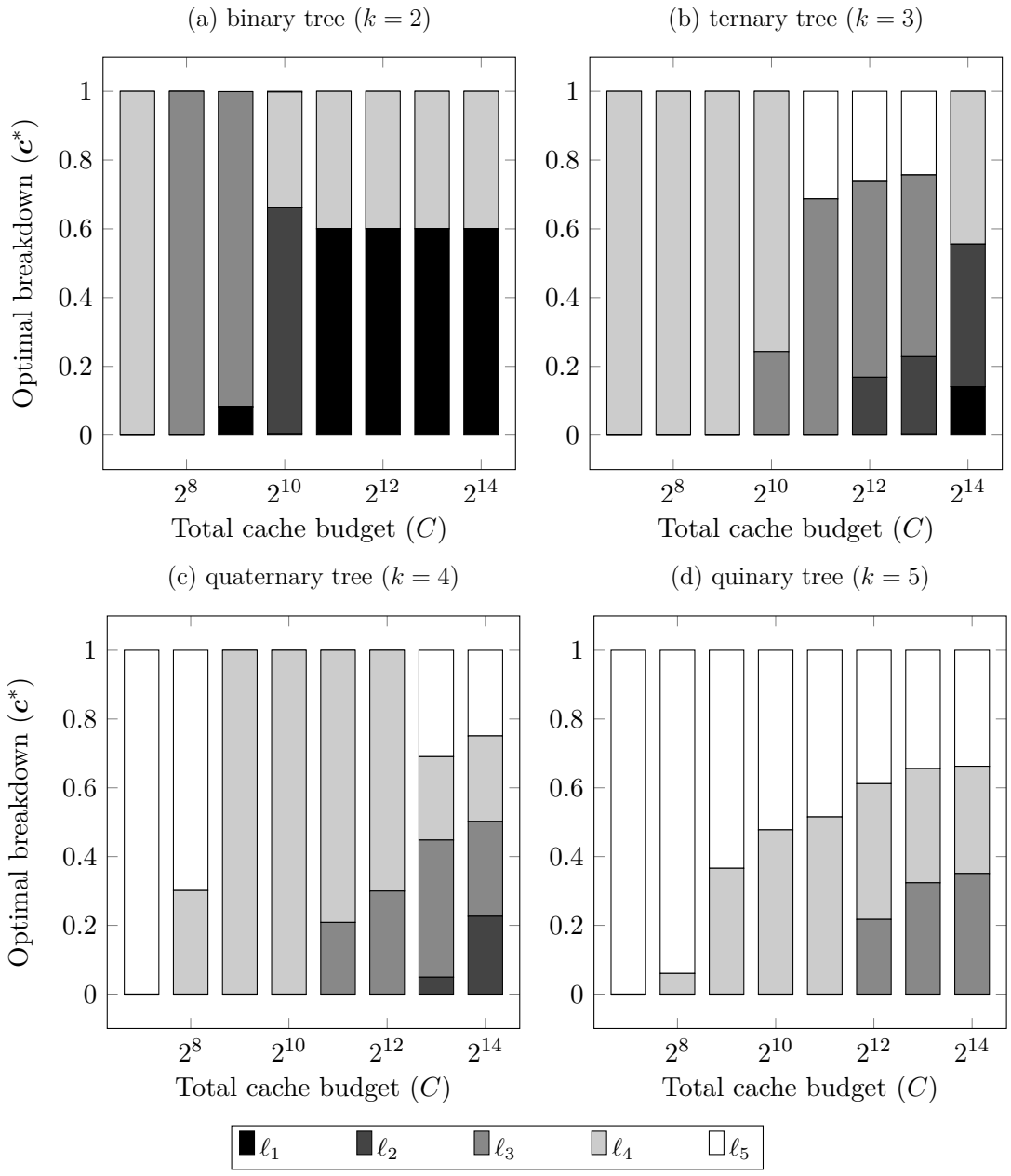

Figure 3.2: Optimal breakdown of caching budget across various levels of the tree for the given total cache budgets

As seen, there is a drift towards the edge as the caching budget increases. This is trivially expected when ETTA is the objective criterion for the optimization problem. For trees with lower degrees, this behavior is more evident, because for the same total budget, edge caches in lower degree trees receive larger shares, making edge-caching even more effective.

When the caching budget is large, it is clear that edge-caching becomes the optimal strategy. At the limit, a budget of $N \times k^L$ can be broken up evenly across all ℓ_1 caches giving each enough capacity to store a copy of every object in the system. This results in an ETTA of close to 1 in the long run. The available budget, nonetheless, is usually much less than this in practice. Figure 3.3 serves to shed some light on the question of how different optimal caching performs in general as compared with pure edge-caching by comparing the overall ETTA for edge- with optimal caching.

The solid lines in Figure 3.3 represent edge-caching, where all the caching budget is evenly split across ℓ_1 caches. The dashed lines illustrate the optimal caching with a budget breakdown specified in the corresponding part of Figure 3.2. To verify the accuracy of these results, I also designed a discrete-event simulation based on ndnSIM [46], a NS-3 module implementing Named Data Networking [70]. As seen, results from discrete-event simulations demonstrate almost perfect agreement with the proposed model.

Interestingly, Figure 3.3 suggests that edge-caching can perform comparably close to the optimal caching in practice. According to our results, the maximum difference observed between the two schemes is around 10%. The difference is seen to increase slightly with the degree of the tree. However, as Figure 3.2 also illustrates, the optimal breakdown tends towards the edge with an increased caching budget. This essentially means that the difference between the edge- and

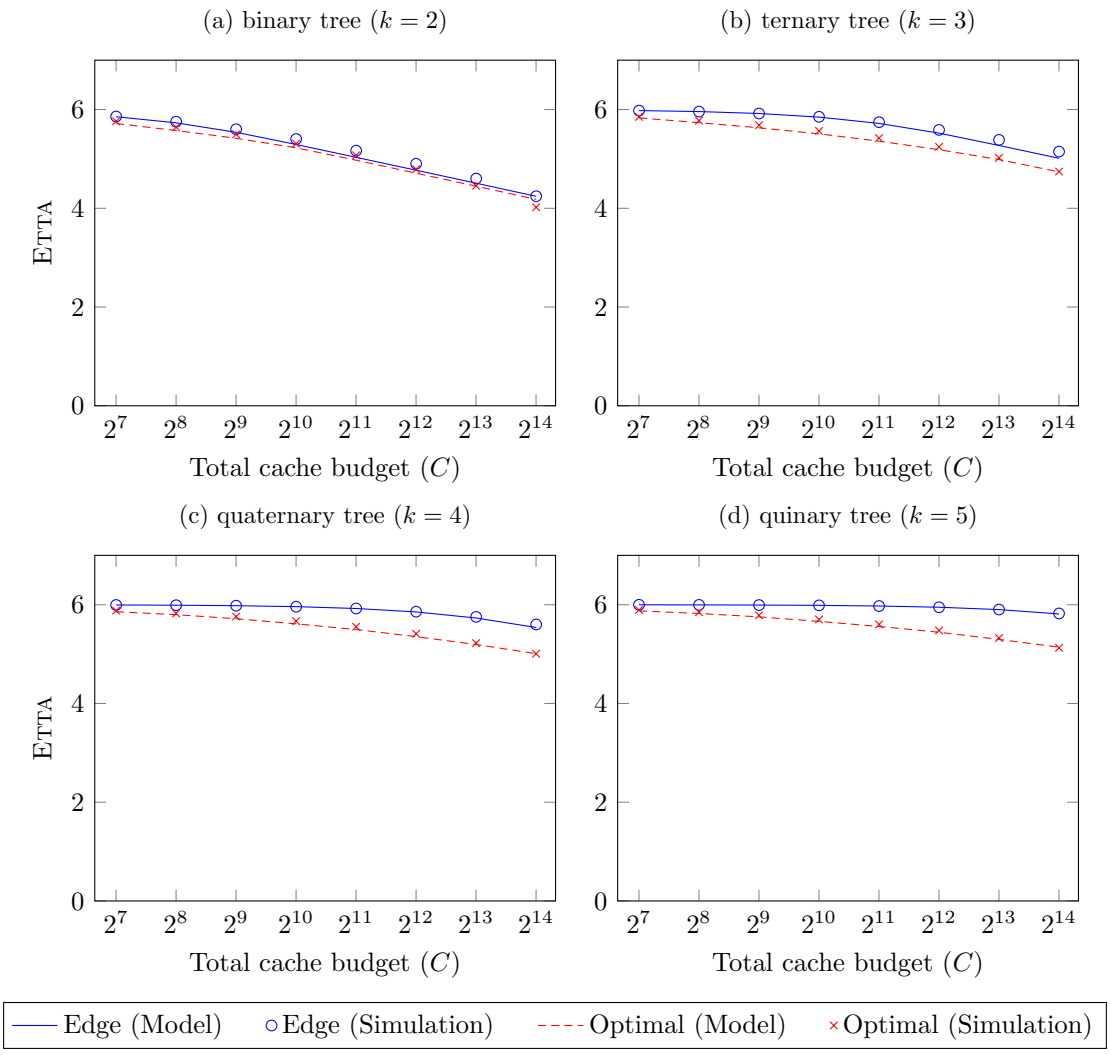

Figure 3.3: The overall expected time to access (ETTA) with respect to the total caching budget for optimal- vs. edge-caching

optimal caching is reduced with further increasing the budget. On the other hand, when the available budget is small, both edge- and optimal caching strategies appear to be equally ineffective. Thus, the maximum observed gap applies only to the cases where the available budget is modest—that is, neither so large to make edge-caching effectively optimal, nor so small to undermine the effectiveness of caching altogether.

Implementing edge-caching is practically more convenient, in that it only requires deploying ℓ_1 caches at the AS-level without any need to manipulate central routers deep in the core of the network. Although the effectiveness of edge-caching has previously been shown through extensive empirical results [2], this chapter, to the best of my knowledge, presents the first formal framework as a basis to compare the two paradigms in more depth.

3.4 Capturing Reference Locality

To obtain useful insights out of the foregoing analysis, it is imperative to evaluate the model under realistic conditions. As discussed, the IRM assumption overlooks the correlation present among subsequent object references occurring over a certain period of time (*i.e.*, temporal locality) and/or a specific region in space (*i.e.*, spatial locality). I introduce a convenient model to generate object references while preserving their spatio-temporal locality properties. Before proceeding, let us have a closer look at the intuitive interpretation of the concepts of spatial and temporal locality.

Spatial locality of reference captures the impact that the geographical diversity of the users has on the observed trace of requested objects by them. More precisely, the requests coming from a specific region in space are more likely to be similar than those collected over regions far apart. For example, a certain

news content might be of special interest in a certain area, while its global impact in the geography of interest remains limited. On the other hand, globally popular objects are seen to be requested from a wider range of geographical regions.

Temporal locality of reference captures the effect that, if an object is requested at a certain point in time, more likely it will be requested again in near future. In fact, nor are the object references scattered randomly and independently over time; rather, an object might be of particular interest at a certain time interval, while its popularity gradually fades out.

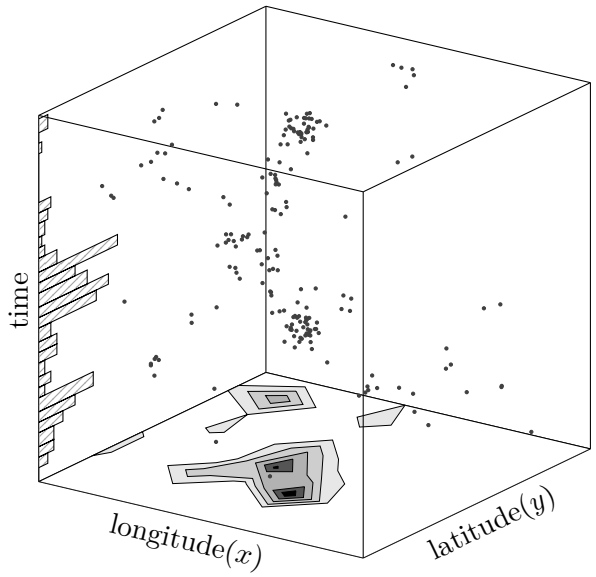

Figure 3.4: A cluster process representing references to a specific object file. The projection of points over X-Y plane represents the spatial density of requests, whereas the projection along the time axis reflects the temporal distribution.

3.4.1 Using Cluster Point Processes

In the light of the above discussion, an intuitive approach for simulating the spatio-temporal locality of object references is using cluster point processes [42].

A generic method for producing one such process works as follows. First, a point process $\mathbf{\Pi}$ generates "centers" of the process. Next, a point process $\boldsymbol{X}_p$ for each $p \in \mathbf{\Pi}$ produces the off-springs. The combination of these points $\boldsymbol{X} = \cup_{p \in \mathbf{\Pi}} \boldsymbol{X}_p$ constitutes a cluster process. Particularly, $\boldsymbol{X}$ is called a "Poisson cluster process" if $\mathbf{\Pi}$ is a Poisson process.

A specific example of the Poisson cluster process is "Hawkes process" [43] that is generated as follows. First, a Poisson process on $\mathbb{R}^d$ with intensity function $\rho(\cdot)$ creates the cluster centers $\mathbf{\Pi}$. Then, for each cluster center $p \in \mathbf{\Pi}$, the first-generation off-springs are generated as a Poisson process of intensity $\varphi(x - p)$, where $\varphi(\cdot)$ is a positive function on $\mathbb{R}^d$. This process continues repeatedly such that for every first-generation off-spring p_1, a Poisson process of intensity $\varphi(x - p_1)$ generates the second generation off-springs and so on. The mean number of off-springs for each center point is determined as $\beta = \int \varphi(x)\,\mathrm{d}x$. A natural requirement for this process to stop is $\beta < 1$. Figure 3.4 illustrates one realization of the Hawkes process in $\mathbb{R}^3$. The contour plot on the X-Y plane represents the spatial density of the requests for a certain object, while the histogram along the time axis shows the temporal evolution of the object popularity.

This procedure can be repeated for the number of content objects in the system to generate a collective trace of all references. The procedure GENERATE-TRACE in Algorithm 3.1 shows the pseudocode for this with inputs N denoting the total number of objects, α as the parameter of the Zipf distribution for object popularity, and β specifying the localization factor. For an object n, Line 4 calculates the intensity q_n at which the references to that object should be generated. To ensure that the global object popularity profile follows the desired Zipf distribution, we choose this rate to be directly proportional to the global popularity of the object in the system. Consequently, references to more popular objects will be placed

Algorithm 3.1 Method for generating object references with localization in a d-dimensional space

Input: Number of objects (N), Zipf parameter (α) and localization factor (β)
Output: An aggregate Poisson cluster process X
Ensure: X is Zipf distributed with parameter α

```
1 procedure GENERATE-TRACE(N, α, β)
2 |  X ← ∅
3 |  for n from 1 to N do
4 |  |  q_n ← M × n^{-α}                ▷ M is some constant multiplier
5 |  |  Π_n ← HAWKES-PROCESS(q_n, β)
6 |  |  X ← X ∪ Π_n
7 |  end for
8 |  return X
9 end procedure
```

over a wider geography and a longer course of time, as opposed to the less popular items which may only be requested from a specific region and a certain period. The multiplier M in Line 4 is a positive constant which can be interpreted as the maximum intensity—*i.e.*, the desired intensity for the most popular (first rank) object. Depending on the choice of N and α, one may need to set the value of this multiplier sufficiently large to ensure that the lower rank objects at the tail of the popularity distribution will also have a reasonable chance to appear in the output trace.

Once the reference intensity is determined, a call to procedure HAWKES-PROCESS is made at Line 5 to produce the actual trace of references. The parameters ρ and β respectively determine the intensity of the centers and the expected number of next-generation off-springs in the underlying cluster process. The centers are uniformly scattered throughout the region, and for each center, the off-springs are normally distributed around it. The procedures UNIFORM and

Algorithm 3.2 Method for generating a Hawkes point process

Input: Intensity of cluster centers (ρ) and the expected number of off-springs (β)
Output: A Poisson cluster process Π
Require: $\rho \geq 0$ and $0 \leq \beta < 1$

```
1  procedure HAWKES-PROCESS(ρ, β)
2      n_t ← POISSON(ρ)
3      for i from 1 to n_t do
4          Π(i) ← UNIFORM(0,1)
5      end for
6      idx ← 1
7      end ← n_t
8      while idx < n_t do
9          n_c ← POISSON(β)
10         for j from 1 to n_c do
11             end ← end + 1
12             Π(end) ← Π(idx)+ NORMAL(0,σ)
13         end for
14         n_t ← n_t + n_c
15         idx ← idx + 1
16     end while
17     return Π
18 end procedure
```

NORMAL are assumed to return coordinates in d-dimensional space with corresponding distributions and the parameters specified.

3.4.2 Caching under Non-stationary References

With a non-stationary stream of references, the popularity profile of objects in the system varies over both space and time. Consequently, the model discussed in Section 3.3 can no longer be used to analyze the behavior of the caching system. A useful insight which can help remedy this limitation is that while, in the big

picture, the underlying process of object references is dynamic, when studied at a finer granularity, it can be well-approximated as a "piecewise" stationary process. In other words, if we look at the process of references through a sufficiently small window in the time-space domain, the subprocess observed exhibits a rather stationary behavior.

Let $q_u(n,t)$ be the popularity of the n^{th} object observed by a cache node u around a particular time t. As a generalization of Eq. (3.3), we can compute this quantity as follows:

$$q_u(n,t) = \begin{cases} \lim\limits_{\Delta t \to 0} \dfrac{\mathbb{E}[N_u(n, t+\Delta t]}{\Delta t} & u \in \{\ell_1\}\,, \\ \sum\limits_{c \in \mathcal{C}_u} q_c(n,t)\, m_c(n,t) & u \notin \{\ell_1\}\,. \end{cases} \tag{3.6}$$

Here, $\{\ell_1\}$ denotes the set of ℓ_1 caches; $N_u(n, t+\Delta t)$ denotes the number of references to object n coming at node u during interval $(t, t+\Delta t)$; and $\mathcal{C}_u$ represents the set of caches which have u as their upstream node. In other words, the aggregate miss streams of nodes in $\mathcal{C}_u$ form the input stream of u.

In most scenarios, it is neither practical nor necessary to work with the infinitesimal limit given in Eq. (3.6). Rather, the input stream of a ℓ_1 cache could be partitioned into a number of smaller time bins over which the input process is assumed to be stationary. The size of the time bins does indeed depend on the degree of reference locality. The more localized the input stream, the more clustered are the occurrences of the references over time; hence, smaller time bins will be required to mitigate the approximation error.

This treatment can be used in conjunction with Che-aprx when dealing with non-stationary streams of references. The miss rates and the corresponding Etta's can be computed separately for individual intervals. The overall Etta, subsequently, can be calculated as the time-average of individually computed Etta's over specific intervals.

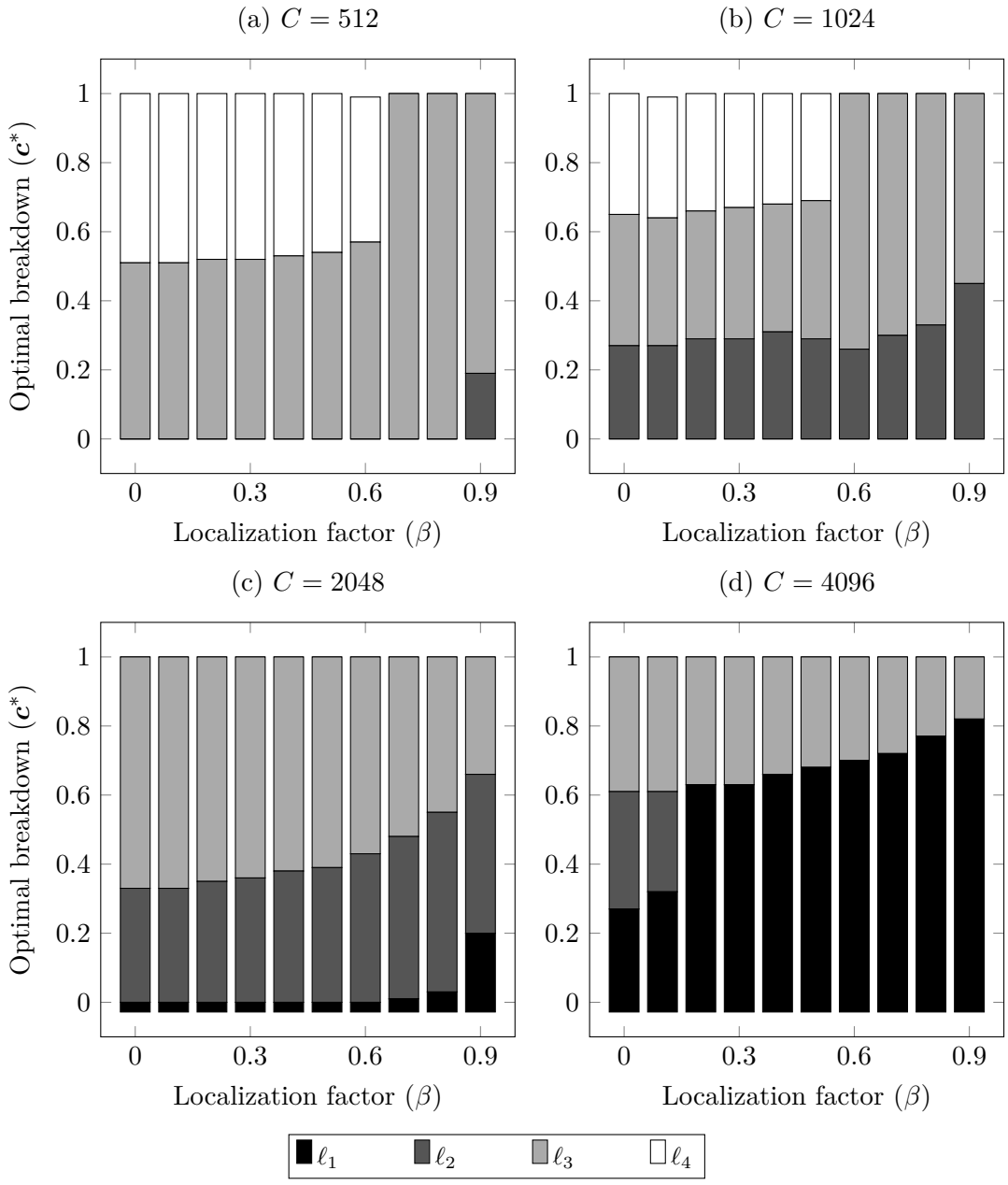

Figure 3.5: Optimal breakdown of caching budget across various levels of the tree for various localization factors

To evaluate the accuracy of this method, we conduct some Monte Carlo simulations backed by a series of discrete-event simulations performed in ndnSIM [46]. The underlying topology we consider is a complete tree of degree 4 and depth 6 with 4 layers of intermediate caches. The object catalog contains 100 files with a Zipf popularity profile of parameter 1. Algorithm 3.1 is used to generate a 2-D trace of object references with various degrees of localization. Leaves of the tree span across one dimension and object references are directed at their L_1-closest cache. The other dimension captures the temporal distribution of references as discussed before.

Figure 3.5 demonstrates the optimal breakdown of cache budget across levels of the tree for various degrees of reference locality. Again, a drift towards the edge can be observed as the available caching budget increases. This transition is further accelerated with a larger localization factor. Figure 3.5(d) reflects this phenomenon more vividly. In particular, we observe that an increased localization factor from 0.0 to 0.9 has almost the same impact on the performance of the caching hierarchy as doubling the caching budget does.

Figure 3.6 compares the overall expected time to access for edge- vs. optimal caching with the same configurations as plots in Figure 3.5. As discussed earlier, for numerical analysis, we split the time into smaller non-overlapping intervals (bins). With zero localization, references are generated independently. The generated trace, therefore, conforms to the IRM assumption and hence, one single time bin is considered. With a localization factor of 0.9, I found that 5 time bins yield a good approximation with a maximum error of 6% over a wide range of configurations. For other cases in between, the number of bins are chosen proportionally. For the general case, the number of bins should be chosen such that the approximation error is minimized. Finding the optimal number of bins requires

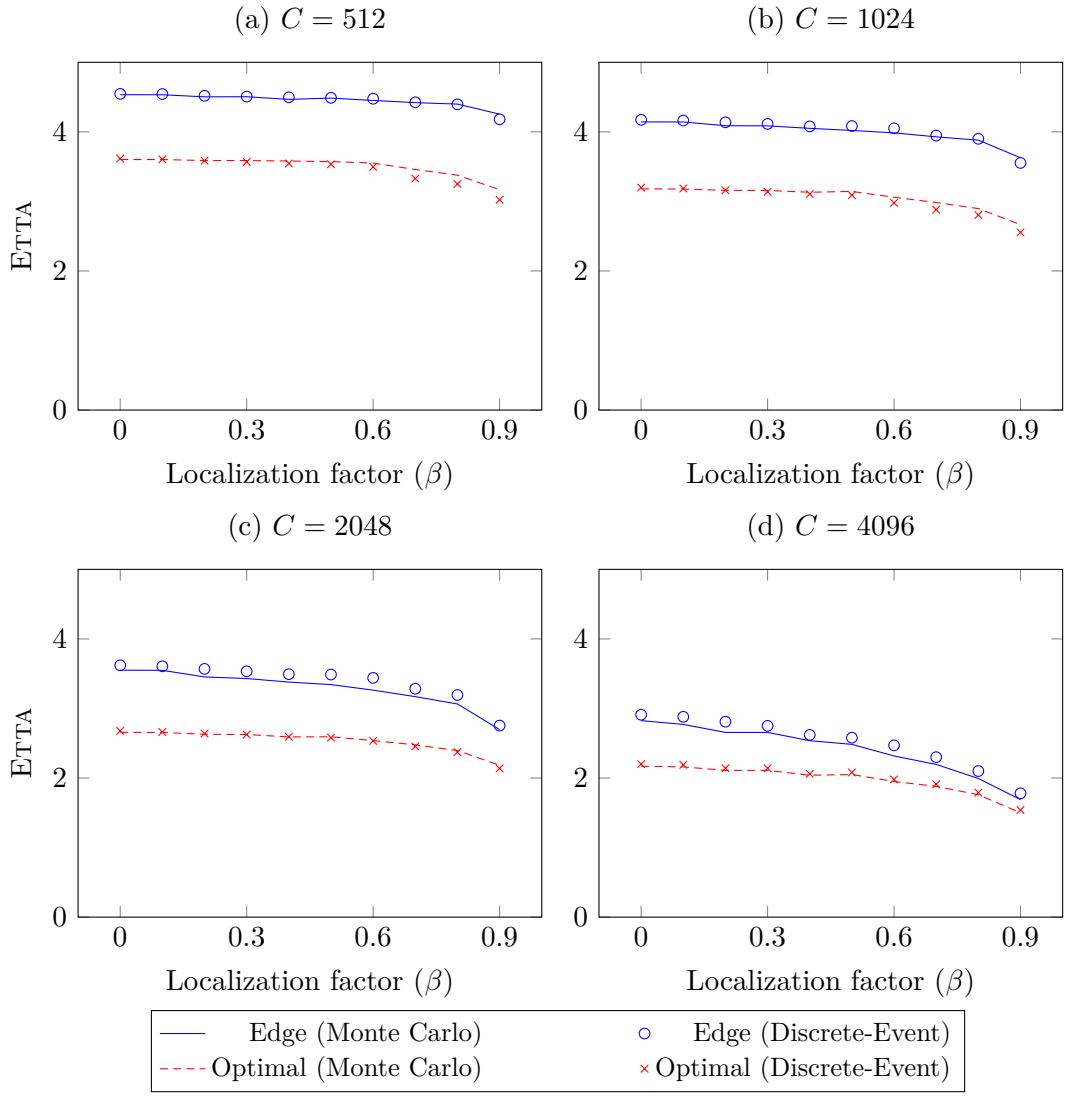

Figure 3.6: The overall expected time to access (ETTA) with respect to the localization factor for optimal- vs. edge-caching

a deeper understanding of the behavior of the underlying point process, and we leave this as a subject for future research.

Figure 3.6 also captures how fast optimal on-path caching converges towards the edge as both the caching budget and localization factor increase. In the examples depicted, the maximum difference between the two schemes is around 35%. This is reduced to 8% on the far right of Figure 3.6(d) where each ℓ_1 cache has capacity to store 16 content objects.

The main focus of our study so far was primarily on a well-defined hierarchy of caches. In what follows, we broaden the scope of these findings by considering a more arbitrary topology for cache networks.

3.5 Caching on Random Networks

Let Π_0 be a point process on d-dimensional space representing a random deployment of cache routers. We define the local cache of a sub-region in the space of interest as follows.

Definition 3.2. *A node $x \in \Pi_0$ is said to be the local cache for the region $\mathcal{C}_x$ defined as:*

$$\mathcal{C}_x = \left\{ y \in \mathbb{R}^d : \|y - x\| \leq \inf_{x' \in \Pi_0, x' \neq x} \|y - x'\| \right\} .$$

In fact, $\mathcal{C}_x$ comprises the closed set of points that are geographically closer to x than any other point in Π_0. In this sense, the process Π_0 forms a Voronoi tessellation of the space similar to the construction depicted in Figure 3.7. The solid dots are the points generated by Π_0 and the polygons where they reside are the Voronoi cells. We shall refer to $\mathcal{C}_x$ as the cell of node x, hereinafter.

Each cache node is equipped with two types of storage. One type is used to permanently store content originally published by the cache node. The other type

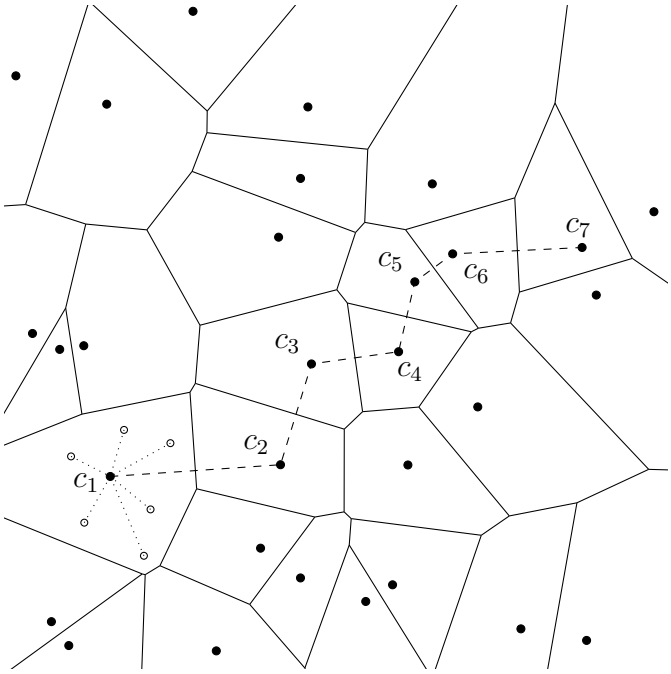

Figure 3.7: A Voronoi tessellation of the terrain via randomly deployed cache nodes. Solid dots are local caches to the cells they belong to, and empty dots indicate requests directed to them (shown only for one cell for clarity). The dashed line represents a path from cell 1 to cell 7. Cache node c_7 is the original source of the content to be routed to c_1, the initial requesting cell. On this way, if the content is cached at every cache node $c_i, i = 1, \ldots, 6$, it is called on-path caching. If the content is cached only at c_1, we call it edge-caching.

is used for caching content from other nodes while the cache node serves to route the content towards some end-user/subscriber.

Subscription requests (generated by Algorithm 3.1), form a second point process. A request originating from cell $\mathcal{C}_x$ is first forwarded to x, the local cache of that cell. If x happens to have a copy of the requested object, it serves the request locally. Otherwise, it forwards the request towards the original owner of the content in a multi-hop fashion.

The connectivity among cache nodes is defined based on Euclidean distance such that every $x_i, x_j \in \Pi_0$ are connected through a bidirectional link iff $\|x_i - x_j\| \leq r$ for some constant $r > 0$. Such a paradigm is often adopted for modeling of wireless ad hoc networks and might not be well-suited to represent a typical wired topology. Still, we believe it is insightful to study the performance of various caching schemes on a more general and irregular type of topology such as that of a random geometric graph.

3.5.1 The Routing and Caching Process

The routing is performed along the shortest path connecting source-destination pairs. For simplicity, we choose the critical radius r large enough to ensure that the network is connected. Hence, there always exists at least one path connecting each subscribing cell to the publishing source. Such paths typically cross through several cells and the traffic carried along them may be cached at the local caches of the cells where they intersect.

On the way towards the source, if a valid replica of the requested object is found at any of the intermediate caches along the path, the request will be handled locally and it will not be forwarded beyond that point. However, if no cache hit occurs along the path, the request will be served by the original source and the requested object will be routed back towards the requester on the reverse path.

For a well-structured tree topology, we observed suggestive evidence that edge-caching can perform comparably close to optimal caching in certain situations. Implementing optimal caching on a random configuration, nevertheless, is challenging if not impossible at all. This is perhaps why many existing ICN proposals adopt a simplified version of on-path caching in which all routers blindly cache every piece of content they relay.

Implementing edge-caching, in contrast, is not much of a burden on a random network so long as a clear definition of "edge" is given. Since object requests are scattered all over the network, there is no physical boundary to separate edge from the rest of the network. Instead, we give a logical definition of the edge. In fact, we say edge-caching takes place if caching is only performed at the local cache of the last cell where the traffic is being served—*i.e.*, the destination cell. According to this definition, a cache router is an edge cache for the cell it resides in and a non-edge cache for the traffic it relays to all other cells. An interesting aspect to investigate is a performance comparison of the simple on-path caching—which for brevity we shall refer to as "on-path"—versus edge-caching in the above described configuration.

Although appearing different, the above-described system has many characteristics in common with the hierarchical topology we discussed in the previous section. In particular, for any given object, the original publisher (source) serves as the root of a tree. The object requesters (*i.e.*, destinations) are the leaves, which can be from anywhere within the network. Of course, the induced tree structures differ for various content objects. Consequently, a cache node can be part of several such logical trees and at different levels.

3.5.2 Simulation Results

I performed event-driven simulations on ndnSIM [46] to compare the performance of caching at the edge versus the standard on-path caching on a random geometric topology. The network consisted of 200 cache nodes distributed uniformly and at random over a region of 100×100 square units. Nodes' radio range was set to 12 units giving each node an average degree of 8.93. I considered a total of 1000 content objects with a Zipf popularity distribution of parameter 1.

The objects are also uniformly distributed among nodes which act as original publishers of the designated objects. Hence, some nodes may publish more than one data object while some others none. With the foregoing settings, the following measurements were performed in the steady-state of the system when all caches were full.

Figure 3.8 shows how the average hop-count decreases with an increased locality factor when each node has a caching storage of size 10. In this case, edge-caching outperforms the standard on-path caching even under the IRM assumption (*i.e.*, $\beta = 0$). This behavior is rather surprising because when the references are generated independently, there should seemingly exist no difference between the two schemes in terms of cache hits. However, a subtle observation is that replacements generally take place at a higher rate with on-path caching than with edge-caching. This is due to the replacements that a cache incurs while relaying traffic to other cells. These replacements do not take place when caching is only done for the edge traffic.

Not all such replacements are useful. In fact, with a heavy-tailed distribution such as Zipf, a vast majority of objects are individually unpopular and very unlikely to be requested. Yet, in the big picture, it is much more likely to see *some* object from this whole population of less popular items being referenced by *some* node within the network. With on-path caching, all such references result in replacements along the entire path serving the traffic to the destination cell. Once referenced, however, because the object is of little global interest, the odds are small that any of these affected caches serve any subsequent reference to the same object in near future. The replaced item takes up a space that could have otherwise been dedicated to a more popular item and thereby, diminishes the caching gain.

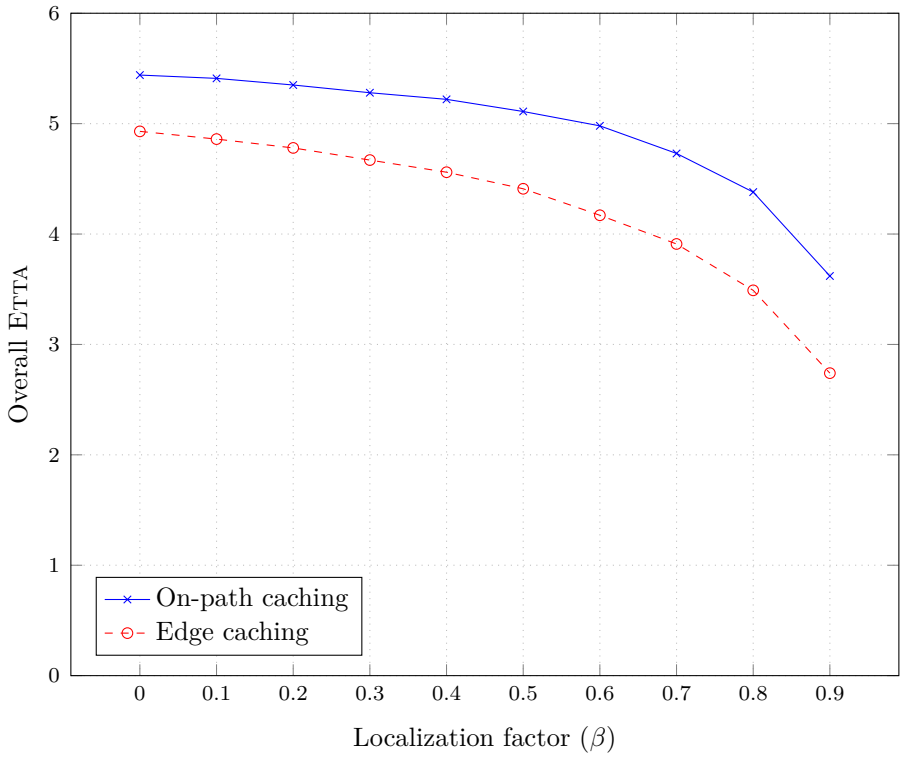

Figure 3.8: Overall expected time to access an object in a random geometric topology for various degrees of reference locality (from IRM ($\beta = 0$) to highly localized ($\beta = 0.9$))

With edge-caching, such useless replacements occur at a lower rate and the caching capacity is utilized more efficiently. The same arguments hold in case of higher degrees of locality resulting in an even sharper contrast.

Figure 3.9 illustrates the impact of increasing the caching budget on the average hop-count. As seen, edge-caching outperforms on-path caching for all cache sizes and over various degrees of locality. The enhancements attained through increasing the budget size become more pronounced with a higher degree of reference locality. Another observation is that a localization factor of 0.9 requires roughly 6 times less caching budget to yield the same overall Etta than it does under the independent reference model.

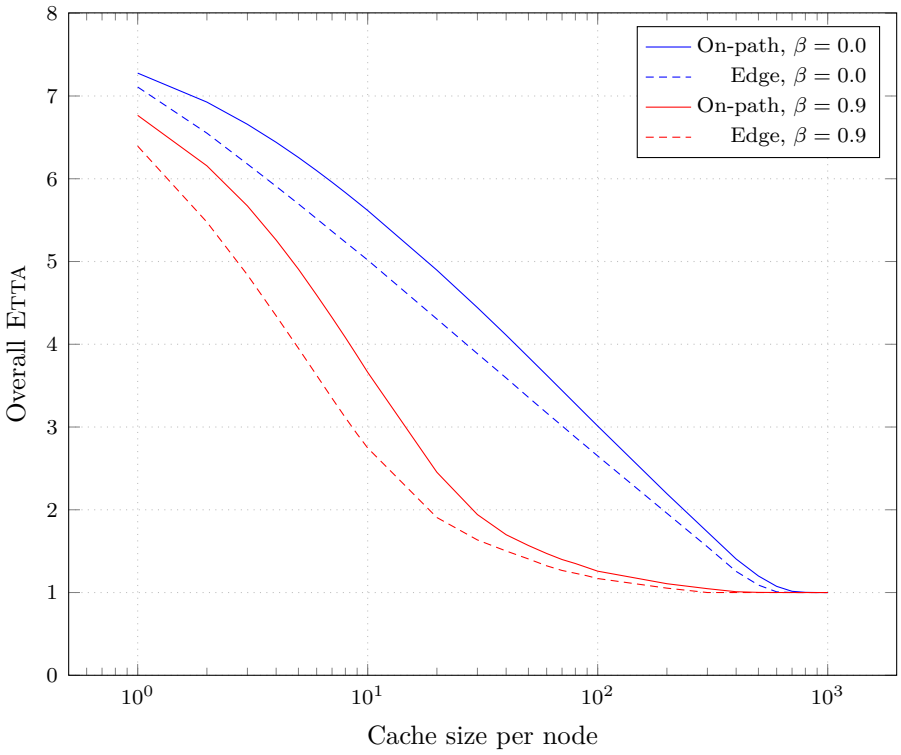

Figure 3.9: The impact of increasing the cache size on the caching gain for various degrees of reference locality. Edge-caching outperforms on-path caching in almost all scenarios.

3.6 Conclusions and Future Work

In this chapter, a computational framework was presented to compare the performance of in-network caching in ICN. In particular, we compared optimal on-path caching against the simple strategy of caching only at routers near the consumers of an ICN in terms of their overall expected time to accessing content objects. The results using the commonly-used independent reference model showed that while the optimal breakdown of caching budget is markedly influenced by factors such as topology and caching budget, optimal caching provides only marginal benefits over edge-caching in most scenarios.

Next, we investigated the impact of locality of reference on the performance of ICN's, and introduced a novel tool to synthesize traces of object requests that preserves their spatial and temporal locality properties. The results using this model demonstrate that, while the optimal breakdown of caching naturally drifts towards the edge with an increased caching budget, higher degrees of reference locality further accelerate this transition. This suggests that the difference between edge- and optimal on-path caching is far less than what the existing models based on the IRM assumption predict.

We also compared the prevalent on-path caching approach assumed in most ICN architectures today against edge-caching on random geometric graphs representing ICN's with irregular topologies. The results of simulations in ndnSIM [46] confirm the result from our modeling, and in fact show that edge-caching outperforms on-path caching.

The discussions in this chapter open up avenues for new research in ICN. New ICN architectures should be investigated that exploit content routing with edge-caching. It is important to broaden the scope of this study by considering more realistic types of topologies. Also, the locality model introduced should be fit against real-world traces to examine how localized content requests are in actual networks.

Chapter 4

Performance Analysis of ICN's Stateful Forwarding Plane

The Named Data Networking (NDN) and Content-Centric Networking (CCNx) architectures are Interest-oriented communication models in which users request content through sending out Interest packets. These designs require the use of Pending Interest Tables (PIT) that keep track of the state of every Interest packet traversing a router. This information is used for routing data back to the requester(s) as well as enabling optimization mechanisms such as Interest aggregation. In the past, it has been assumed that such "stateful" forwarding plane can reduce end-to-end latency and bandwidth consumption. In practice, however, it is not well understood how realistic these expectations are or how effective this design is. In this chapter, I present a thorough analysis of the stateful forwarding plane in NDN and CCNx that carefully captures the reciprocal impact of Interest hit rate at the cache and aggregation probability at the PIT in an NDN router. Based on this model, I next develop an algorithm to study the performance of large-scale content-centric networks of interconnected routers. My findings reveal that, under realistic assumptions, PIT tends to quickly expand in size while only

an insignificant fraction of Interests in the system end up benefiting from aggregation. These results call into question the use of a stateful forwarding plane in content-centric networks. The content of this chapter is published and presented in IFIP Networking 2016 [71].

4.1 Introduction

The observation that it is the content and not its location that matters to end-users has initiated a new direction in the networking research which is often referred to as Information-Centric Networking (ICN) [34, 35, 36]. In the past decade, ICN has been pursued as a unifying framework realizing the evolution from today's host-centric architecture of the Internet towards a clean-slate content-centric design for the future.

CCNx [72] and NDN [70] are two prominent ICN design blueprints. In CCNx and NDN, content consumers request content objects (CO) through *Interest* packets that are handled and processed by routers. Routers support ubiquitous content caching enabled through *content store* (CS). When a router receives an Interest packet, it first attempts to satisfy that Interest locally by looking up the CS for a cached copy of the requested content. If successful, the router sends the requested CO back to the consumer. Otherwise, the router queries a *pending Interest table* (PIT) to check if an Interest for the same CO is already being processed by the router. In that case, the PIT simply records the incoming port through which the recent Interest is received and dismisses the Interest. In this way, the PIT prevents unnecessary forwarding of similar Interests while a desired piece of content is being downloaded. This mechanism is often referred to as Interest aggregation. The idea of Interest aggregation is hardly new; it has been implemented in Web caching architectures in the past, *e.g.*, Squid [73]—where

it was referred to as *collapsed forwarding*—and commercially used on production content delivery networks since the early days of the Web.

An Interest not finding a matching record in the PIT results in the creation of a new PIT entry under the requested CO name. The router next forwards the Interest using its *forwarding information base* (FIB), which is populated by a name-based content routing protocol. The FIB enables routing to names rather than Internet addresses and specifies the best next hop to forward an Interest to based on the given name prefix.

According to CCNx and NDN, forwarding state is stored at the PIT for every Interest packet traversing the router. Therefore, existing literature [74, 75] describes the forwarding plane of NDN as "stateful." Over the years, it has been argued that Interest aggregation in such a stateful forwarding plane renders inherent multicast support, reduced load on content servers, optimized bandwidth utilization and improved end-to-end latency when combined with caching. However, all these benefits come at the price of creating, optimizing and maintaining very large data structures (*i.e.*, PITs and FIBs) which are expected to operate at line speed. The routers used in the Internet backbone handle hundreds of thousands of packets every second. It can be very challenging to design these components efficiently with fast enough lookup and update operations to not act as a source of latency and overhead themselves.

In recent past, considerable body of work has focused on the optimization and scalability of the PIT (*e.g.*, see [76, 77, 78, 79]). Nevertheless, there exists no comprehensive analytical work evaluating the effectiveness of this stateful design for NDN which is realized through the PIT. Despite some experimental efforts [80, 81] in dimensioning the PIT size, important questions such as: *What are the dynamics of the PIT size under realistic conditions? What fraction of Interests*

are subject to aggregation? How much traffic can effectively be eliminated through Interest aggregation and *whether or not these justify the use of PITs* have mainly remained unanswered to this date. A more detailed discussion of related work is provided in Section 4.2

In an attempt to answer the above questions, Section 4.3 presents comprehensive analytical models for characterizing an NDN router with CS and PIT components. Closed forms are derived for the CS hit rate, the PIT aggregation probability, as well as the router response time as functions of the popularity of content objects. In Section 4.4, I use these constructions for the analysis of a network of interconnected NDN routers. Particularly, I develop an algorithm that evaluates how the CS cache hit and Interest aggregation at the PIT jointly affect the shape of the egress traffic to other routers in the network. Through extensive event-driven simulations, Section 4.5 demonstrates how accurately the proposed framework can predict the steady-state behavior of such a complex system. Furthermore, it shows how the model can be used to study the performance of a large network under realistic conditions, such as that of today's Internet for which event-driven simulations are prohibitive.

Section 4.5 shows numerical examples quantifying the performance of the stateful forwarding plane of NDN in a network containing more than 100,000 routers and hundreds of millions of COs. The results reveal many shortcomings to the existing design of the NDN's forwarding plane. Despite the common beliefs, the results from this chapter show that: (1) the benefits from Interest aggregation are marginal for regular traffic; (2) most Interest aggregation takes place closer to sites where the content is permanently stored—hence offering minimal gain on reducing the internal traffic load; (3) the PIT size grows exponentially every level deeper inside the network; and (4) increasing the caching storage eliminates most

benefits from Interest aggregation though hardly helps the growth of the PIT size.

The insights from this analysis lead to the necessary conclusion that NDN's stateful forwarding plane must be rethought and PIT should *not* be an integral component of the future ICN architectures and content-centric networks. Network-wide content caching inherently suppresses a great number of Interests requesting similar content throughout the network without the costs of maintaining PITs. In fact, if per-Interest forwarding state is not needed for other reasons, a "stateless" forwarding plane makes content-centric networking at Internet scale more feasible than the currently existing NDN design, given that forwarding data structures (*e.g.*, CCN-DART [75, 82] and CCN-GRAM [83]) smaller and more efficient than PIT could be used for routing data back to the consumers.

4.2 Related Work

To address the problem of PIT size explosion, great [76, 77, 78, 79, 84] effort has been put in recently to design and implement fast and scalable pending interest tables, as well as to reduce the required memory to store it. Among first who initiated the study of PIT were Dai *et al.* [76]. They proposed a tree-like structure for PIT that enables fast lookup and update operations. A separate line of work is by You *et al.* [85] that take advantage of Bloom Filters in order to reduce the memory space needed for operation of the PIT. One of the first papers pinpointing the high overhead of the PIT was by Tsilopoulos *et al.* [86]. They proposed a semi-stateless forwarding scheme in which instead of tracking Interests at every hop, the state is stored by the routers every d hops on the path. Virgilio *et al.* citevirgilio:13 evaluated the performance of several existing PIT architectures in terms of resilience to overload conditions. Revealing differentiated weaknesses for every architecture, their results emphasizes the need for better

PIT management strategies. Their solution is to manage a dynamic PIT timeout that dynamically adapts to the network load conditions. The same idea was later adopted and expanded by Dehghan *et al.* [87] who proposed a thorough analysis of the PIT with dynamic timeout and its interaction with the content store.

4.3 Characterizing an NDN Router

I develop a mathematical model to characterize an NDN router with a CS enabling the caching functionality and a PIT for Interest aggregation. In an NDN router, the CS and PIT are implemented as data structures holding collections of references to content objects (CO). To characterize the dynamics of these complex data structures, we abstract away unnecessary intricacies by looking at them in their simplest mathematical form as sets of elements. The elements are references to the actual COs in the system and at any point in time, CS and PIT may potentially contain any arbitrary combination of such references.

Let $U = \{1, 2, ..., N\}$ be the universal set containing unique ID's of all COs in the system and consider a subset $S \subseteq U$. Let a Bernoulli random variable $\mathbb{1}_{\{i \in S\}}$ indicate whether an element $i \in U$ is also in S or not independent of all $j \in U, j \neq i$. Assume $\mathbb{1}_{\{i \in S\}}$ takes on values 1 with probability p_i and 0 with probability $1 - p_i$ and let $\mathcal{S}$ denote the cardinality of S. We have

$$\mathcal{S} = \sum_{i=1}^{N} \mathbb{1}_{\{i \in S\}} .$$

Given that $\mathbb{1}_{\{i \in S\}}$'s are independent Bernoulli trials, though not necessarily identically distributed, their summation $\mathcal{S}$ follows a Poisson binomial distribution whose first two central moments are as follows:

$$\mathbb{E}[\mathcal{S}] = \sum_{i=1}^{N} \mathbb{E}\big[\mathbb{1}_{\{i \in S\}}\big] = \sum_{i=1}^{N} p_i \,, \tag{4.1}$$

$$\begin{aligned}\mathrm{Var}(\mathcal{S}) &= \sum_{i=1}^{N}\sum_{j=1}^{N}\mathrm{Cov}(\mathbb{1}_{\{i\in S\}},\mathbb{1}_{\{j\in S\}}) \\ &= \sum_{i=1}^{N}\mathrm{Var}(\mathbb{1}_{\{i\in S\}}) = \sum_{i=1}^{N} p_i(1-p_i)\,. \end{aligned} \tag{4.2}$$

Consider an NDN router serving Interests conforming to the independent reference model (IRM)—that is, for each CO, Interests inter-arrival times are independent identically distributed random variables. Let h_i and a_i respectively denote the occupancy probabilities of CO i in the CS and the PIT. Replacing p_i's with h_i's or a_i's in Eqq. (4.1) and (4.2) would accordingly give the average and variance of the cache size (*i.e.*, number of COs in the CS if there is no cache size restriction) or the PIT size. Assuming that the arrival stream of Interests into an NDN router is a Poisson process, the occupancy probabilities h_i and a_i are respectively equivalent to the cache hit probability of the CS and the Interest aggregation probability at the PIT. This is due to the fact that Poisson arrivals see time averages—a.k.a. the PASTA property [88].

To derive the cache hit- and PIT aggregation probabilities, we need to take a closer look at the internal mechanisms of the router. At the packet level, the operations a router performs when handling Interest and Data packets for a particular CO i can be analyzed in periodic cycles. One such cycle is illustrated in Figure 4.1.

In Figure 4.1, the router receives an Interest for CO i on an incoming face. It first searches the CS for a cached copy of the requested CO and a *cache miss* occurs (see the first solid red comb). The router creates an entry in the PIT for CO i and forwards the Interest on an outgoing face. The router then awaits receiving the CO. It takes some duration d_i until a copy of CO i is located in the network and downloaded into the router. We shall refer to d_i as the *download delay* of CO i. Meanwhile, subsequent Interests (shown as dotted red combs) requesting

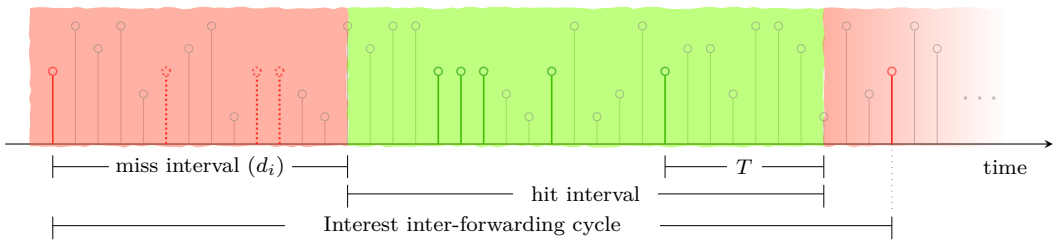

Figure 4.1: Interests arriving at an NDN router over time. Interests for different COs are distinguished by their heights. Interests for a particular CO i are drawn in color. The red/green zones indicate time intervals when CO i is absent/present in the cache. Interests drawn in red are cache misses; solid ones are forwarded and dotted ones are aggregated. It takes d_i units of time since the first miss until a copy of CO i is fetched. Interests arriving after that point will result in cache hits (green combs) so long as the CO is present in the cache. A similar cycle repeats upon CO eviction.

CO i are aggregated at the PIT because a pending entry for CO i already exists there. All incoming requests for CO i up until the time its download is complete in the CS result in cache misses. This duration which we call the *miss interval* is highlighted as the red region on the left side of Figure 4.1.

As soon as a copy of CO i is fetched and stored in the CS, the PIT entry for CO i is removed and the CO is sent out on all faces from which an Interest for CO i was received. Thereafter, subsequent Interests requesting CO i will be served directly from the CS (*i.e.*, *cache hits*) so long as CO i is not replaced in the cache. This duration is highlighted as the green region in Figure 4.1, which we call the *hit interval*. The cycle is completed with the eviction of CO i from the CS and the next Interest for CO i would initiate a new cycle.

We compute the total number of Interests handled during one such inter-

forwarding cycle assuming that the arrivals of Interests for CO i into the NDN router follow a Poisson distribution with parameter λ_i. Accordingly, denoting with $\mathbb{E}[d_i]$ the expected length of the miss interval, the expected number of missed Interests (denoted by $\bar{n}_i$) is

$$\mathbb{E}[\bar{n}_i] = 1 + \lambda_i \mathbb{E}[d_i]\,, \tag{4.3}$$

of which 1 is forwarded and the rest are aggregated at the PIT.

To compute the number of cache hits during an Interest inter-forwarding cycle we need to know the length of the hit interval. For that we leverage the notion of *cache characteristic time*, which was first introduced by Che *et al.* [49]. The cache characteristic time denoted by $T_{\mathcal{C}}(i)$ is a random variable specifying the duration of time from a last reference to CO i (while it is already in the cache) until it is replaced where $\mathcal{C}$ is the capacity of the cache (consult Figure 4.1 for an illustration). Once CO i is referenced in the cache, if the inter-arrival time of the subsequent reference to CO i is greater than $T_{\mathcal{C}}(i)$, then CO i will be replaced in the cache with some other newly referenced CO. For caches with zero download delay and large capacity $\mathcal{C}$, a useful property is that the cache characteristic times become nearly deterministic and independent of i [49]; hence index i may be omitted. Dehghan *et al.* [87] showed that this property holds even when download delays are non-zero. For brevity, we drop the subscript $\mathcal{C}$ in the rest of this chapter and simply denote the cache characteristic time by T while keeping in mind that the characteristic time of a cache does depend on the cache size.

The Interest inter-arrival times for CO i are independent exponential random variables with rate λ_i. The p.m.f. of exactly $n_i = k$ cache hits is the probability of the event that the first k Interest inter-arrival times are smaller than the cache characteristic time T, while the following Interest inter-arrival time is greater than T. This probability is defined by the p.m.f. of the geometric distribution as

follows:

$$\mathbb{P}\left(n_i = k\right) = \left(1 - e^{-\lambda_i T}\right)^k e^{-\lambda_i T}\,.$$

Hence, the expected number of cache hits is

$$\mathbb{E}[n_i] = \sum_{k=0}^{\infty} k\,\mathbb{P}\left(n_i = k\right) = e^{\lambda_i T} - 1\,. \tag{4.4}$$

From Eqq. (4.3) and (4.4), the total number of Interests for CO i handled by the router during an Interest inter-forwarding cycle is derived as:

$$\mathbb{E}[N_i] = \mathbb{E}[n_i] + \mathbb{E}[\bar{n}_i] = \lambda_i \mathbb{E}[d_i] + e^{\lambda_i T}\,.$$

Now we compute the Interest aggregation probability at the PIT as follows:

$$a_i = \frac{\mathbb{E}[\bar{n}_i] - 1}{\mathbb{E}[N_i]} = \frac{\lambda_i \mathbb{E}[d_i]}{\lambda_i \mathbb{E}[d_i] + e^{\lambda_i T}}\,. \tag{4.5}$$

And the probability of an incoming Interest resulting in a cache hit at the CS is

$$h_i = \frac{\mathbb{E}[n_i]}{\mathbb{E}[N_i]} = \frac{e^{\lambda_i T} - 1}{\lambda_i \mathbb{E}[d_i] + e^{\lambda_i T}}\,. \tag{4.6}$$

Eqq. (4.5) and (4.6) indicate what fraction of Interest packets received by the NDN router in the long run result in aggregation at the PIT or hit the CS cache, respectively. As seen, both a_i and h_i depend on the cache characteristic time T. For computing T we return to Eq. (4.1) and leverage PASTA property to write

$$\mathcal{C} = \sum_{i=1}^{N} h_i\,. \tag{4.7}$$

Eq. (4.7) can be solved for T efficiently using numerical computing environments like MATLAB and the results used to compute per-CO probabilities of PIT aggregation and CS hit through Eqq. (4.5) and (4.6) respectively. Using Eqq. (4.1) and (4.2), we can also derive the mean and variance of the PIT size (denoted by $\mathcal{P}$) as follows:

$$\mathbb{E}[\mathcal{P}] = \sum_{i=1}^{N} a_i\,, \quad \mathrm{Var}(\mathcal{P}) = \sum_{i=1}^{N} a_i(1 - a_i)\,. \tag{4.8}$$

Except the trivial case where a router is directly connected to a producer, computing the download delays for CO's can be tricky. In a network of interconnected routers this computation is more involved because the CO may be retrieved from any of the following sources:

- The original producer if the Interest does not hit any intermediate cache, nor does it get aggregated at PITs along the path to the producer.
- The CS of an intermediate router if a cache hit occurs as the Interest is being routed towards the CO producer.
- An intermediate router which does not have the CO readily available in the CS, but has a relevant entry for that CO in its PIT.

In the former case, the time it takes that router to satisfy the Interest would be d_i, while in the latter two cases, it can be any value from interval $(0, d_i]$.

We define the *pending time* of an Interest in the router as the time difference between the arrival of an Interest and the subsequent moment when the Interest is served. With Poisson arrivals, Interest arrival times are uniformly distributed over $(0, d_i]$; hence, the sum W_i of pending time of Interests during a download interval d_i can be formulated as

$$W_i = d_i + \lambda_i \int_0^{d_i} (d_i - t)\,\mathrm{d}t = d_i(1 + 0.5\lambda_i d_i)\,.$$

We define the *response time* r_i of the NDN router for a particular CO i as the expected pending time of Interests for that CO which is readily derived as

$$r_i = \frac{\mathbb{E}[W_i]}{\mathbb{E}[N_i]} = \frac{\mathbb{E}[d_i(1 + 0.5\lambda_i d_i)]}{\lambda_i \mathbb{E}[d_i] + e^{\lambda_i T}}\,. \tag{4.9}$$

Note that for the computation of the router response time knowledge of only the first two moments of download delays is sufficient.

4.4 An Algorithmic Approach to Performance Analysis of Cache Networks

In the previous section, we discussed important characteristics of an LRU cache when operating in isolation. In other words, the input into and the output from the cache were directly provided by the consumers. In this section, we investigate how the dynamics of PIT and CS in an interconnected network of NDN routers can be analyzed using the results from the previous section.

For the underlying network topology, we need a realistic configuration which is simple enough to be modeled without having to deal with complex geometric details. Also, the desired topology has to make a clear distinction between the network edge (where the consumers reside) and the network core (where the content is stored) so that the behavior of the routers at different regions of the network can separately be analyzed. The simplest topology that matches the foregoing description is a hierarchical tree structure as depicted in Figure 4.2. Consumers are located at the outermost level where their requests for objects of interest are directed to the first level NDN routers (*i.e.*, ℓ_1 routers). An ℓ_1 router searches its local CS for a copy of the requested object and if failed, it forwards the request to the next level (*i.e.*, the parent ℓ_2 router). This process is repeated on every cache miss and in the worst case, the requested object is downloaded directly from the producer at the top of the hierarchy storing permanent copies of all the objects in the system. On the reverse path back to the original requester, a copy of the object is stored in the CS of every NDN router it passes through.

For simplicity, we consider one single producer in the network. The producer in our model may conceivably represent an array of several producers at the core of the network which are collapsed into one single entity. This single-source spanning-

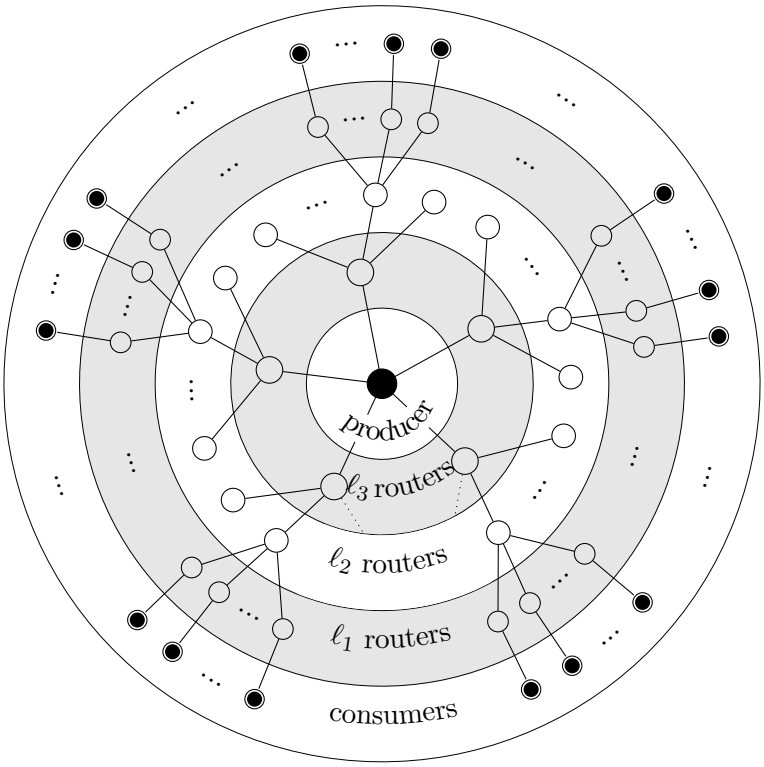

Figure 4.2: A hierarchy of interconnected NDN routers. Consumers are located at the bottom level. Their Interests go through at most L layers of content routers towards a producer as the root of the hierarchy in the center.

tree simplification of the network topology is fairly standard in the study of content delivery [89] and publish-subscribe networks [90]. It is noteworthy that more complex topologies can also be constructed by superposing several instances of this simplified topology. The symmetry of the topology makes routers of the same level share identical statistics making the model results easier to interpret and understand.

To analyze this structure, we face two critical challenges. First, the Interest stream into a higher-level router (*i.e.*, all except ℓ_1 routers) is no longer a simple Poisson process, but an aggregate of miss streams from a number of lower-level

routers. It is known, however, that the superposition of multiple streams tends toward Poisson as the load increases [91, 92]. We shall use this insight when extending the results of the previous section to the analysis of CCN networks by primarily focusing on trees of high arity (*i.e.*, node degree of 10 or more).

Secondly, chaining routers may cause circular dependencies in the computation of some router performance metrics. For instance, the cache hit probability in an ℓ_1 router depends on the download delay of the COs as mandated by Eq. (4.6). That delay is determined by the response time of the parent ℓ_2 router which in turn is a function of its input rate. The input into an ℓ_2 router itself partially relies on the miss stream of its descendant ℓ_1 routers, and that is where a dependency cycle is formed. To overcome this hurdle, I present an algorithmic approach with iterative refinements as outlined in Algorithm 4.1.

Procedure Analyze-CCN-Tree is proposed to compute the important router performance metrics discussed in Section 4.3 for a hierarchical network structure such as the one portrayed in Figure 4.2. This procedure analyzes a complete k-ary tree of height $L+2$ in which consumers are at level 0; L layers of NDN routers are employed in the middle, and the producer is located at level $L+1$ as the root of the tree. The available caching budget is provided by vector $\boldsymbol{C}$ in which element C_ℓ indicates the allocated CS capacity for each of the routers at level ℓ. The initial rate at which Interests are produced by the consumers and fed into each ℓ_1 router is λ. The object popularity profile follows a Zipf distribution as determined by the probability vector $\boldsymbol{q}$. Without loss of generality, in this chapter we always rank and index objects in their decreasing order of popularity. As such, the normalized popularity of the n^{th} ranked object is determined by the power-law

$$q(n) = \frac{n^{-\alpha}}{\sum_{u=1}^{N} u^{-\alpha}},$$

where exponent $\alpha > 0$ is the parameter of the Zipf distribution. Finally, each link

induces a round-trip delay of δ for transporting an individual piece of content. These parameters are inputs to Algorithm 4.1.

As pointed out, Algorithm 4.1 works in iterations to tackle circular dependencies. The superscript (i) used throughout the algorithm denotes the latest count of iterations. At the 0^{th} iteration, *i.e.*, the initial phase, since all caches are empty and all requests are fulfilled directly by the producer, the router response times (denoted by $\boldsymbol{r}$) are simply set based on the hop-distance of routers from the root (lines 3–5). The notation $\boldsymbol{r}_{\ell+1}^{(i)}$ describes the response time of an $(\ell+1)^{\text{th}}$ level router computed at the i^{th} iteration. Note that variables denoted in bold face are vectors with values corresponding to individual objects in the system as ordered in popularity profile $\boldsymbol{q}$.

Algorithm 4.1 Method to characterize a hierarchical CCN

Input:

k: arity of the tree;
L: number of tree levels;
λ: consumer input rate to each first level router;
δ: round-trip delay across each link;
$\boldsymbol{\mathcal{C}}$: vector of caching budget per node per layer;
$\boldsymbol{q}$: probability vector reflecting the object popularity profile.

Output:

T: characteristic time of caches at each level;
$\boldsymbol{h}$: vector of cache hit probabilities at each level;
$\boldsymbol{a}$: vector of aggregation probabilities at each level;
$\boldsymbol{r}$: vector of router response times at each level;
$\boldsymbol{m}$: vector of incoming Interest rates to each level.

```
1 procedure ANALYZE-CCN-TREE(k, L, λ, δ, C, q)
2 |  i ← 0 ,   T_ℓ^(0) ← ∞
3 |  for ℓ from 1 to L do
4 |  |  r_{ℓ+1}^(i) ← δ × (L − ℓ)
5 |  end for
```

```
6   repeat
7       i ← i + 1
8       for j from 1 to N do
9           m_{1,j}^{(i)} ← λ × q_j
10          for ℓ from 1 to L do
11              d_{ℓ,j}^{(i)} ← δ + r_{ℓ+1,j}^{(i−1)}
12          end for
13      end for
14      for ℓ from 1 to L do
15          T_ℓ^{(i)} ← Char-Time(m_ℓ^{(i)}, d_ℓ^{(i)}, C_ℓ)
16          for j from 1 to N do
17              a_{ℓ,j}^{(i)}  ← Agg-Prob(m_{ℓ,j}^{(i)}, d_{ℓ,j}^{(i)}, T_ℓ^{(i)})
18              h_{ℓ,j}^{(i)}  ← Hit-Prob(m_{ℓ,j}^{(i)}, d_{ℓ,j}^{(i)}, T_ℓ^{(i)})
19              r_{ℓ,j}^{(i)}  ← Resp-Time(m_{ℓ,j}^{(i)}, d_{ℓ,j}^{(i)}, T_ℓ^{(i)})
20              m_{ℓ+1,j}^{(i)} ← Miss-Rate(k, m_{ℓ,j}^{(i)}, h_{ℓ,j}^{(i)}, a_{ℓ,j}^{(i)})
21          end for
22      end for
23  until |T_ℓ^{(i)} − T_ℓ^{(i−1)}| < ϵ for all ℓ from 1 to L
24  for ℓ from 1 to L do
25      μ_ℓ ← 0
26      σ_ℓ^2 ← 0
27      for j from 1 to N do
28          μ_ℓ ← μ_ℓ + a_{ℓ,j}^{(i)}
29          σ_ℓ^2 ← σ_ℓ^2 + a_{ℓ,j}^{(i)} · (1 − a_{ℓ,j}^{(i)})
30      end for
31  end for
32 end procedure
```

Next, at any subsequent iteration:

1. Download delays for all levels are updated (lines 10–12) according to the heuristic that the delay for downloading files into the CS of an arbitrary router is equal to the response time of its parent router plus the round-

trip delay of the link connecting them together. Assuming all objects are unit-sized, we can deduce that the average link delays are the same. In a hierarchical structure, thus, we can compute the download delays into a particular router by knowing the average link delays and the response time of the next level (*i.e.*, parent) router.

2. All performance measures discussed in Section 4.3 are computed/updated across all tree levels (lines 14–22).

Starting from the bottom working towards the top, at each tree level the measures are computed in the following order:

Procedure Char-Time is called at line 15 to compute the cache characteristic times by solving the following fixed-point equation for variable T_ℓ:

$$\sum_{j=1}^{N} \frac{e^{m_{\ell,j}T_\ell} - 1}{m_{\ell,j}\, d_{\ell,j} + e^{m_{\ell,j}T_\ell}} = \mathcal{C}_\ell \,, \tag{4.10}$$

where $m_{\ell,j}$ and $d_{\ell,j}$ respectively denote the input rate and download delay of CO j at a level ℓ router. Note that Eq. (4.10) is in fact the expanded form of (4.7) using (4.6).

Procedures Agg-Prob, Hit-Prob and Resp-Time are called at lines 17–19 to use the above computed characteristic time for computing the PIT aggregation probability, CS hit probability and router response time for each individual CO according to Eqq. (4.5), (4.6) and (4.9), respectively.

Procedure Miss-Rate is called at line 20 to compute the aggregate input rate of any CO j *into* the next level (*i.e.*, parent) router using the above computed hit- and aggregation probabilities according to the following relation:

$$m_{\ell+1,j} = k \cdot m_{\ell,j}\big(1 - h_{\ell,j}\big)\big(1 - a_{\ell,j}\big) \,. \tag{4.11}$$

In essence, Eq. (4.11) reflects the fact that the input stream of a router at level $(\ell + 1)$ is the superposition of k miss streams from its descendant level-ℓ routers. The only exception are ℓ_1 routers whose inputs are directly provided by consumers according to line 9.

To better understand the dependency between these procedure calls, the diagram in Figure 4.3 provides a pictorial view of their relationship. At the initialization phase, all caches are empty and the download delays of CO's into the edge routers are computed based on the hop distance to the producer. The input rate into the edge routers is also the rate of Interests produced by the consumers. This information is used to compute the cache characteristic time for all ℓ_1 routers. Cache hit- and PIT aggregation probabilities as well as router response times are then computed for ℓ_1 routers. Next these results are used to calculate the input rate and download delays for ℓ_2 routers. Then level 2 becomes the current level and a similar procedure is repeated for all remaining levels from the bottom to the top of the tree.

The computations in the middle and bottom boxes in Figure 4.3 are repeated in consecutive iterations as needed; the results from one iteration will be used to refine the next as the computed measures gradually converge to their steady-state values. This is handled by the condition in line 23 of Algorithm 4.1. The convergence is determined by the variations of the cache characteristic times. In fact, when the absolute difference between characteristic times of caches in consecutive iterations drops below a certain pre-specified threshold ϵ, the main repeat loop stops. In my numerical simulations—to be discussed next—I noticed that the first few iterations usually suffice to get an accuracy of better than 0.1% while no more than 10 iterations were needed in all cases studied (irrespective of the input size).

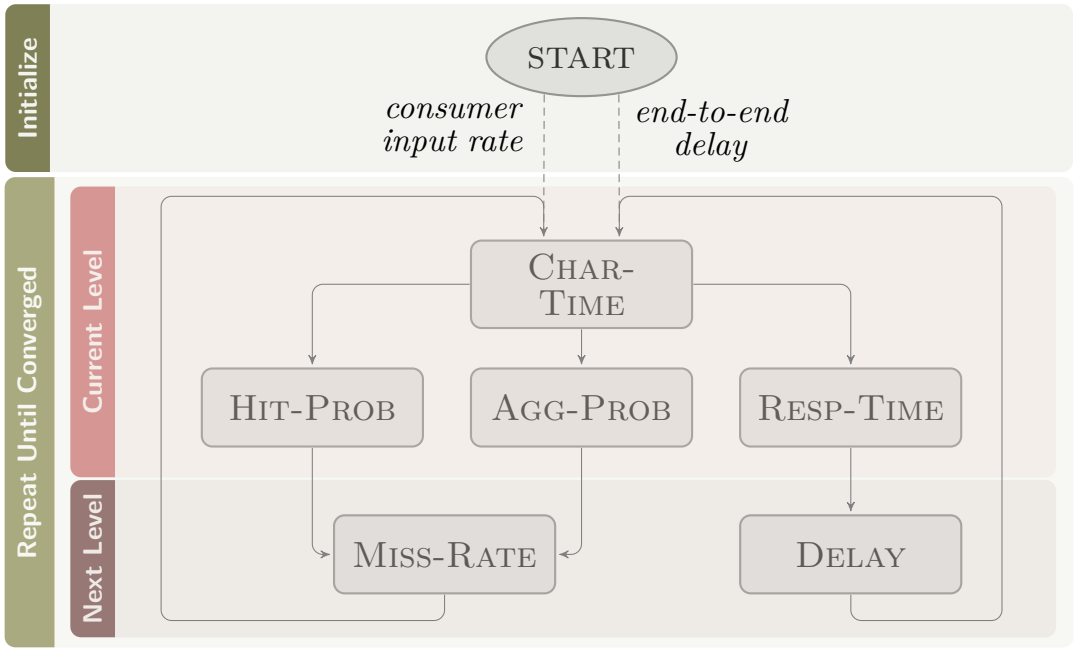

Figure 4.3: Dependency among procedure calls in Algorithm 4.1.

Finally, the Interest aggregation probabilities are used (lines 24–31) to compute the mean (μ) and variance (σ^2) of the PIT size at various levels according to Eq. (4.8).

Implementing Algorithm 4.1 is straightforward in many off-the-shelf numerical computing environments. In my simulations, for solving Eq. (4.10) I leveraged `fsolve` function from MATLAB's Optimization Toolbox which uses trust-region methods [93] for solving systems of nonlinear equations. It is known [94] that trust-region methods take $\mathcal{O}(\varepsilon^{-2})$ iterations to drive the norm of the gradient of the objective function below desired threshold ε. The time-complexity of Algorithm 4.1 is hence $\mathcal{O}(NL\,\varepsilon^{-2})$.

4.5 Performance Evaluation

In this section, I present simulation results to show how the proposed method can be used to accurately predict the complex behavior of a large network of interconnected content routers. First, a detailed comparison is provided of the numerical results of the presented model versus the results from extensive event-driven simulations in ndnSIM [46]. Next, the results from my model are used to analyze more complex scenarios, such as networks with much larger content base and many more content routers, for which performing event-driven simulations is extremely resource- and time-intensive. The objective is to show through numerous examples how the current stateful design for the NDN's forwarding plane performs in a large scale.

In the experiments to be discussed next, we shall focus on two strategies of cache allocation, namely *uniform caching* and *edge caching*. In the former, a fixed caching budget is evenly distributed among all content routers, whereas with the latter, the budget is entirely allocated to the routers at the edge of the network, *i.e.*, ℓ_1 routers. With edge caching, the upper level routers act as routers with no caching capability (that is, their CS size is set to zero); yet they perform normal Interest aggregation upon receiving Interests for which they have pending entries in their PITs.

4.5.1 Comparison of Model with Event-driven Simulations

We consider a tree of degree $k = 10$ and height $H = 5$ as the underlying topology, where $L = 3$ levels of content routers are employed in the middle. The reason for using such configuration is to keep the overall aggregate traffic pattern in the middle layers as close to being Poisson as possible, as discussed in Section 4.4. Although the model was able to capture the overall trends in our experiments with

trees of lower arity, we noticed that more accurate results are generally obtained when nodes have higher fan-in (*e.g.*, 10 or more). This assumption, however, is not unrealistic as some studies [95] of the actual Internet router-level topology have reported an average degree of more than 22 per router.

When performing event-driven simulations, I found that with a large content base the system takes much longer to come to a steady-state; while increasing the caching budget worsens the situation. In such case, a large number of requests must be used just to "warm-up" the system—hence, not to be used for collecting statistics—from the initial state where all caches are empty. Besides, because of the Zipfian nature of CO popularity, a larger number of requests must be generated in total to ensure that COs at the long tail of the distribution also get a reasonable chance to appear in the generated stream of Interests. Hence, for the first set of experiments, I decided to use a small content catalog comprising only 100 COs. Even for such a small content catalog with a Zipf parameter of 1, I had to generate roughly 4 million requests—while disregarding the first half—to make sure all caches in all levels have their capacity almost fully utilized before collecting statistics.

For the foregoing set-up, Figure 4.4 compares the Interest aggregation probability for individual COs as predicted by model versus the results from extensive event-driven simulations. Curves in each plot represent the PIT aggregation probability as attained by *each* of the content routers at the corresponding level. Thanks to the symmetry of the topology, all routers at the same level share similar statistics. Graphs in the top row contrast uniform caching against edge caching employed in graphs at the bottom. In each row, the total caching budget (CB) increases from left to right and top to bottom. The model accurately predicts aggregation across various caching levels even at a fine object-scale resolution.

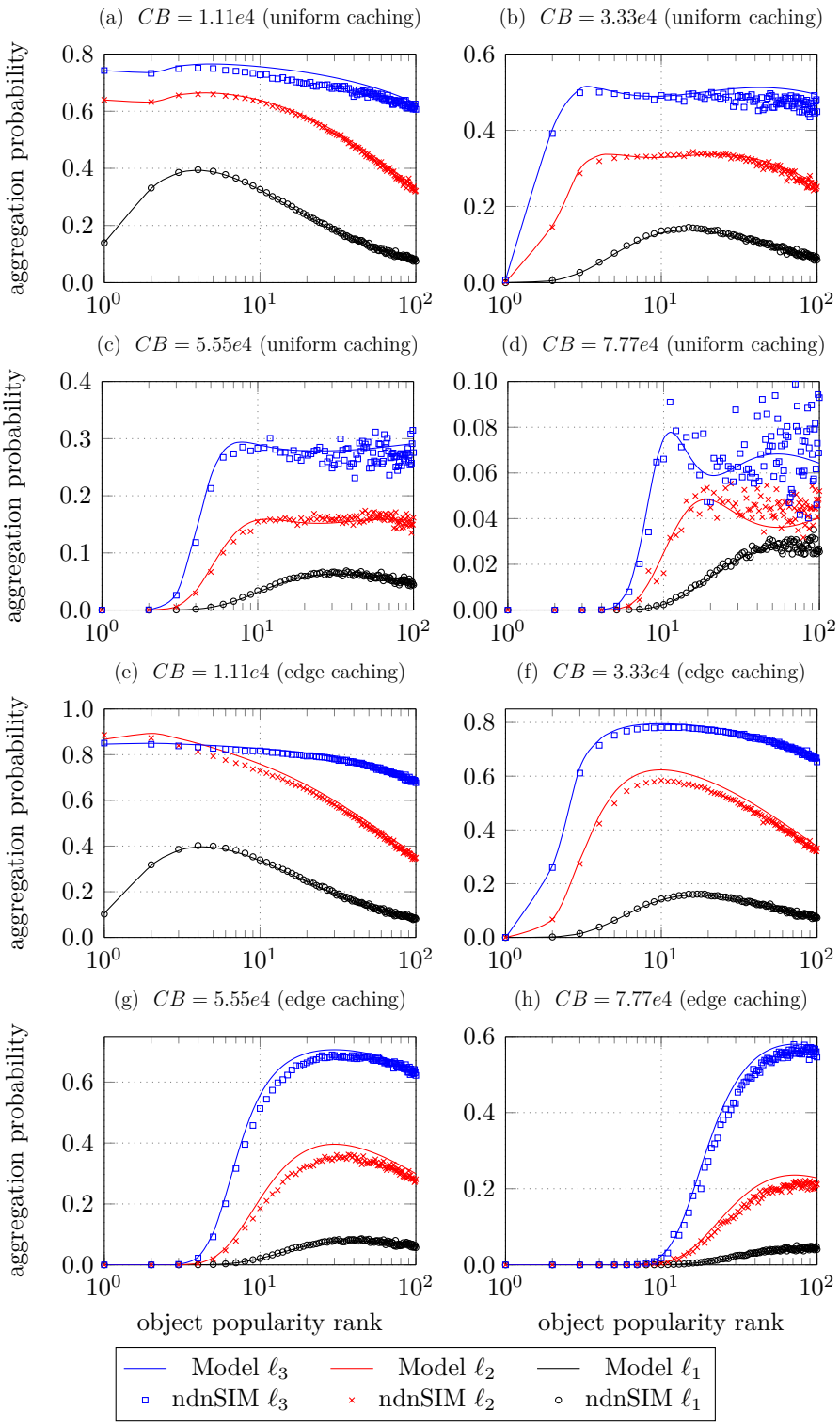

Figure 4.4: A comparison of model versus event-driven simulations. Input rate into each edge router is 100 Interests/sec. Model predicts aggregation probability for individual objects fairly accurately across all levels of the tree topology.

As seen, edge caching generally results in higher aggregation probability at higher levels. This behavior is expected because with edge caching naturally no cache hit may occur at higher levels in the tree. Hence, many requests that would have hit content stores if a non-zero cache size were used now end up getting aggregated at PITs.

To obtain a more insightful view of Interest aggregation, graphs in Figure 4.5 show the odds of a generic Interest (irrespective of the object popularity rank) getting aggregated at each level of the tree when the link delay gradually increases. It is clear that at a fixed Interest rate, an increased link delay naturally improves the aggregation probability. However, larger cache sizes tend to offset some of these improvements, more noticeably with uniform caching strategy.

Interest aggregation occurs at a higher probability at upper levels of the tree. This can be attributed to the higher input rate into those levels considering the fact that the aggregate miss stream from many lower level routers constitutes the input of their parent router. Results from Figure 4.6 suggest that significant benefits are likely to accrue from Interest aggregation; however, this promising gain should be taken with a grain of salt due to the reasons discussed in the following.

First, the small object catalog consisting of only 100 COs naturally gives rise to a higher frequency of similar Interests arriving at the router, thereby an increased aggregation probability. Despite the event-driven simulation which turns out to be extremely tedious especially for a large number of COs, numerical simulations using the proposed model are practicable even on commodity hardware. Our numerical results in the next subsection confirm that the actual benefits of Interest aggregation are indeed much smaller in reality.

Secondly, the notion of aggregation probability itself may give a magnified

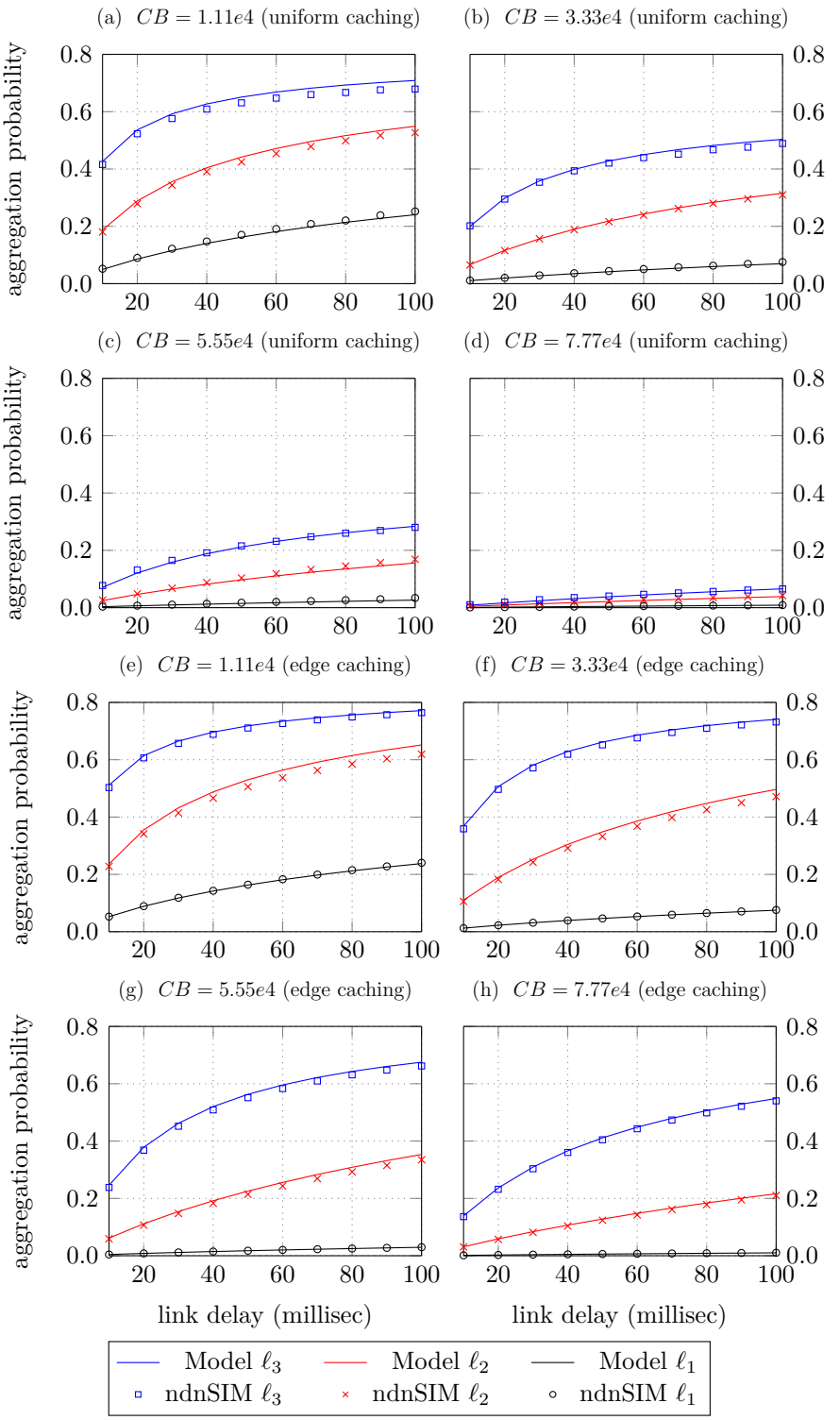

Figure 4.5: Interest aggregation probability at various router levels as a function of link delay for increasing cache sizes (left to right) and different cache allocation strategies (top vs bottom rows). Input rate into each edge router is 100 Interests/sec.

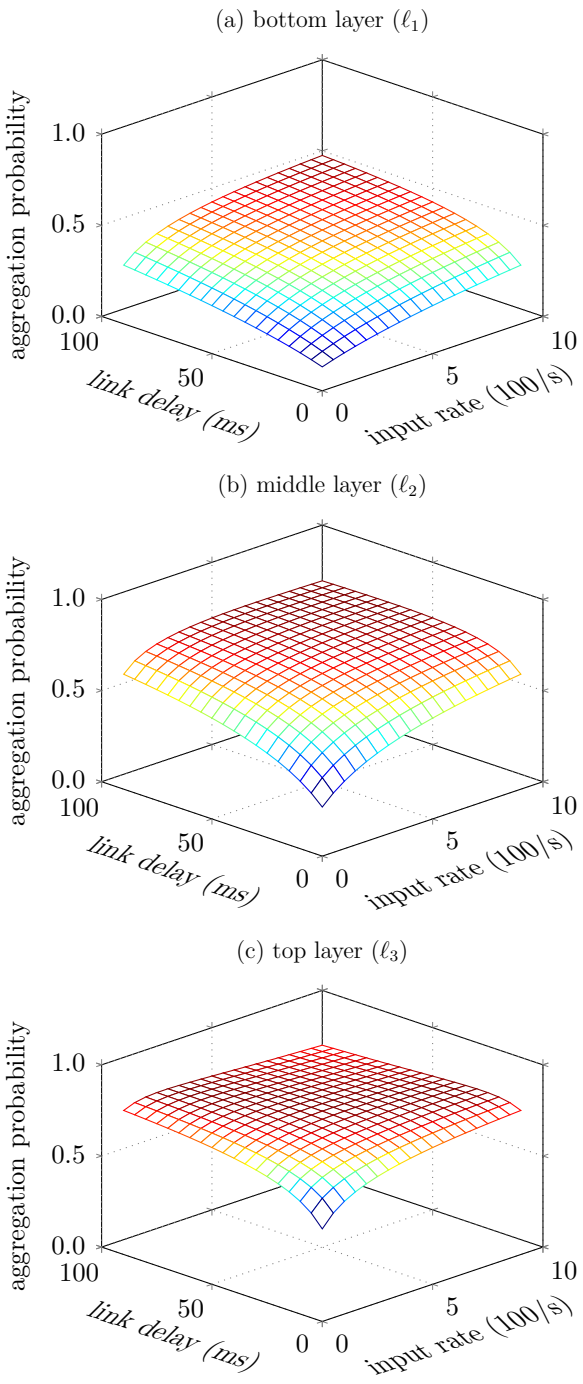

Figure 4.6: The combined impact of link delay and input rate on aggregation probability. Increasing one factor has exactly the same effect on Interest aggregation as increasing the other.

image of the real benefits. In fact, aggregation probability at a certain level in the hierarchy indicates what fraction of Interests making it up to that level end up getting aggregated. Since the request stream observed by the higher level routers is a "filtered" version of the input stream to their descendants, it is clear that fewer Interests are received in total at each level towards the top of the hierarchy. For this, we define a new measure called *aggregation percentage* that determines the percentage ratio of the count of aggregated Interests at a certain level (or at a particular router) over the total count of produced Interests in the whole system. Since every generated Interest can be aggregated at most once along its path towards the producer, aggregation percentage provides a more representative and unbiased measure. From now on, aggregation percentage shall be used in lieu of aggregation probability to assess the real benefits of Interest aggregation.

Given the foregoing remarks, I emphasize that the results demonstrated in Figures 4.4 and 4.5 are particularly meant to verify the accuracy of my proposed analytical framework, and to provide a side-by-side comparison of how varying different parameters affects the relative odds of Interest aggregation. The true benefits of Interest aggregation are discussed in the following subsection, where more realistic input parameters are used.

4.5.2 Numerical Evaluations of Interest Aggregation

As previously discussed, Interest aggregation rate is highly dependent on two key factors, namely, download delay and input rate. The combination of these parameters plays an essential role in the dynamics of the PIT. To understand the co-impact of these factors on the probability of Interest aggregation, Figure 4.6 provides some useful insights. The symmetry of plots in Figure 4.6 suggests that varying the link delay or the input rate regulates the overall trend of aggregation

probability in the same exact way. In other words, doubling the input rate for a fixed link delay has the same effect on the aggregation probability as keeping the input rate fixed and doubling the link delay. Therefore, we define *system load* as the product of these two quantities on which we build our next set of experiments. As a combined metric, system load does not identify a specific delay or input rate, rather defines an infinite range for these parameters. For example, a system load of 10 may imply an input rate of 100 Interests/sec with link delay of 0.1 seconds, or equivalently, an input rate of 500 Interests/sec with link delay of 0.02 seconds.

For the experiments to be discussed next, Table 4.1 summarizes the general configurations unless specifically stated otherwise. We consider a tree topology of degree 10 with height 7 where the five middle levels are NDN routers—a total of more than 100,000 nodes. The total number of objects considered, *i.e.*, 140 million, is an estimate [96] of the total number of videos on YouTube in 2008 and the Zipf parameter of 0.8 is taken from empirical studies [66, 97] of real content networks. The input rate of 100,000 Interests/sec and link delay of 15 milliseconds are also chosen such that the average generated traffic in the network is comparable with the load experienced by the Internet's backbone routers [98, 99].

Figure 4.7 shows the probability of Interest aggregation at each tree level as a function of system load. Contrary to the results in Figure 4.5, a side-by-side comparison of uniform- (Figure 4.7(a)) vs. edge-caching (Figure 4.7(b)) reveals that when the object catalog is large, there is no remarkable difference between these two cache allocation strategies. It is interesting that even with the highest system load of 3000, the maximum aggregation probability observed at the third level routers is less than 0.1. This seven-fold degradation compared to the results of Figure 4.5 highlights the importance of the size of the object catalog in the overall odds of Interest aggregation.

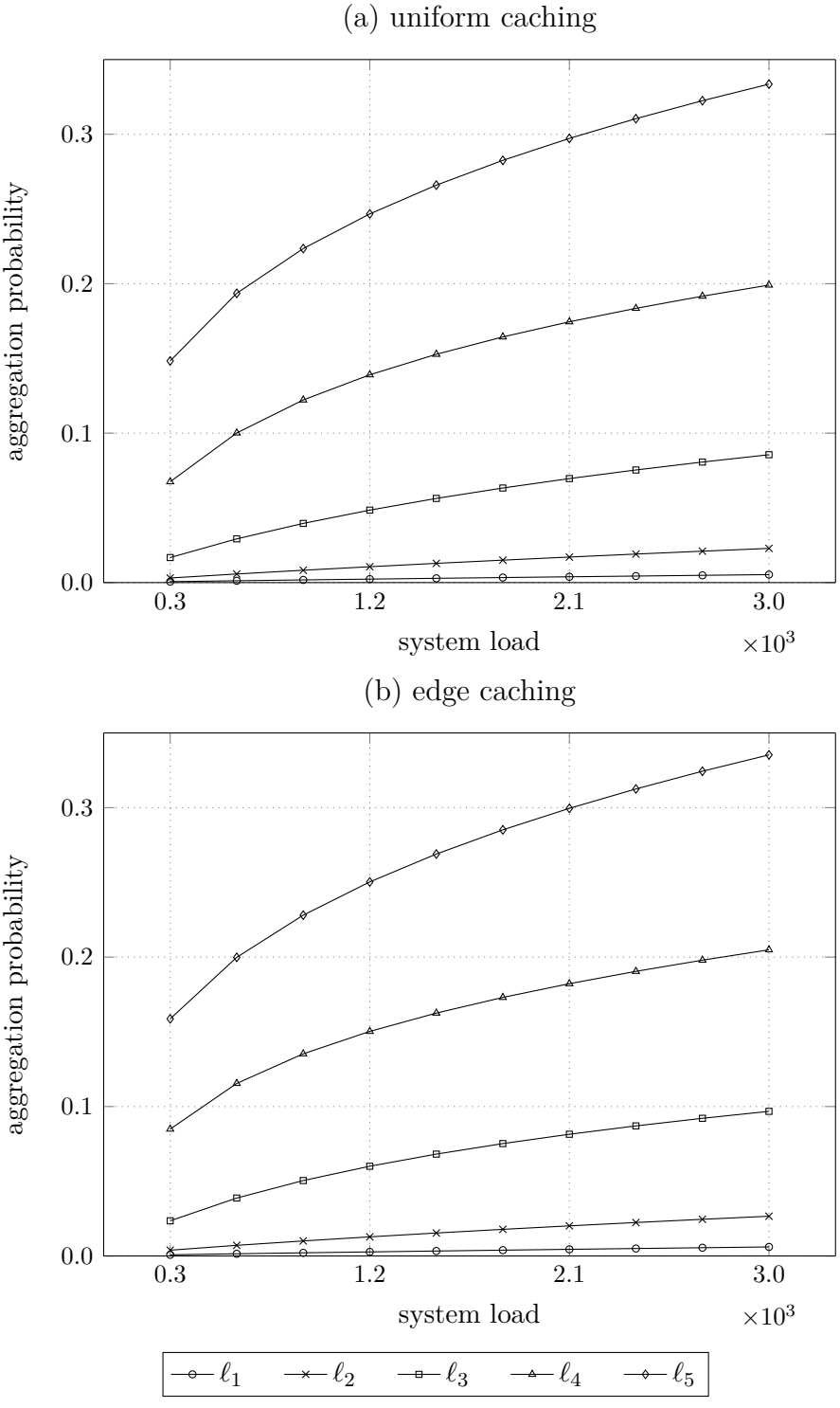

Figure 4.7: Impact of system load on the aggregation probability

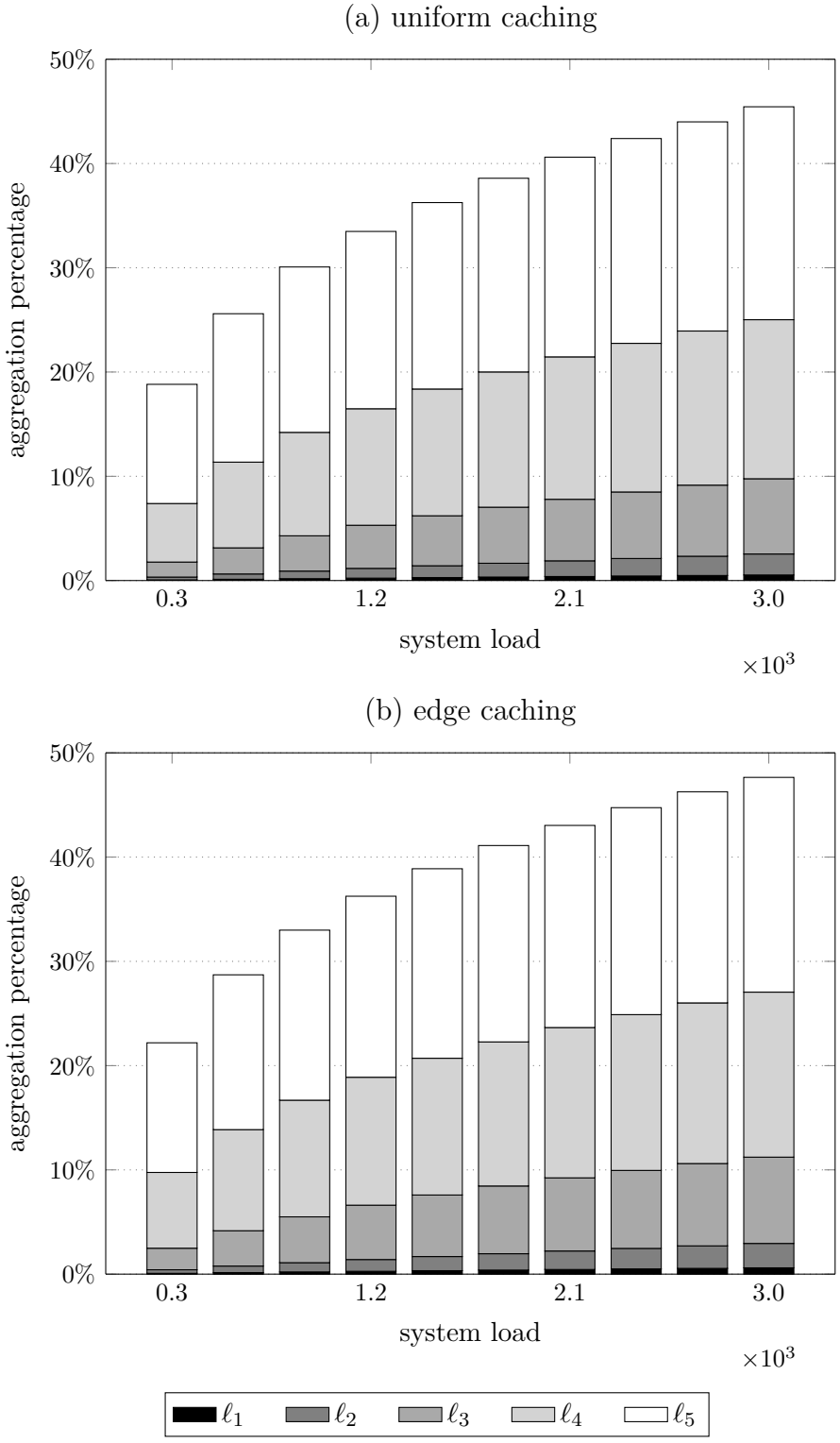

Figure 4.8: Impact of system load on cumulative aggregation percentage

Table 4.1: Table of default values for simulation parameters

Parameter	Symbol	Value
Tree height	H	7
Number of cache layers	L	5
Node degree	k	10
Total number of objects	N	140 million
CS capacity per cache node	C	100,000 COs
Zipf exponent	α	0.8
Input rate into each edge cache	λ	100,000/sec
Link delay each way	d	15 milliseconds

This rather surprising finding can be explained as follows. Given the Zipf distribution of the objects popularity, a highly popular object is requested frequently. Hence, once such an object is downloaded into the CS, due to the frequent references to it, it stays there for a long time. Interests for that object, therefore, mostly result in cache hits and are rarely aggregated. On the other hand, Interests for an unpopular object (in the long tail of the distribution) are received so sporadically over time that the odds of them co-occurring in the short time span when the router is awaiting the content are negligible. As a result, aggregation only occurs for those Interests which are neither so popular nor so unpopular—*i.e.*, those at the middle portion of the popularity distribution.

To obtain a better understanding, Figure 4.8 draws the cumulative percentage of aggregated Interests in the system against an increasing system load. Evidently, the overall percentage of Interests aggregated varies from around low 20% under light load to high 45% under heavy load, while the dominant majority occur at the top two layers, and the cumulative rate of Interest aggregation at the lower

three layers of the hierarchy is capped by 10%. Note that for these results each cache node has capacity to store only 0.07% of the entire object catalog.

Increasing the cache size offsets the Interest aggregation gain by improving the overall cache hit rates. As Figure 4.9 suggests, with little caching capacity, sizable gain (around 50% total) can be attained through Interest aggregation. Evidently, with small cache size, all layers in the hierarchy contribute to the aggregation percentage; however, as more cache is added to the routers, most Interest aggregations occur at the upper layers, while aggregation percentage at the edge decays to zero more rapidly. With a cache size of one million COs per router (*i.e.*, 0.7% of the size of the content base), although the cumulative aggregation percentage remains around 20%, almost none of it is contributed by the first three layers.

Figure 4.10 captures the impact of the object popularity distribution on the cumulative percentage of aggregated Interests. The non-monotonic trend of curves in Figure 4.10 exhibits a diminishing returns type of effect. To explain this behavior, we note that with a Zipf popularity distribution, COs can heuristically be categorized into two groups, namely, an unpopular majority and a popular minority. The Zipf parameter (α) controls the relative size of each group as well as the skewness of the distribution. In fact, the larger the Zipf parameter, the smaller the proportion of the minority group, and the greater their popularity intensity. The latter renders a higher access frequency to the COs in the popular group as α increases, hence a higher aggregation rate for them. However, as α increases and the proportion of the popular COs shrinks, the higher access frequency also results in higher cache hit rates, because many of those COs eventually find their way into the caches; therefore, subsequent requests for them no longer get aggregated. On the other hand, thanks to their diminishing popularity, the Interests

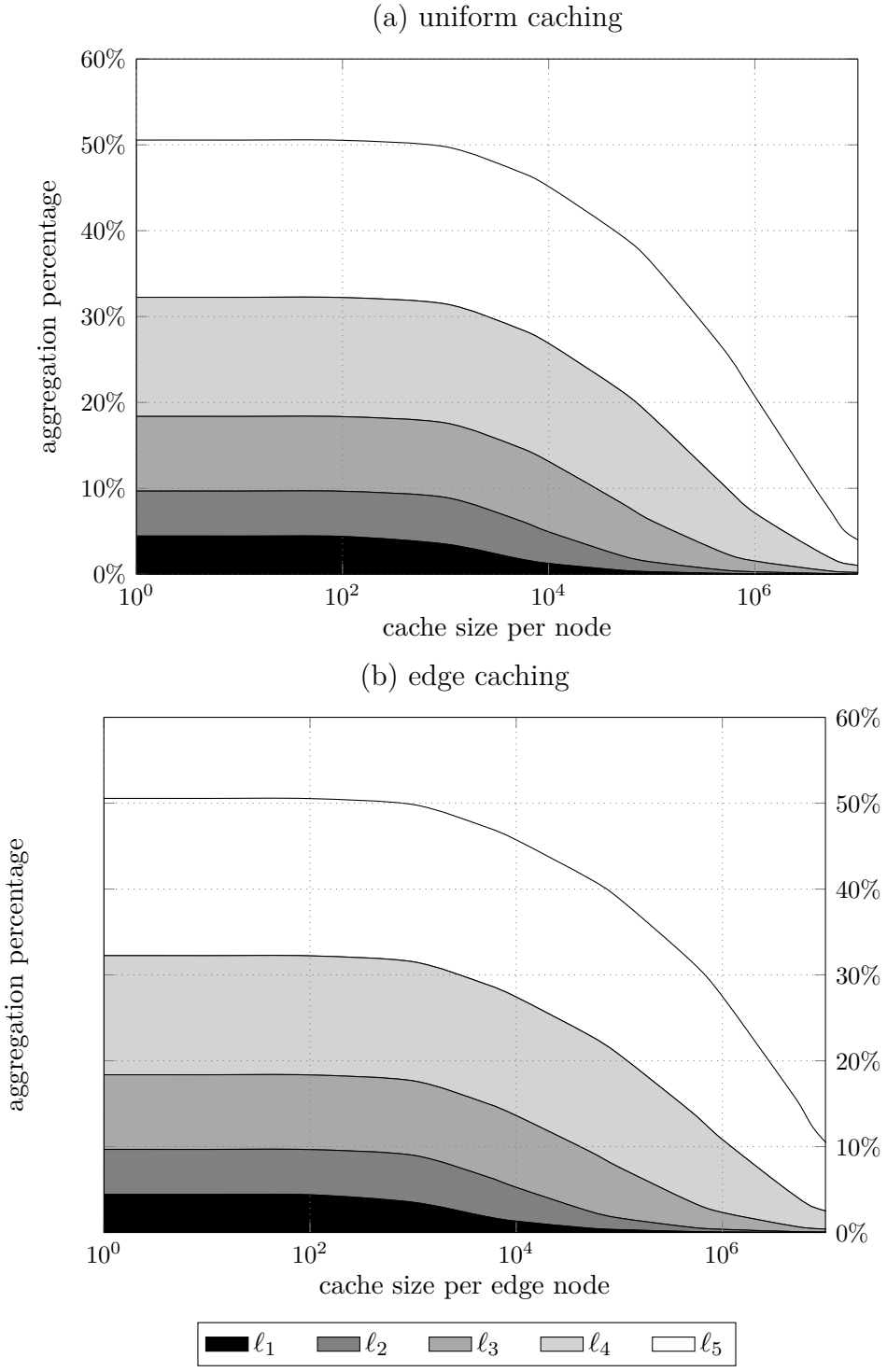

Figure 4.9: Impact of cache size on overall aggregation percentage

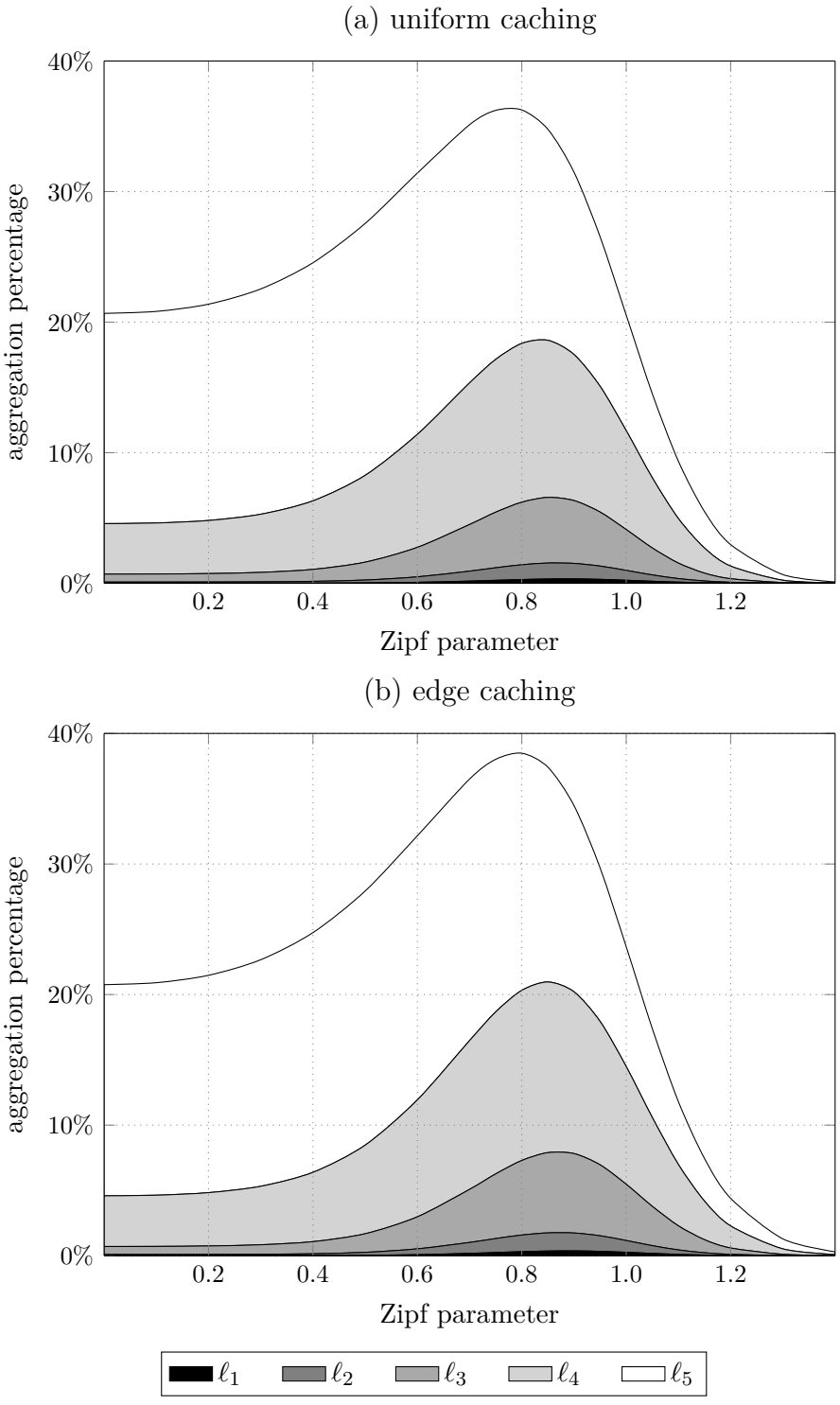

Figure 4.10: Impact of popularity distribution on aggregation percentage

for the majority group are also becoming so sparsely dispersed that the odds of finding a relevant entry for them at the PIT becomes negligible. For this, only a small fraction of Interests representing those fairly popular objects which may not have found a free spot in (limited-size) caches remain subject to aggregation. Further increasing the Zipf parameter shrinks down the size of the popular group gradually such that at some point, every one of them finds a permanent place in all caches. Thenceforth, the probability of aggregation becomes effectively zero.

From another viewpoint, Figure 4.10 also provides suggestive evidence that even under a non-stationary content popularity distribution, no remarkable benefit can be anticipated from Interest aggregation. For example, when object references are temporally localized, a CO becomes highly popular over a certain epoch, while its popularity gradually vanishes over time as other COs gain popularity. In that case, if the object popularity is measured within smaller discrete time windows, each piece can independently be approximated with a Zipf distribution with a possibly different parameter. As Figure 4.10 suggests, irrespective of how different the popularity profile looks like, only an insignificant number of Interests may be aggregated particularly at the first few layers; hence, the benefits of Interest aggregation would still remain minimal, with the maximum aggregation taking place around a Zipf parameter of 0.8.

As evident from Figures 4.8, 4.9 and 4.10, a cumulative aggregation percentage between 20% to 50% can be expected leveraging the stateful forwarding plane of NDN. However, as clearly seen, most of these aggregations actually take place near the core of the network. Aggregating Interests closer to the producers is less effective in terms of preventing network congestion and reducing delay. In order to draw more informed conclusions, the following results summarize the benefits of Interest aggregation in terms of expected number of hops Interests

save towards the producer compared to the case where PIT is not used and no Interest aggregation is performed.

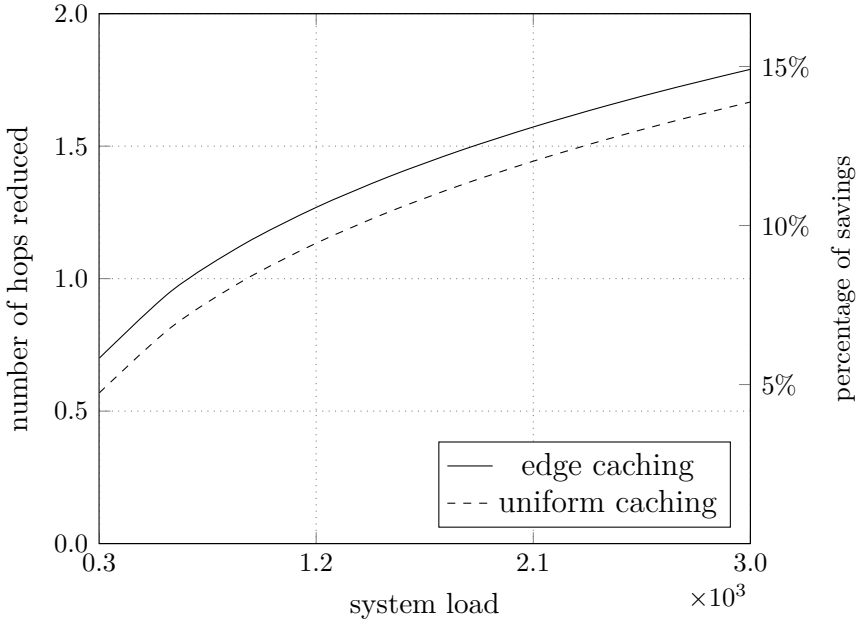

Figure 4.11: The effect of system load on reducing hop-count through Interest aggregation

In Figures 4.11, 4.12 and 4.13, the left vertical axes show the average number of hops reduced while the right vertical axes represent the corresponding percentage of savings taking into account the length of the path from the consumers to the producer. These quantities are calculated based on the hop-distance of aggregated Interests from the producer. Clearly, Interests aggregated closer to the edge save more hops compared to those aggregated near the core. Hence, as Figure 4.11 shows, the 20% to 45% rate of Interest aggregation in Figure 4.8 under varying system load translates to an average hop-count reduction of only 5% to 15%. Likewise, Figure 4.12 shows that with caching capacity for 100,000 COs (*i.e.*, 0.07% of the size of the content base) per router, aggregated Interests save on average less than 1.5 hops on their path from the consumers to the producer

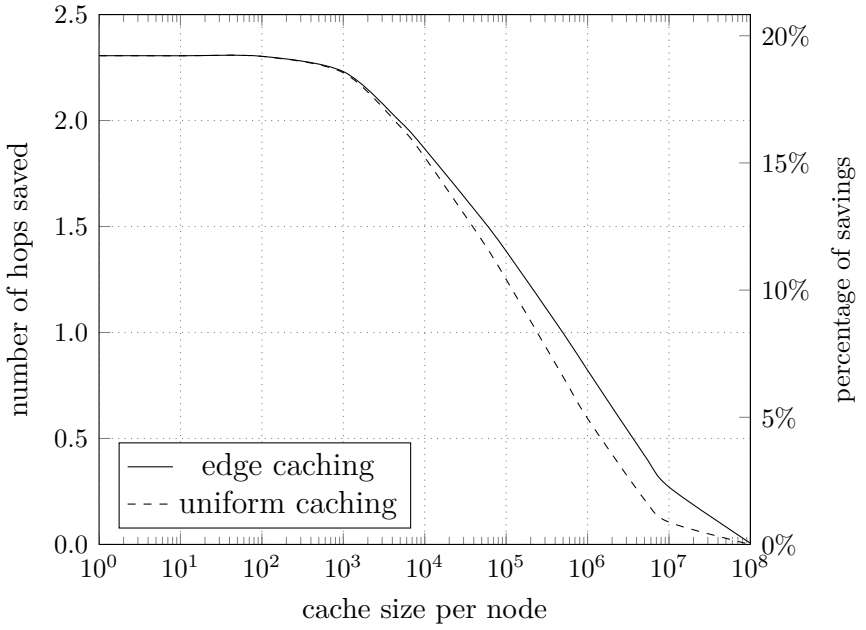

Figure 4.12: The effect of cache size on reducing hop-count through Interest aggregation

and back. Also, Figure 4.13 conveys that the maximum percentage of reduced hops does not far exceed 10% for a Zipf parameter of 0.8. These findings together confirm that aggregating Interests can only bring about marginal improvements in terms of reducing traffic load throughout the network. These improvements come at the cost of maintaining and updating large PITs. Next subsection discusses the dynamics of the PIT size.

4.5.3 PIT Size Distribution

In Section 4.3, I showed that the PIT size follows a Poisson binomial distribution. It is well-known [100] that a Poisson binomial distribution can be approximated by a Poisson distribution with rate equal to the sum of the success probabilities of individual Bernoulli trials. Figure 4.14 compares the two distributions in Q–Q plots where horizontal axes are the theoretical Poisson distribution

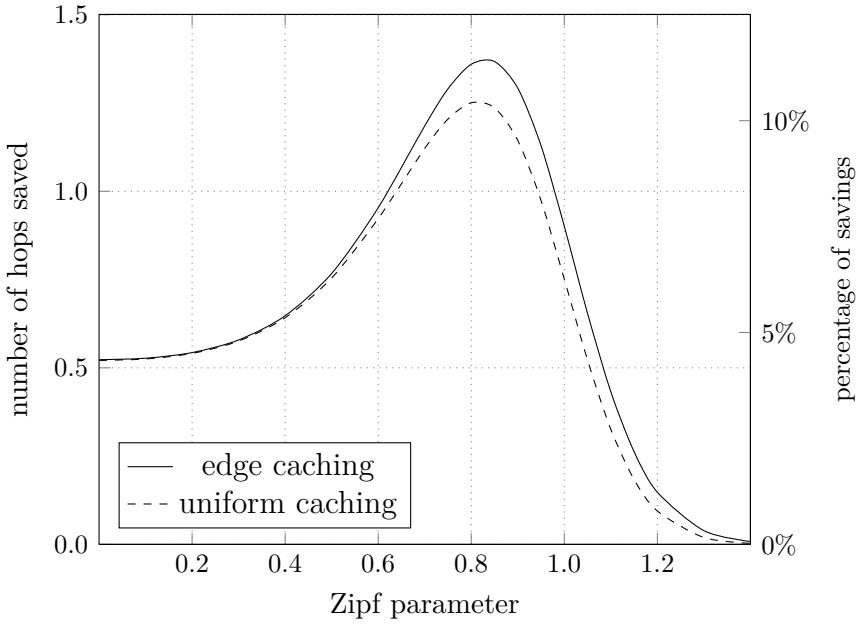

Figure 4.13: The effect of CO popularity distribution on reducing hop-count through Interest aggregation

with rate $\lambda_\ell = \sum_{j=1}^{N} a_{\ell,j}$ and vertical axes are the samples of the PIT size taken from ndnSIM simulations of an NDN tree of degree 10 and height 5. Due to the high computational expense of event-driven simulations, for this experiment I considered a total of 100 COs whose popularity follow a Zipf distribution with parameter $\alpha = 0.8$. The linear relationships between the two distributions in graphs of Figure 4.14 confirm the agreement between the model and simulations.

Next we study how the PIT size responds to varying simulation parameters. For conciseness, I shall only discuss uniform caching strategy as I found the results for the edge-caching scenario (not graphed) very similar. Figure 4.15 shows the average PIT size growth with respect to an increasing system load. Besides the average PIT size, the standard deviations are also shown around each marker point as error bars, though they are so small compared to the average PIT size that are hardly visible in most cases. Clearly, the heavier the system load, the larger

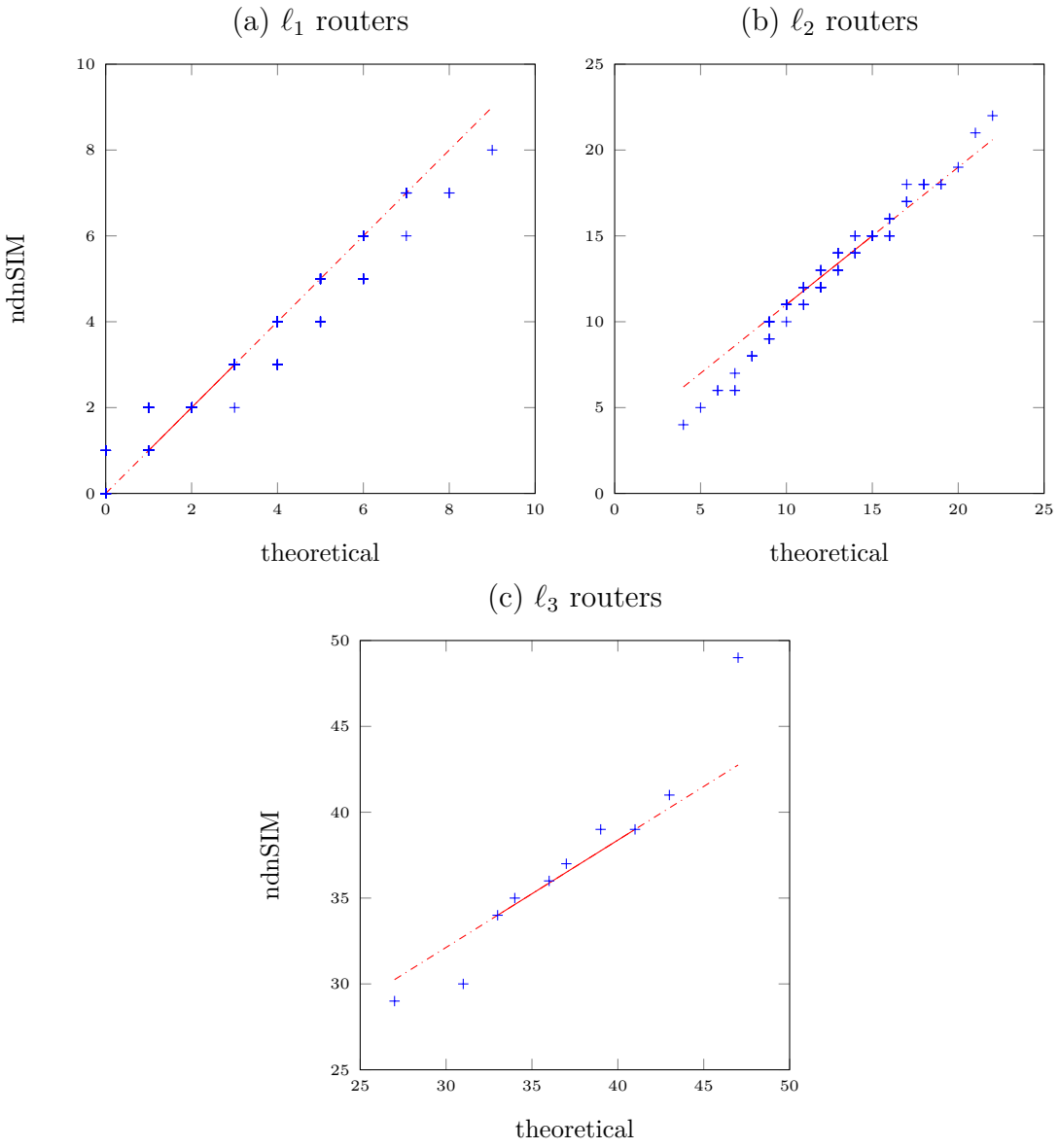

Figure 4.14: Q–Q plots comparing the quantiles of theoretical Poisson distribution (horizontal axes) versus the quantiles of the PIT size samples taken from ndnSIM simulations (vertical axes). The linear relationship indicates good agreement between the two distributions.

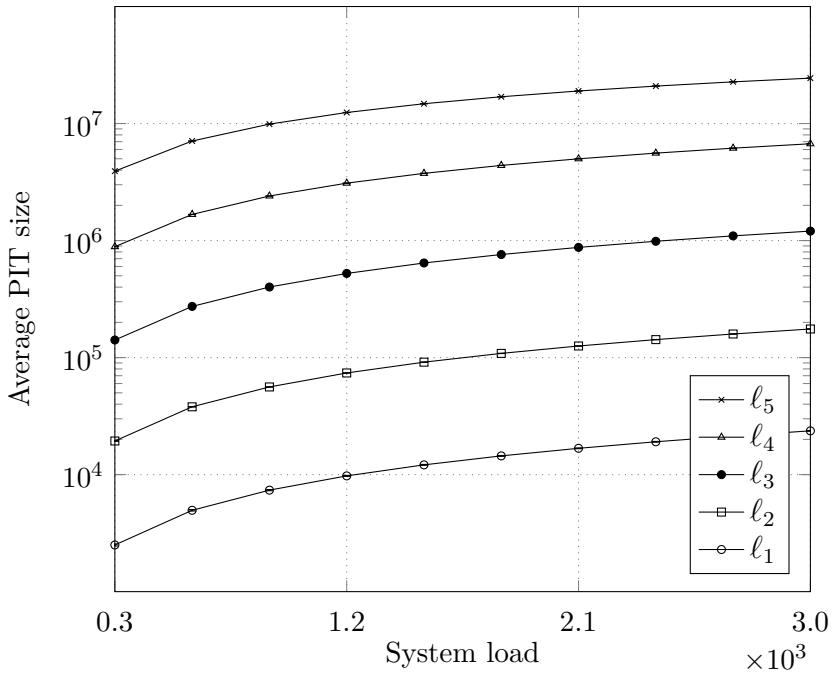

Figure 4.15: Average PIT size vs. increasing system load

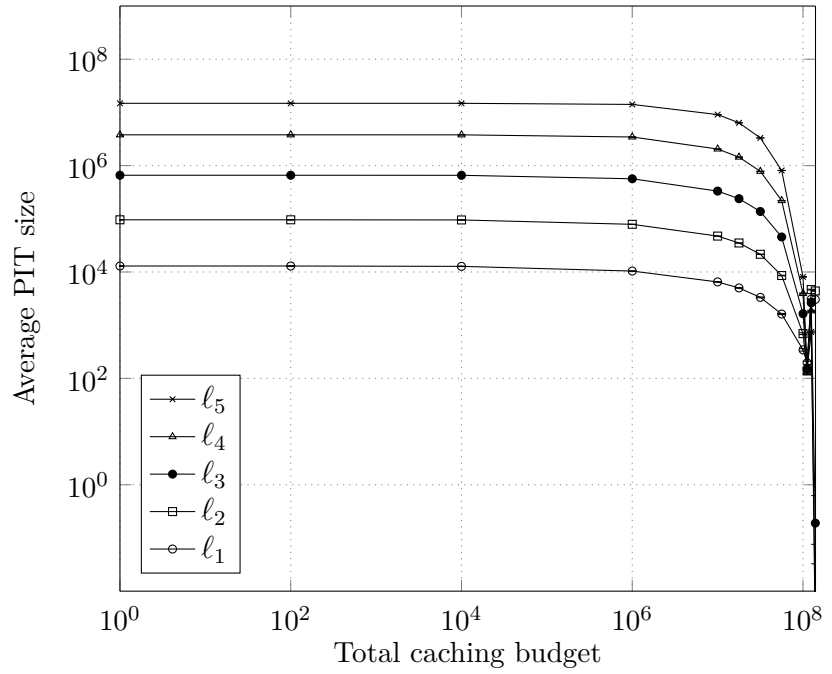

Figure 4.16: Average PIT size vs. increasing cache size

the PIT size. As the results in Figure 4.15 (also Figures 4.16 and 4.17) show, an order of magnitude increase in the PIT size can be observed every deeper level in the network hierarchy. This can pose a critical performance challenge on the core routers which have to constantly update PITs with tens to hundreds of millions of entries. This observation is more insightful when combined with the results of Figure 4.11 demonstrating the expected number of saved hops for an increasing system load. As can be seen, the stateful forwarding plane of NDN makes routers to maintain colossal PITs which is hardly justified by the negligible expected reduction of 5% to 15% we observed on the hop-count.

Figure 4.16 shows an interesting result regarding the effect of an increasing caching budget on the PIT size. In Figure 4.12, we saw that the expected number of saved hops diminish with an increasing cache size. Intuitively, increasing the caching budget improves the cache hit probability and thereby must result in smaller PITs as well. Figure 4.16, however, reveals that the PIT size is not as sensitive to increasing the cache size unless the amount of caching budget available to each router is comparable with the size of the object catalog. Particularly, we see that while savings on the hop-count quickly vanish by stretching the caching capacity past 1000 COs, the PIT does not begin to shrink in size unless the caching storage exceeds a threshold of 10 million COs. This budget is significantly larger than what is usually available to caches in real life. Hence, with a moderate amount of caching storage in an actual NDN deployment, PITs should be expected to operate at near their maximum theoretical size while the expected savings from aggregating Interests remain minimal.

Figure 4.17 demonstrates the impact of the CO popularity (determined by the parameter of the Zipf distribution) on the PIT size. As seen, for a Zipf parameter between 0 (*i.e.*, invariant content popularity) and 0.8, the PIT size is

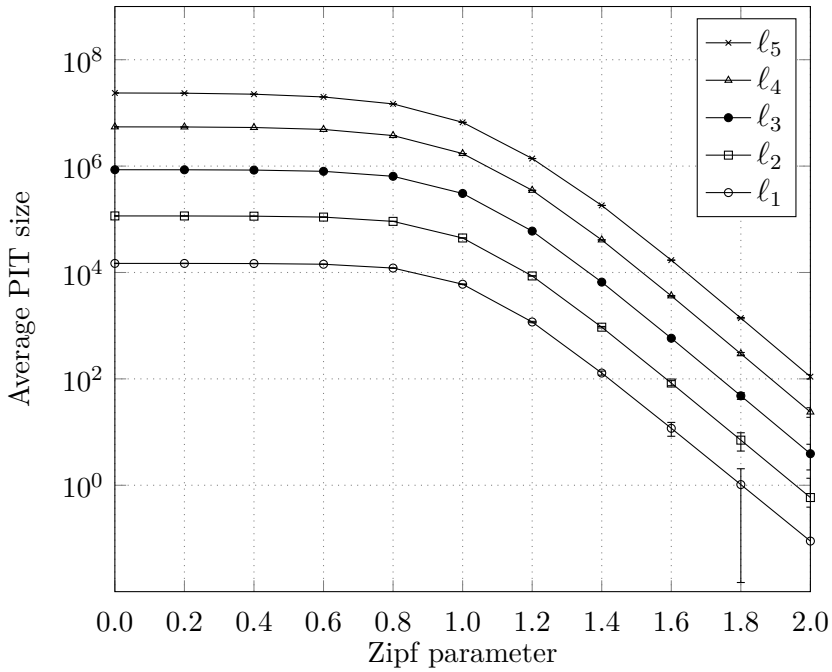

Figure 4.17: Average PIT size vs. CO popularity parameter

maximum. With a Zipf exponent greater than 0.8, the PIT size begins to decay at a steady rate. Empirical studies [97] of the CO popularity distributions on real content networks report a Zipf exponent between 0.8 and 1.0. Although for this range the expected savings on the number of hops reduced through aggregation is maximum (see Figure 4.13), the corresponding PIT size is also close to the maximum according to the results in Figure 4.17.

4.6 Final Remarks

In this chapter, I presented an analytical treatment of Interest aggregation in Content-Centric Networks using a simple yet accurate model where content download delays into the routers are non-zero. Based on my model, I introduced an algorithmic approach based on iterative refinements to analyze a hierarchical

network of content routers in terms of CS hit- and PIT aggregation probabilities and router response times. This method enables the evaluation of large-scale hierarchical caching structures, such as that of an ICN at Internet scale, with high accuracy and low computational cost for which discrete-event simulations are entirely impractical due to high processing and time demands.

My numerical evaluations of a network with over 100,000 NDN routers and hundreds of millions of COs revealed that:

1. With reasonable caching storage available to routers, only 20% of Interests are subject to aggregation.

2. PITs grow at an almost exponential rate every level deeper inside the network.

3. Increasing the caching storage rapidly diminishes the percentage of aggregated Interests, though is hardly effective on reducing the PIT size.

4. Most Interests are aggregated closer to the producers, negating the expected benefits of reducing latency and bandwidth utilization desired from aggregation.

5. With regular traffic and reasonable in-network caching storage, Interest aggregation can only filter between 5% to 15% of the traffic load.

Together, these observations imply that Interest aggregation should only be considered as an optional optimization mechanism in Content-Centric Networking. Furthermore, if per-Interest forwarding state is not needed for other purposes, the stateful forwarding plane of NDN (realized through PITs) can effectively be replaced with more efficient mechanisms, such as CCN-DART [75, 82] and CCN-GRAM [83], which eliminate the need of PIT for Interest forwarding and multicasting while providing similar end-to-end content delivery latencies.

My presented model relies on the assumption that input streams conform to the independent reference model, which need not be true in reality. Nonetheless, the simulation results in [75, 83] indicate that in-network caching makes Interest aggregation unnecessary even with spatio-temporal locality of Interests.

Chapter 5

Conclusions

With the proliferation of mobile phones and other handheld devices and in conjunction with the advent of social media, the modern era has witnessed an ever-increasing demand for content production and sharing among the users. The underlying networking infrastructure to furnish these services, nonetheless, does not expand rapidly enough to accommodate such tremendous traffic demand. Consequently, it is imperative to understand how communication networks respond to a growing number of end users and a steadily increasing amount of traffic, and to devise pragmatic solutions that let the existing infrastructures evolve into more scalable systems.

To gain a better understanding of this problem, in this thesis we decomposed networking systems into several conceptual subsystems: (1) a *social layer* consisting of the end users and the interactions among them, (2) an *information layer* comprising the content files and data objects they share, and (3) a *communication layer* including the physical devices and the underlying backbone and protocols that actually enable the exchange of information. Focusing on the specific characteristics of each component separately, this decomposition remarkably reduces the complexities of the given problem. Moreover, it is easier to understand the

dynamics of the actual system by investigating the reciprocal impact that each layer has on the performance of the others.

In connection with the described approach, the second chapter of this thesis began by analyzing how the spatial diversity of *social* interactions can affect the performance of a *communication* network. This analysis was based on the observation that people naturally tend to communicate with parties that are geographically closer to them more often than with ones at farther distances [13]. In order to quantify how intense this tendency is, a *clustering parameter* α was introduced that determined how localized the inter-personal communications look like on the social layer. The more concentrated the social interactions, the greater the value of α and vice versa. The clustering parameter has a critical impact on the scalability of communication networks, because transferring information over longer distances is generally more costly in terms of the networking resources required. A larger clustering parameter means that the social contacts are geographically more concentrated around each node, and thus, the average amount of resources the network utilizes for the exchange of information is smaller.

A natural question there was for what ranges of α a communication network exhibits scalability; that is, the network becomes able to handle a large number of users—without significant performance loss—given only finite amount of resources. A careful treatment of this problem was conducted and it was found that there actually exists a threshold on the clustering parameter α beyond which the communication networks become generally scalable. This critical value is indeed $\alpha = 3$, which we termed *the scalability threshold*. At this threshold and beyond, nodes are still free to establish both local and long-distance social connections; however, the combination of the nodes' social connections on the average look clustered enough that the whole system scales properly.

That the communication networks scale under spatially clustered social interactions was an interesting finding; yet we identified a problem with the applicability of this result. Social relationships are inherent to the networks; that is, users cannot be dictated as to how to interact with one another. Extensive studies (*e.g.*, [14, 16]) of real-world social networks show that humans interactions have a clustering parameter of less than 2 which is clearly smaller than the scalability threshold of 3 we found. This gap, nonetheless, does not indicate that communication networks are inherently unscalable, thanks to optimization mechanisms such as in-network content caching. While social contacts cannot physically be brought closer, the content they share can. In fact, we can take advantage of content replication and caching on the *information layer* of the network to bring the information closer to the consumers and thereby, bridge the scalability gap imposed by the *social layer*. This study was the subject of the third chapter of this thesis.

Information-centric networking (ICN) was studied in the third chapter as a foundation upon which the future Internet can be built. This design is motivated by the evolution of the Internet usage in which most traffic is for accessing vast amount of information irrespective of its physical location. In accordance to this shift, in-network content caching is a fundamental principle of most ICN proposals that is to increase information availability and reduce end-to-end latency by placing useful information near the consumers. Clearly, how the caching storage is distributed throughout the network can pose a prominent impact on the magnitude of the desired benefits. One objective of the third chapter was to find the optimal cache allocation strategy that minimizes end-to-end latency given a fixed storage budget. Solutions to this problem were presented using integer programming and compared against other methods of caching, namely uniform on-path

caching and edge-caching. The former strategy is the ICN's de facto standard that promotes caching all kind of traffic everywhere in the network. The latter, however, is an underappreciated method that advocates caching content only near the consumers at the edge of the network. It was shown in the third chapter that edge-caching's performance is only within a small margin from the optimal solution in terms of reduction of end-to-end delay, while always outperforming on-path caching for a wide range of scenarios studied. Next, the impact of reference locality in space and time was studied using a novel algorithm for generating localized references. It was shown that highly localized references make edge-caching perform even closer to the optimal caching strategy. This insight is important as it enables achieving maximum benefits of in-network content caching by just adding caching functionality to the edge routers.

To enable fast content retrieval, a core component of ICN is the location-agnostic content addressing by name. In contrast to the conventional model of host address resolution using IP, ICN enables name-based routing of content objects and ICN routers build routing tables that route to content names rather than addresses where they are located. In the absence of IP addressing, the routers must keep track of all users' queries (a.k.a. Interests) forwarded as well as the corresponding ingress interfaces so that correct data can be delivered to the requesters on the way back. In the fourth chapter, we discussed how this information—commonly referred to as *the state*—is populated and stored in Pending Interest Tables (PITs). A thorough characterization of the stateful forwarding plane of ICN was next presented that quantified the key performance metrics such as cache hit rate, Interest aggregation probability, cache response time and the PIT size distribution. Using this model, we observed that in general less than 20% of Interests are subject to aggregation, out of which the majority happen near the

producers. This much aggregation was shown that on the average saves between 5% to 15% of the total in-network transmissions at the expense of maintaining PITs that have to constantly handle hundreds of millions of entries. We concluded that the stateful forwarding plane of ICN can hardly achieve its expected objectives of reducing end-to-end latency and improving bandwidth utilization, leading to the necessary conclusion that the forwarding plane of ICN must be rethought. In sum, our results confirmed that content caching by itself realizes most benefits sought after through Interest aggregation. Therefore, if storing the forwarding state per Interest is not needed for other reasons, a stateless forwarding plane is indeed a more reasonable approach to implement ICN's at Internet scale.

In this direction, a useful addendum to the results discussed in this thesis would be an analysis of the content discovery and routing in information-centric networks which is achieved through ICN's Forwarding Information Base (FIB). Unlike conventional FIB's in today's Internet that enable route to IP addresses, the FIB's in ICN provide route to names. Since the number of content objects in networks is usually orders of magnitude larger than the number of hosts, it is expected that the FIB size in ICN's must be considerably larger than today's existing FIB's. The exact distribution of the FIB size and its scaling properties, however, are not yet well understood. A mathematical framework for modeling the performance of the FIB will enhance our understanding of the performance of ICN's from yet another perspective. We shall leave this as an avenue for future research.

Bibliography

[1] P. Gupta and P. Kumar. The Capacity of Wireless Networks. *IEEE Trans. Inf. Theory*, 46(2):388–404, March 2000.

[2] Seyed Fayazbakhsh, Yin Lin, Amin Tootoonchian, Ali Ghodsi, Teemu Koponen, Bruce Maggs, K. Ng, Vyas Sekar, and Scott Shenker. Less Pain, Most of the Gain: Incrementally Deployable ICN. *ACM SIGCOMM Computer Communication Review*, 43(4):147–158, October 2013.

[3] Ali Dabirmoghaddam and J. J. Garcia-Luna-Aceves. Opportunistic Walks on Random Geometric Networks and Their Application in Scalability Analysis. In *Proc. IEEE SECON*, pages 559–567, New Orleans, LA, USA, June 2013.

[4] Ali Dabirmoghaddam and J. J. Garcia-Luna-Aceves. Scaling Properties of Random Networks Under Proximity-based Social Relations. In *Proc. IEEE NetSciCom*, pages 1–6, May 2014.

[5] Ali Dabirmoghaddam and J. J. Garcia-Luna-Aceves. Proximity-driven Social Interactions and Their Impact on the Throughput Scaling of Wireless Networks. In *Proc. IEEE IPCCC*, pages 1–10, December 2014.

[6] M. Grossglauser and D. Tse. Mobility Increases the Capacity of Ad Hoc Wireless Networks. *IEEE/ACM Trans. Netw.*, 10(4):477–486, August 2002.

[7] S. Kulkarni and P. Viswanath. A Deterministic Approach to Throughput Scaling in Wireless Networks. *IEEE Trans. Inf. Theory*, 50(6):1041–1049, June 2004.

[8] J. J. Garcia-Luna-Aceves, H. Sadjadpour, and Z. Wang. Challenges: Towards Truly Scalable Ad Hoc Networks. In *Proc. ACM MobiCom*, pages 207–214, Montreal, Canada, September 2007.

[9] Z. Wang, H. Sadjadpour, and J. J. Garcia-Luna-Aceves. A Unifying Perspective on the Capacity of Wireless Ad Hoc Networks. In *Proc. IEEE INFOCOM*, pages 753–761, Phoenix, AZ, USA, April 2008.

[10] Duncan Watts and Steven Strogatz. Collective Dynamics of 'Small-World' Networks. *Nature*, 393(6684):440–442, June 1998.

[11] J. Kleinberg. The Small-World Phenomenon: An Algorithm Perspective. In *Proc. ACM STOC*, pages 163–170, Portland, OR, USA, May 2000.

[12] M. Dietzfelbinger and P. Woelfel. Tight Lower Bounds for Greedy Routing in Uniform Small World Rings. In *Proc. ACM STOC*, pages 591–600, Maryland, DC, USA, May 2009.

[13] B. Latané, J. Liu, A. Nowak, M. Bonevento, and L. Zheng. Distance Matters: Physical Space and Social Impact. *Personality and Social Psychology Bulletin*, 21(8):795–805, August 1995.

[14] Renaud Lambiotte, Vincent D Blondel, Cristobald de Kerchove, Etienne Huens, Christophe Prieur, Zbigniew Smoreda, and Paul Van Dooren. Geographical Dispersal of Mobile Communication Networks. *Physica A: Statistical Mechanics and its Applications*, 387(21):5317–5325, September 2008.

[15] Jacob Goldenberg and Moshe Levy. Distance is not Dead: Social Interaction and Geographical Distance in the Internet Era. arXiv:0906.3202v2 [cs.CY], October 2009.

[16] Johannes Illenberger, Kai Nagel, and Gunnar Flötteröd. The Role of Spatial Interaction in Social Networks. *Networks and Spatial Economics*, 13(3):255–282, September 2013.

[17] Abbas El Gamal, James Mammen, Balaji Prabhakar, and Devavrat Shah. Optimal Throughput-Delay Scaling in Wireless Networks—Part I: The Fluid Model. *IEEE Transactions on Information Theory*, 52(6):2568–2592, June 2006.

[18] Eugene Perevalov, Rick Blum, and Danny Safi. Capacity of Clustered Ad Hoc Networks: How Large Is "Large"? *IEEE Transactions on Communications*, 54(9):1672–1681, September 2006.

[19] Benyuan Liu, Don Towsley, and Ananthram Swami. Data Gathering Capacity of Large Scale Multihop Wireless Networks. In *Proc. IEEE MASS*, pages 124–132, Atlanta, GA, USA, September 2008.

[20] Brad Karp and Hsiang Kung. GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. In *Proc. ACM MobiCom*, pages 243–254, Boston, MA, USA, August 2000.

[21] Fabian Kuhn, Roger Wattenhofer, Yan Zhang, and Aaron Zollinger. Geometric Ad-Hoc Routing: Of Theory and Practice. In *Proc. ACM PODC*, pages 63–72, Boston, MA, USA, July 2003.

[22] Xiaoyan Hong, Kaixin Xu, and Mario Gerla. Scalable Routing Protocols for Mobile Ad Hoc Networks. *IEEE Network*, 16(4):11–21, July 2002.

[23] Y. Wang, , and J. J. Garcia-Luna-Aceves. Collision Avoidance in Multi-hop Ad Hoc Networks. In *Proc. ACM/IEEE MASCOTS*, pages 145–154, Fort Worth, TX, USA, October 2002.

[24] Piyush Gupta and Panganamala Kumar. Critical Power for Asymptotic Connectivity. In *Proc. IEEE CDC*, pages 1106–1110, Tampa, FL, USA, December 1998.

[25] Paolo Santi and Douglas Blough. The Critical Transmitting Range for Connectivity in Sparse Wireless Ad hoc Networks. *IEEE Trans. Mobile Comput.*, 2(1):25–39, January 2003.

[26] Massimo Franceschetti, Olivier Dousse, N David, and Patrick Thiran. Closing the Gap in the Capacity of Wireless Networks Via Percolation Theory. *IEEE Transactions on Information Theory*, 53(3):1009–1018, March 2007.

[27] Benyuan Liu, Zhen Liu, and Don Towsley. On the Capacity of Hybrid Wireless Networks. In *Proc. IEEE INFOCOM*, pages 1543–1552, San Francisco, CA, USA, March 2003.

[28] Ulaş Kozat and Leandros Tassiulas. Throughput Capacity of Random Ad Hoc Networks with Infrastructure Support. In *Proc. ACM MobiCom*, pages 55–65, San Diego, CA, USA, September 2003.

[29] Su Yi, Yong Pei, and Shivkumar Kalyanaraman. On the Capacity Improvement of Ad Hoc Wireless Networks Using Directional Antennas. In *Proc. ACM MobiHoc*, pages 108–116, Annapolis, MD, USA, June 2003.

[30] Hong Dai, Kam Ng, Raymond Wong, and Min Wu. On the Capacity of Multi-Channel Wireless Networks Using Directional Antennas. In *Proc. IEEE INFOCOM*, pages 628–636, Phoenix, AZ, USA, April 2008.

[31] Anders Høst-Madsen. Capacity Bounds for Cooperative Diversity. *IEEE Transactions on Information Theory*, 52(4):1522–1544, April 2006.

[32] Ayfer Ozgur, Olivier Lévêque, and David Tse. Hierarchical Cooperation Achieves Optimal Capacity Scaling in Ad Hoc Networks. *IEEE Transactions on Information Theory*, 53(10):3549–3572, October 2007.

[33] Ali Dabirmoghaddam, Maziar Mirzazad-Barijough, and J. J. Garcia-Luna-Aceves. Understanding Optimal Caching and Opportunistic Caching at ẗhe Edgeöf Information-Centric Networks. In *Proc. ACM ICN*, pages 47–56, September 2014.

[34] Jim Kurose. Information-Centric Networking: The Evolution from Circuits to Packets to Content. *Computer Networks*, 66(0):112–120, June 2014.

[35] Ali Ghodsi, Scott Shenker, Teemu Koponen, Ankit Singla, Barath Raghavan, and James Wilcox. Information-Centric Networking: Seeing the Forest for the Trees. In *Proc. ACM HotNets*, pages 1–6, Cambridge, MA, USA, November 2011.

[36] Bengt Ahlgren, Christian Dannewitz, Claudio Imbrenda, Dirk Kutscher, and Börje Ohlman. A Survey of Information-Centric Networking. *IEEE Communications Magazine*, 50(7):26–36, July 2012.

[37] Stefano Traverso, Mohamed Ahmed, Michele Garetto, Paolo Giaccone, Emilio Leonardi, and Saverio Niccolini. Temporal Locality in Today's Content Caching: Why It Matters and How to Model It. *ACM SIGCOMM Computer Communication Review*, 43(5):5–12, October 2013.

[38] Elisha Rosensweig, Jim Kurose, and Don Towsley. Approximate Models for

General Cache Networks. In *Proc. IEEE INFOCOM*, pages 1–9, San Diego, CA, USA, March 2010.

[39] Giovanna Carofiglio, Massimo Gallo, and Luca Muscariello. Bandwidth and Storage Sharing Performance in Information Centric Networking. In *Proc. ACM SIGCOMM Workshop on ICN*, pages 26–31, Toronto, Canada, August 2011.

[40] Christine Fricker, Philippe Robert, and James Roberts. A Versatile and Accurate Approximation for LRU Cache Performance. In *Proc. ITC*, pages 1–8, Krakow, Poland, September 2012.

[41] James Roberts and Nada Sbihi. Exploring the Memory-Bandwidth Tradeoff in an Information-Centric Network. In *Proc. ITC*, pages 1–9, Shanghai, China, September 2013.

[42] Daryl Daley and David Vere-Jones. *An Introduction to the Theory of Point Processes*, volume I: Elementary Theory and Methods. Springer, New York, NY, USA, second edition, 2003.

[43] Alan Hawkes. Spectra of Some Self-Exciting and Mutually Exciting Point Processes. *Biometrika*, 58(1):83–90, April 1971.

[44] Lee Breslau, Pei Cao, Li Fan, Graham Phillips, and Scott Shenker. Web Caching and Zipf-like Distributions: Evidence and Implications. In *Proc. IEEE INFOCOM*, pages 126–134, New York, NY, USA, March 1999.

[45] Phillipa Gill, Martin Arlitt, Zongpeng Li, and Anirban Mahanti. Youtube Traffic Characterization: A View from the Edge. In *Proc. ACM IMC*, pages 15–28, San Diego, CA, USA, October 2007.

[46] Alexander Afanasyev, Ilya Moiseenko, and Lixia Zhang. ndnSIM: NDN Simulator for NS-3. Technical Report NDN-0005, NDN, October 2012.

[47] Asit Dan and Don Towsley. An Approximate Analysis of the LRU and FIFO Buffer Replacement Schemes. *ACM SIGMETRICS Performance Evaluation Review*, 18(1):143–152, April 1990.

[48] Predrag Jelenković and Ana Radovanović. Least-Recently-Used Caching with Dependent Requests. *Theoretical Computer Science*, 326(1):293–327, October 2004.

[49] Hao Che, Ye Tung, and Zhijun Wang. Hierarchical Web Caching Systems: Modeling, Design and Experimental Results. *IEEE Journal on Selected Areas in Communications*, 20(7):1305–1314, September 2002.

[50] Pablo Rodriguez, Christian Spanner, and Ernst Biersack. Analysis of Web Caching Architectures: Hierarchical and Distributed Caching. *IEEE/ACM Transactions on Networking*, 9(4):404–418, August 2001.

[51] Valentina Martina, Michele Garetto, and Emilio Leonardi. A Unified Approach to the Performance Analysis of Caching Systems. In *Proc. IEEE INFOCOM*, Toronto, Canada, April 2014.

[52] Ioannis Psaras, Richard Clegg, Raul Landa, Wei Chai, and George Pavlou. Modelling and Evaluation of CCN-Caching Trees. In *IFIP Networking*, pages 78–91. Springer, May 2011.

[53] Ioannis Psaras, Wei Chai, and George Pavlou. Probabilistic In-Network Caching for Information-Centric Networks. In *Proc. ACM SIGCOMM Workshop on ICN*, pages 55–60, Helsinki, Finland, August 2012.

[54] Massimo Gallo, Bruno Kauffmann, Luca Muscariello, Alain Simonian, and Christian Tanguy. Performance Evaluation of the Random Replacement Policy for Networks of Caches. *Performance Evaluation*, 72(0):16–36, February 2014.

[55] N Melazzi, G Bianchi, A Caponi, and A Detti. A General, Tractable and Accurate Model for a Cascade of LRU Caches. *IEEE Communications Letters*, 18(5):877–880, May 2014.

[56] Wei Chai, Diliang He, Ioannis Psaras, and George Pavlou. Cache "Less for More" in Information-Centric Networks (Extended Version). *Computer Communications*, 36(7):758–770, April 2013.

[57] Yonggong Wang, Zhenyu Li, G. Tyson, S. Uhlig, and Gaogang Xie. Optimal Cache Allocation for Content-Centric Networking. In *Proc. IEEE ICNP*, pages 1–10, Göttingen, Germany, October 2013.

[58] Elisha Rosensweig, Daniel Menasche, and Jim Kurose. On the Steady-State of Cache Networks. In *Proc. IEEE INFOCOM*, pages 863–871, Turin, Italy, April 2013.

[59] Teemu Koponen, Mohit Chawla, Byung Chun, Andrey Ermolinskiy, Kye Kim, Scott Shenker, and Ion Stoica. A Data-Oriented (and Beyond) Network Architecture. *ACM SIGCOMM Computer Communication Review*, 37(4):181–192, August 2007.

[60] Van Jacobson, Diana Smetters, James Thornton, Michael Plass, Nicholas Briggs, and Rebecca Braynard. Networking Named Content. In *Proc. ACM CoNEXT*, pages 1–12, Rome, Italy, December 2009.

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

[62] Guilherme Domingues, Edmundo Silva, Rosa Leão, and Daniel Menasché. Enabling Information Centric Networks through Opportunistic Search, Routing and Caching. arXiv:1310.8258 [cs.NI], October 2013.

[63] Gareth Tyson, Sebastian Kaune, Simon Miles, Yehia El-khatib, Andreas Mauthe, and Adel Taweel. A Trace-Driven Analysis of Caching in Content-Centric Networks. In *Proc. ICCCN*, pages 1–7, Munich, Germany, July 2012.

[64] Peter Danzig, Richard Hall, and Michael Schwartz. A Case for Caching File Objects Inside Internetworks. *ACM SIGCOMM Computer Communication Review*, 23(4):239–248, October 1993.

[65] Dario Rossi and Giuseppe Rossini. Caching Performance of Content Centric Networks Under Multi-Path Routing (and More). Technical report, Telecom ParisTech, 2011.

[66] C. Fricker, P. Robert, J. Roberts, and N. Sbihi. Impact of Traffic Mix on Caching Performance in a Content-Centric Network. In *Proc. IEEE INFOCOM Workshops*, pages 310–315, Orlando, FL, USA, March 2012.

[67] R. Fonseca, V. Almeida, M. Crovella, and B. Abrahao. On the Intrinsic Locality Properties of Web Reference Streams. In *Proc. IEEE INFOCOM*, pages 448–458, San Francisco, CA, USA, March 2003.

[68] Ludmila Cherkasova and Minaxi Gupta. Analysis of Enterprise Media Server Workloads: Access Patterns, Locality, Content Evolution, and Rates of Change. *IEEE/ACM Transactions on Networking*, 12(5):781–794, October 2004.

[69] Giovanna Carofiglio, Massimo Gallo, Luca Muscariello, and Diego Perino. Modeling Data Transfer in Content-Centric Networking. In *Proc. ITC*, pages 111–118, San Francisco, CA, USA, September 2011.

[70] Lixia Zhang, Alexander Afanasyev, Jeffrey Burke, Van Jacobson, kc claffy, Patrick Crowley, Christos Papadopoulos, Lan Wang, and Beichuan Zhang. Named Data Networking. *ACM SIGCOMM Computer Communication Review*, 44(3):66–73, 2014.

[71] Ali Dabirmoghaddam, Mostafa Dehghan, and J. J. Garcia-Luna-Aceves. Characterizing Interest Aggregation in Content-Centric Networks. In *Proc. IFIP Networking*, pages 449–457, May 2016.

[72] `http://www.ccnx.org/releases/latest/doc/technical/`.

[73] `http://www.squid-cache.org`.

[74] Cheng Yi, Alexander Afanasyev, Ilya Moiseenko, Lan Wang, Beichuan Zhang, and Lixia Zhang. A Case For Stateful Forwarding Plane. *Computer Communications*, 36(7):779–791, 2013.

[75] J. Garcia-Luna-Aceves and Maziar Mirzazad-Barijough. A Light-Weight Forwarding Plane for Content Centric Networks. In *Proc. IEEE ICNC*, Kauai, HI, USA, 2016.

[76] Huichen Dai, Bin Liu, Yan Chen, and Yi Wang. On Pending Interest Table

in Named Data Networking. In *Proc. ACM/IEEE ANCS*, pages 211–222, Austin, TX, USA, 2012.

[77] Yi Wang, Keqiang He, Huichen Dai, Wei Meng, Junchen Jiang, Bin Liu, and Yan Chen. Scalable Name Lookup in NDN Using Effective Name Component Encoding. In *Proc. IEEE ICDCS*, pages 688–697, Macau, China, 2012.

[78] Matteo Varvello, Diego Perino, and Leonardo Linguaglossa. On the Design and Implementation of a Wire-Speed Pending Interest Table. In *Proc. IEEE INFOCOM Workshops*, pages 369–374, Turin, Italy, 2013.

[79] Haowei Yuan and Patrick Crowley. Scalable Pending Interest Table Design: From Principles to Practice. In *Proc. IEEE INFOCOM*, pages 2049–2057, Toronto, Canada, 2014.

[80] Matteo Virgilio, Guido Marchetto, and Riccardo Sisto. PIT Overload Analysis in Content Centric Networks. In *Proc. ACM SIGCOMM Workshop on ICN*, pages 67–72, Hong Kong, China, 2013.

[81] Amuda Abu, Brahim Bensaou, and Jason Wang. Interest Packets Retransmission in Lossy CCN Networks and Its Impact on Network Performance. In *Proc. ACM ICN*, pages 167–176, Paris, France, 2014.

[82] J Garcia-Luna-Aceves. A More Scalable Approach to Content Centric Networking. In *Proc. IEEE ICCCN*, pages 1–8, Las Vegas, NV, USA, 2015.

[83] J. Garcia-Luna-Aceves and Maziar Mirzazad-Barijough. Content-Centric Networking Using Anonymous Datagrams. In *Proc. IFIP Networking*, Vienna, Austria, 2016.

[84] Diego Perino and Matteo Varvello. A Reality Check for Content Centric Networking. In *Proc. ACM SIGCOMM Workshop on ICN*, pages 44–49, Toronto, Canada, August 2011.

[85] Wei You, Bertrand Mathieu, Patrick Truong, Jean-François Peltier, and Gwendal Simon. DiPIT: A Distributed Bloom-Filter based PIT Table for CCN Nodes. In *Proc. ICCCN*, pages 1–7, Munich, Germany, July 2012.

[86] Christos Tsilopoulos, George Xylomenos, and Yannis Thomas. Reducing Forwarding State in Content-Centric Networks with Semi-Stateless Forwarding. In *Proc. IEEE INFOCOM*, pages 2067–2075, Toronto, Canada, April 2014.

[87] Mostafa Dehghan, Bo Jiang, Ali Dabirmoghaddam, and Don Towsley. On the Analysis of Caches with Pending Interest Tables. In *Proc. ACM ICN*, pages 47–56, San Francisco, CA, September 2015.

[88] Ronald W Wolff. Poisson Arrivals See Time Averages. *Operations Research*, 30(2):223–231, 1982.

[89] Jian Ni and Danny Tsang. Large-Scale Cooperative Caching and Application-Level Multicast in Multimedia Content Delivery Networks. *IEEE Communications Magazine*, 43(5):98–105, 2005.

[90] Raphael Chand and Pascal Felber. XNET: A Reliable Content-Based Publish/Subscribe System. In *Proc. IEEE SRDS*, pages 264–273, Florianpolis, Brazil, 2004.

[91] Jin Cao, William Cleveland, Dong Lin, and Don Sun. Internet Traffic Tends Toward Poisson and Independent as the Load Increases. In *Nonlinear Esti-*

mation and Classification, volume 171 of *Lecture Notes in Statistics*, pages 83–109. Springer New York, 2003.

[92] E Cinlar and R Agnew. On the Superposition of Point Processes. *Journal of the Royal Statistical Society. Series B (Methodological)*, 30(3):576–581, 1968.

[93] Andrew Conn, Nicholas Gould, and P Toint. *Trust Region Methods*. MPS-SIAM Series on Optimization. SIAM, Philadelphia, PA, USA, 2000.

[94] Serge Gratton, Annick Sartenaer, and Philippe Toint. Recursive Trust-Region Methods for Multiscale Nonlinear Optimization. *SIAM Journal on Optimization*, 19(1):414–444, 2008.

[95] Internet Topology at Router- and AS-levels. Accessed: 2015-07-17.

[96] Artem Russakovskii. How to Find Out the Number of Videos on Youtube, 2008. Accessed: 2015-07-22.

[97] A. Mahanti, C. Williamson, and D. Eager. Traffic Analysis of a Web Proxy Caching Hierarchy. *IEEE Network*, 14(3):16–23, May 2000.

[98] James Cowie, David Nicol, and Andy Ogielski. Modeling the Global Internet. *Computing in Science & Engineering*, 1(1):42–50, 1999.

[99] Charles Fraleigh. *Provisioning Internet Backbone Networks to Support Latency Sensitive Applications*. PhD thesis, Stanford University, Stanford, CA, USA, May 2002.

[100] J. Hodges and Lucien Cam. The Poisson Approximation to the Poisson Binomial Distribution. *The Annals of Mathematical Statistics*, 31(3):737–740, September 1960.