NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS

SCHOOL OF SCIENCE

DEPARTMENT OF INFORMATICS AND TELECOMMUNICATIONS

PROGRAM OF POSTGRADUATE STUDIES

PhD DISSERTATION

Cooperative mechanisms for information dissemination and retrieval in networks with autonomous nodes

Eva H. Jaho

ATHENS

DECEMBER 2011

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

Συνεργατικοί μηχανισμοί για διάδοση και ανάκτηση πληροφορίας σε δίκτυα με αυτόνομους κόμβους

Εύα Χ. Γιάχο

ΑΘΗΝΑ

ΔΕΚΕΜΒΡΙΟΣ 2011

PhD DISSERTATION

Cooperative mechanisms for information dissemination and retrieval in networks with autonomous nodes

Eva Jaho

ADVISOR: **Prof. Ioannis Stavrakakis**, Professor NKUA

THREE-MEMBER ADVISING COMMITTEE:

Ioannis Stavrakakis, Professor NKUA

Vassilios Zissimopoulos, Professor NKUA

Stathes Hadjiefthymiades, Assistant Professor NKUA

SEVEN-MEMBER EXAMINATION COMMITTEE

Ioannis Stavrakakis	**Vassilios Zissimopoulos**	**Stathes Hadjiefthymiades**
Professor NKUA	Professor NKUA	Asst. Professor NKUA
Symeon Papavassiliou	**Athanasia Alonistioti**	**Christos Douligeris**
Assoc. Professor NTUA	Lecturer NKUA	Professor University of Piraeus

Konstantinos Oikonomou

Asst. Professor Ionian University

Examination Date: December 9, 2011

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

Συνεργατικοί μηχανισμοί για διάδοση και ανάκτηση πληροφορίας σε δίκτυα με αυτόνομους κόμβους

Εύα Γιάχο

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: **Ιωάννης Σταυρακάκης**, Καθηγητής ΕΚΠΑ

ΤΡΙΜΕΛΗΣ ΕΠΙΤΡΟΠΗ ΠΑΡΑΚΟΛΟΥΘΗΣΗΣ:

Ιωάννης Σταυρακάκης, Καθηγητής ΕΚΠΑ

Βασίλειος Ζησιμόπουλος, Καθηγητής ΕΚΠΑ

Ευστάθιος Χατζηευθυμιάδης, Επίκουρος Καθηγητής ΕΚΠΑ

ΕΠΤΑΜΕΛΗΣ ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ:

Ιωάννης Σταυρακάκης	**Βασίλειος Ζησιμόπουλος**	**Ευστάθιος Χατζηευθυμιάδης**
Καθηγητής ΕΚΠΑ	Καθηγητής ΕΚΠΑ	Επικ. Καθηγητής ΕΚΠΑ
Συμεών Παπαβασιλείου	**Αθανασία Αλωνιστιώτη**	**Χρήστος Δουληγέρης**
Αναπ. Καθηγητής ΕΜΠ	Λέκτορας ΕΚΠΑ	Καθηγητής Παν/μίου Πειραιά

Κωνσταντίνος Οικονόμου

Επίκ. Καθηγητής Παν/μίου Ιονίου

Ημερομηνία Εξέτασης: 9 Δεκεμβρίου 2011

ΠΕΡΙΛΗΨΗ

Αυτή η διατριβή συνεισφέρει στη βιβλιογραφία με το να προτείνει και να μοντελοποιήσει καινοτόμους αλγορίθμους και σχήματα που επιτρέπουν στις διεργασίες διάδοσης και ανάκτησης πληροφοριών – και γενικότερα της διαχείρισης περιεχομένου – να εκτελεστούν πιο αποτελεσματικά σε ένα σύγχρονο περιβάλλον δικτύωσης. Εκτός από τη διάδοση και ανάκτηση των πληροφοριών, άλλες πτυχές της διαχείρισης περιεχομένου που εξετάζουμε είναι η αποθήκευση και η κατηγοριοποίηση. Η πιο σημαντική πρόκληση που αφορά πολλά από τα σχήματα που προτείνονται στην παρούσα εργασία είναι η ανάγκη να διαχειριστούν την αυτονομία των κόμβων, διατηρώντας παράλληλα τον κατανεμημένο, καθώς και τον ανοικτό χαρακτήρα του συστήματος. Κατά το σχεδιασμό κατανεμημένων μηχανισμών σε δίκτυα με αυτόνομους κόμβους, ένα σημαντικό επίσης ζητούμενο είναι να δημιουργηθούν κίνητρα ώστε οι κόμβοι να συνεργάζονται κατά την εκτέλεση των καθηκόντων επικοινωνίας. Ένα καινούργιο χαρακτηριστικό των περισσοτέρων από τα προτεινόμενα σχήματα είναι η αξιοποίηση των κοινωνικών χαρακτηριστικών των κόμβων, εστιάζοντας στο πώς τα κοινά ενδιαφέροντα των κόμβων μπορούν να αξιοποιηθούν για τη βελτίωση της αποδοτικότητας στην επικοινωνία.

Για την αξιολόγηση της απόδοσης των προτεινόμενων αλγορίθμων και σχημάτων, κυρίως αναπτύσσουμε μαθηματικά στοχαστικά μοντέλα και λαμβάνουμε αριθμητικά αποτελέσματα. Όπου είναι απαραίτητο, παρέχουμε αποτελέσματα προσομοίωσης που επαληθεύουν την ακρίβεια αυτών των μοντέλων. Πραγματικά ίχνη δικτύου χρησιμοποιούνται όπου θέλουμε να υποστηρίξουμε περαιτέρω τη λογική για την πρόταση ενός συγκεκριμένου σχήματος. Ένα βασικό εργαλείο για τη μοντελοποίηση και την ανάλυση των προβλημάτων συνεργασίας σε δίκτυα με αυτόνομους κόμβους είναι η θεωρία παιγνίων, η οποία χρησιμοποιείται σε μερικά τμήματα αυτής της διατριβής για να βοηθήσει στην εξακρίβωση της δυνατότητας διατήρησης της συνεργασίας μεταξύ των κόμβων στο δίκτυο. Με την αξιοποίηση των κοινωνικών χαρακτηριστικών των κόμβων, μπαίνουμε επίσης στον τομέα της ανάλυσης των κοινωνικών δικτύων, και χρησιμοποιούμε σχετικές μετρικές και τεχνικές ανάλυσης.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Δίκτυα Επικοινωνιών

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: δικτύωση, κατανεμημένοι αλγόριθμοι, αυτόνομοι κόμβοι, συνεργατικότητα, κοινωνικά δίκτυα

ABSTRACT

This thesis contributes to the literature by proposing and modeling novel algorithms and schemes that allow the tasks of information dissemination and retrieval – and more generally of content management – to be performed more efficiently in a modern networking environment. Apart from information dissemination and retrieval, other aspects of content management we examine are content storage and classification. The most important challenge that will preoccupy many of the proposed schemes is the need to manage the autonomy of nodes while preserving the distributed, as well as the open nature of the system. In designing distributed mechanisms in networks with autonomous nodes, an important challenge is also to develop incentives for nodes to cooperate while performing communication tasks. A novel characteristic of most of the proposed schemes is the exploitation of social characteristics of nodes, focusing on how common interests of nodes can be used to improve communication efficiency.

In order to evaluate the performance of the proposed algorithms and schemes, we mainly develop mathematical stochastic models and obtain numerical results. Where it is deemed necessary, we provide simulation results that verify the accuracy of these models. Real network traces are used where we want to further support the rationale for proposing a certain scheme. A key tool for modeling and analyzing cooperation problems in networks with autonomous nodes is game theory, and it is used in parts of this thesis to help identify the feasibility of sustaining cooperation between nodes in the network. By exploiting social characteristics of nodes, we also enter the field of social network analysis, and use related metrics and techniques.

SUBJECT AREA: Communication Networks

KEYWORDS: networking, distributed algorithms, autonomous nodes, cooperation social networks

To kindness and logic

ACKNOWLEDGEMENTS

I would like to express my gratitude to my advisor, Professor Ioannis Stavrakakis, for the valuable guidance during this PhD work and continuous, tireless advice. Also I have learnt many things and would like to thank Dr. Ioannis Koukoutsidis and Dr. Merkouris Karaliopoulos for their cooperation, the many fruitful discussions I had with them, their help and suggestions related to my research work. In addition, I am grateful to Dr. Nikos Laoutaris, Dr. Athanasios Vaios, Dr. Antonis Panagakis, Dr. Costas Vassilakis, Dr. Leonidas Tzevelekas, Asst. Professor Kostas Oikonomou, Dr. Dimitrios Kogias, Panagiotis Pantazopoulos and Evangelia Kokolaki for their support and valuable comments on this thesis.

Many people have contributed in my life, directly or indirectly, helping me to reach this stage. First and foremost my parents, my brother Arseni, my sister Ina and my lovely grandmother, whose efforts cannot be easily acknowledged by a few words. I am also thankful to my sweet nephews Marios and Filippos, and my beautiful goddaughter Katerina for bringing me joy and happiness. Last but not least, I would like to thank my husband Georgios Papadopoulos for his patience during the writing of this thesis.

I really hope this thesis makes all the people I love very proud.

Eva Jaho

Athens, December 2011

Contents

List of Figures

List of Tables

Chapter 1

A general view of information dissemination and retrieval in modern networks

1.1 Introduction

In our time, the proliferation of networking technologies has had a huge impact on the way information is disseminated and retrieved. This proliferation was largely due to three reasons: The first is the development of the Internet, which greatly facilitated the interworking of different communication protocols and technologies, by introducing the common TCP/IP protocol suite [Kurose & Ross, 2004]. The second is the development of ad-hoc wireless mobile networks, in which mobile devices can communicate without the need for a fixed infrastructure [Sarkar *et al.*, 2007]. The third is the evolution of the hardware and software of user equipment, which allowed devices to communicate directly between themselves (as in peer-to-peer networks [Liben-Nowell *et al.*, 2002]), to be more independent and intelligent, and to partially abandon the traditional client-server model, where all the information is stored at a single computer with large processing capabilities (server), and clients make requests for information at the known location of the server.

Another major development over the last two decades was the gradual movement from the notion of information, to that of content, with reference to the data carried over by a network. While the term information typically referred to a piece of knowledge, understandable by humans, and transmitted in the form of data by a computer, the term content refers to data organized in the form of a file, such as a document, an audio or video file, which conveys an abundance of useful information to its reader, but requires more management. Together with the organization of information into content, the other characteristic was the fast multiplication of content sources; nowadays, any user device can create and publish its own content, which can be transmitted over the Internet. This has created opportunities for faster dissemination of knowledge, and allowed the provision of information with a more specialized local and temporal scope.

Nevertheless, when considered together, these developments also pose many challenges. The ultimate goal of communication anytime and anywhere cannot be achieved unless communication is meaningful and efficient. To achieve this, algorithms for information dissemination and retrieval must be able to exploit the ability of user devices to communicate directly, exploit the opportunity to obtain content from different sources, and at the same time handle the problem of information explosion caused by the abundance of data.

This thesis aims to provide an overview of some of the latest developments in the field of network information dissemination and retrieval, while focusing at proposing and modeling the performance of novel algorithms that allow these tasks to be performed more efficiently. When speaking of network information dissemination and retrieval, we refer to the delivery and retrieval of content to or from nodes in the network, that respectively request or possess the content. We do not go into the details of the content that is disseminated, or in the process of searching for information in documents or databases, that are also parts of information dissemination and retrieval, but focus on the networking algorithms and procedures for this task, that deal with content as an 'object' to be transferred at one or more destinations over the network.

In the rest of this introductory chapter we survey the developments in the field that have led to the present stage, and present in more detail the characteristics of modern networks and the challenges they pose for information dissemination and retrieval. Finally, we provide a more detailed outline of the chapters that follow, and what we aim to investigate in each one.

1.2 Classical information dissemination: routing schemes

Information dissemination is tied to the existence of a routing scheme, which handles the task of finding the path from a source to one or more destinations in the network. The packets are forwarded through intermediate nodes, typically hardware devices called routers, bridges, gateways, firewalls, or switches. In modern networks, user devices can themselves act as intermediates, albeit with limited functionality.

Standard shortest-path routing schemes (*e.g.*, OSPF [Moy, 1998], RIP [Malkin, 2000], BGP [Caesar & Rexford, 2005]) have long been used in fixed networks and perform remarkably well in network environments where traffic is low and conditions (topology, traffic, *etc.*) vary very slowly (in months or years). However, in more dynamic environments such as ad-hoc networks, where topology changes more quickly and traffic is unpredictable, these routing schemes frequently exhibit oscillatory behaviors and cause performance degradation [Bertsekas, 1982]. Dynamic shortest-path algorithms alleviate this problem to some extent, by adapting to less slowly changing conditions [Wang & Crowcroft, 1992]. Almost all of them are based on estimating link metrics based on past measurements. The idea is that when traffic changes slowly, link metrics are closely correlated in time, so that we may predict future conditions based on past observations. However, it was shown that in cases where traffic is heavy and changing rapidly, the estimation error becomes large and these schemes also fail [Wang & Crowcroft, 1992]. Further, another drawback is the lack of scalability; when the network consists of hundreds or thousands of nodes and conditions change quickly, the required routing updates may overload the network resulting in congestion and loss of packets.

These observations became extremely important with the advent of mobile ad-hoc networks (MANETS) in the 1990s. It was observed that in such networks, intelligent routing strategies are required to efficiently use the limited wireless resources while at the same time being adaptable to the changing network conditions such as network size and traffic density. Additionally, these networks have to deal with the problems of limited power supply of the mobile devices, as well as the problem of partial connectivity (network partitioning). Routing schemes proposed for MANETs take full advantage of the ability of devices to act as routers and communicate directly between themselves. They include global/proactive, ondemand/reactive and hybrid schemes [Royer & Toh, 1999, Abolhasan, 2004]. In proactive routing protocols, the routes to all destinations (or parts of the network) are determined at start up, and maintained by using a periodic route update process. Generally, such schemes are also based on updating routing tables based on shortest-path algorithms, with the addition of some information for characterizing the freshness of a path. In reactive protocols, routes are determined when they are required by the source using a route discovery process. Proactive schemes for MANETs are known to also suffer from scalability problems [Abolhasan, 2004]. Reactive schemes were invented to combat this deficiency, however their disadvantage is increased latency for setting up the route to the destination, as well as increased overhead, since the packet has to carry the whole path from source to destination [Abolhasan, 2004]. Hybrid routing protocols combine the basic properties of the first two classes of protocols into one, and can present improved performance with respect to scalability, latency and packet overhead.

Further advancement of routing schemes came along with a new model for MANETs, whose main characteristic was partial connectivity and opportunistic encounters between nodes, along with delay tolerance. The previous schemes for MANETs worked relatively well if the network graph was connected, or if disconnections did not last very long. However, they failed if significant portions of the network remained disconnected for large periods. This condition created the need for simpler routing protocols, which consist of independent and local store and forwarding decisions, based on the current connectivity information and possible predictions of future connectivity. If different links come up and down, over time, due to occasional partial connectivity or node mobility, and the sequence of connectivity graphs over a time interval are overlapped, then an end-to-end path might exist.

Such schemes usually fall into the categories of opportunistic routing, or epidemic routing/gossiping. Opportunistic routing schemes concentrate on forwarding the packet to the node that has the highest chance of successful delivery. This is estimated based on previous history [Boldrini *et al.*, 2007], or the estimated distance of the next node to the destination [Biswas & Morris, 2004]. Epidemic or gossiping schemes use some form of constrained flooding, where each node can forward or receive a message under some conditions. These conditions may relate to the characteristics of the message (*e.g.*, size, destination), as in [Vahdat & Becker, 2000], or not; in a popular version of a gossiping algorithm, each node forwards each message with some probability [Haas *et al.*, 2006]. These schemes are generally scalable and simple to implement, and under specific conditions can guarantee the delivery of the message to all nodes in the network. At the same time, the number of message copies that are spread in the network is significantly reduced, compared to a flooding scheme.

In this thesis, we analyze and evaluate the performance of a gossiping algorithm, with several parametrizations, for both content dissemination and search. Although in other chapters we are not explicitly concerned with the details of routing, but with optimizations of some parts of the content dissemination or retrieval processes, it should be kept in mind that a fully prescribed routing algorithm should always exist in order to accomplish the transfer of information.

1.3 Content replication techniques

Another important issue related to content dissemination and retrieval in modern networks is that of content replication. The term replication refers to storing the same content at multiple nodes in the network, in order to increase its availability and reduce its access cost. Replication is nowadays a standard operation used by large content and service providers in order to meet the increased demand for Web services and mitigate congestion problems on fixed networks. Having realized its potential, research is also being conducted to carry over these ideas on MANETs.

It is worth noticing the differences between replication and caching, as these terms are often intermingled in the literature. Although they both refer to the storage of information, in replication objects are stored at a node for a longer term by a process that is carried out independently of object requests at this node. In caching, objects are stored in local memory as a result of query execution, and a replacement policy must be applied (*e.g.*, LRU) if the memory becomes full [Reed & Long, 1996]. Furthermore, in replication the placement of objects is a global problem affecting the whole network. Although algorithms for caching also exist in distributed environments, in caching the storage decisions are usually based on local parameters, such as the local request rate for objects and the local memory space available [Gadkari, 2008]). Finally, there are also other different technical parameters, such as different memory access and database maintenance algorithms.

The earliest application of replication techniques can be found in Content Delivery Networks (CDNs). A CDN replicates content from the origin server to cache servers, scattered over the globe, in order to deliver content to end-users in a reliable and timely manner from nearby optimal surrogates. Typical customers of a CDN are media and Internet advertisement companies, data centers, Internet Service Providers (ISPs), online music retailers, mobile operators, and other carrier companies and content providers. The main tasks that must be carried out by a CDN are: content delivery, request routing, distribution, and accounting [Pathan & Buyya, 2006]. The content delivery task consists of delivering copies of content to the end-users by servers that are strategically located at the edges of a network. Request routing refers to directing client requests to the appropriate edge servers. Distribution has to do with moving content from the origin server to the CDN edge servers and ensuring consistency of content in the caches. The distribution and request routing infrastructure interact to keep an up-to-date view of the content stored in the CDN caches in appropriate databases. The accounting infrastructure maintains logs of client accesses and records the usage of the CDN servers. This information is used for traffic reporting and usage-based billing. Finally, other tasks carried out by a CDN are backup and disaster recovery solutions, as well as monitoring, performance measurement and reporting.

The goal of optimal edge server placement is to reduce user perceived latency for accessing content and to minimize the overall network bandwidth consumption for transferring replicated content from servers to clients. In this context, some theoretical approaches have been proposed which model the server placement problem as the center placement problem defined as follows: for the placement of a given number of centers, minimize the maximum distance between a node and the nearest center (minimum k-center problem, k-hierarchically well separated trees [Jamin *et al.*, 2000]).

Another way to improve content delivery is by selecting the right content to be delivered to the end-users. An appropriate content selection approach can assist in reduction of client download time and server load. A model that has frequently been used for theoretical studies of content placement is graphically illustrated in Fig. 1.1:

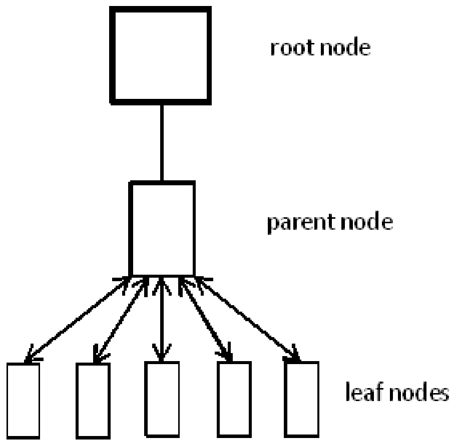


Figure 1.1: Model for theoretical studies of content replication (source: [Borst *et al.*, 2010]).

The root node represents the origin server which is the source for content requested by end-users at the leaf nodes. The parent node acts as a proxy between the root and the leaf nodes, for handling content requests and delivery, or as a proxy for communication between the leaf nodes. Which content to place locally on each leaf node depends primarily on the popularity of objects at the leaf nodes, but also on other factors such as the local storage sizes and communication costs [Borst *et al.*, 2010].

In early CDNs, edge servers were machines dedicated to the content replication tasks, and almost all required operations were under a centralized control. However, as the demand for replication services increased, distributed algorithms and architectures with better scalability were proposed. In such algorithms, leaf nodes could be normal user devices, which could decide for which content to store in a distributed manner, and maintain a local database of other nodes' placements [Mulerikkal & Khalil, 2007]. Although some centralized functions are still necessary (*e.g.*, for updating local databases), these algorithms can have much better scalability without the need for dedicated edge servers. Simplified models for content placement (based on the pattern of Fig. 1.1) showed that optimal content placement obeys rather simple rules [Borst *et al.*, 2010], and that efficient distributed algorithms can be constructed in which all leaf nodes can reduce their access cost [Laoutaris *et al.*, 2006].

The design of distributed algorithms was the first step towards the application of replication strategies in MANETs. In such networks, replication could also assist in reducing the cost of accessing content from the fixed network through congested access points, or through multiple hops and lossy wireless links. There are however, additional challenges that must be met in order to apply an efficient distributed algorithm in such networks. First, the existence of network partitioning, which reduces the data availability when the server that holds the desired data is not in the same partition as the client node. A good algorithm that overcomes such problems should estimate possible partitions beforehand and replicate the same content at different partitions of the network. Secondly, the energy consumption issue, as mobile nodes have limited energy resources. A good algorithm should replicate data so as to balance requests at mobile nodes, in that way that energy consumption is minimal. Further, even if distributed algorithms scale better, problems may appear in networks with thousands of nodes or in networks with highly mobile nodes. In such networks, the number of content queries or database updates may be so huge that links may experience congestion. Finally, a major challenge is how to deal with the requests for real-time applications, which also have stringent delay requirements. Good reviews of replication algorithms addressing these challenges can be found in [Padmanabhan *et al.*, 2008, Derhab & Badache, 2009]. As these authors point out, there are still no algorithms addressing all of the above issues successfully.

1.4 Characteristics of “modern” networks

In classical communication networks based on the telephone network, intelligence was in the network rather than in the devices. The core routers and switches handled all the basic operations in the network, which was dominated by wireline transmissions and a fixed (or very slowly varying with time) structure. In the past decade this is gradually changing, mainly due the proliferation of wireless transmission, ad-hoc network structures, and intelligent devices with greater processing capabilities. Furthermore, we have seen the appearance of techniques for constructing overlay networks, virtually changing the network structure and using tunneling to facilitate peer-to-peer communication.

Two major characteristics that motivate this thesis are the evolution in device capabilities and the changes in network structure. The evolution in device capabilities includes advancement in many aspects, such as main processor cores, digital signal processing techniques and communication capabilities, increased memory sizes, graphic displays with video capabilities, more friendly user-interfaces, *etc.*. This has led to the creation of ultra-portable devices, able to handle many of the tasks of a personal desktop computer[1].

The most prominent change in structure can be seen in wireless ad-hoc networks, in which devices can move freely and are able to communicate between themselves without having to rely on a fixed infrastructure. The lack of a fixed infrastructure and node mobility has led to dynamic topologies, to which – as seen in Section 1.2 – routing and, more generally, content dissemination and retrieval techniques must be adapted.

A rapidly changing network structure can however create scaling problems, which means that the network performance may drop by increasing the number of nodes in the network, their mobility, the amount of messages transmitted, or some other parameter. This, together with the need to avoid central points of failure and to have terminal devices that are easily reconfigurable and can adapt to different conditions, has created the need to develop distributed algorithms for message dissemination and retrieval. In designing such algorithms, the effort is to efficiently manage (retrieve, classify, store and disseminate) content without having a central coordinating entity, relying on the (often partial or limited) information that terminal devices can acquire from other terminals or their environment. This leads to the point that terminal devices (nodes) become autonomous, *i.e.*, administered by themselves. The autonomicity of nodes implies their ability to deviate from prescribed distributed mechanisms, and has to be considered from the very beginning in the design of such mechanisms [Mitchell & Teague, 2003]. Autonomous nodes and distributed mechanisms have also led to the notion of self-organizing networks, which self-optimize parameters and algorithmic behaviour in response to observed network performance and radio conditions[2].

A key to designing efficient content dissemination and retrieval techniques is also understanding the high-level structure of such networks. Important steps in this direction

[1] $http://en.wikipedia.org/wiki/Converged\_device$

[2] $http://en.wikipedia.org/wiki/Self-organizing\_network$

were the discovery of the small-world phenomenon and of techniques for identifying community structure. A small-world network is defined to be a sparse network where most nodes are not neighbours of one another, but each node can be reached from every other by a small number of hops or steps. The two distinguishing structural properties are a small average shortest path length (growing proportionally to the logarithm of the number of nodes N in the network), and a clustering coefficient significantly higher than expected by random chance [Watts & Strogatz, 1998]. Several other properties are often associated with small-world networks. Typically there is an overabundance of hubs – nodes in the network with a high number of connections (a high degree). These hubs serve as the common connections mediating the short path lengths between other edges. Networks with a greater than expected number of hubs will have a greater fraction of nodes with high degree, and consequently the degree distribution will be enriched at high degree values. If this distribution can fit a power-law distribution, the network is an ultra-small world network, also known as a scale free network [Cohen *et al.*, 2002, Cohen & Havlin, 2003].

Recent work has shown that the physical connectivity of the Internet exhibits small-world behaviour [Jin & Bestavros, 2002]. Further, small-world network characteristics were discovered in many social networks, and ad-hoc networks that resemble them [Dousse *et al.*, 2002]. The small world structure can have a profound effect on the design of message dissemination algorithms; a first observation in this direction was the increased speed of epidemic algorithms in such networks [Watts & Strogatz, 1998]. Similar observations are very important in the development of secure peer-to-peer protocols, novel routing algorithms for the Internet and ad hoc wireless networks, and search algorithms for communication networks of all kinds.

Another important characteristic of modern networks is community structure (or clustering). A network is said to have community structure if the nodes of the network can be easily grouped into (potentially overlapping) sets of nodes such that each set of nodes is densely connected internally. The importance of community structure in communications networks comes mainly from their association with social networks, which often include community groups [3] based on common location, interests, occupation, *etc.*. Ad-hoc networks, being composed of portable devices carried by humans and having a

[3]The term community structure in fact originates from social networks.

friendly user interface, often exhibit such network characteristics [Musolesi *et al.*, 2005]. Being able to identify these substructures within a network can provide insight into how network functions and topology affect each other, and drive the design of efficient content dissemination and retrieval algorithms. Several methods for community finding have been developed and employed with varying levels of success. Examples are the minimum-cut method, hierarchical clustering [Newman, 2004c], the Girvan-Newman algorithm [Girvan & Newman, 2002], modularity maximization algorithms [Good *et al.*, 2010], *etc.*.

1.5 Challenges and objectives of modern information dissemination and retrieval schemes

The evolution of networks that we discussed in the previous sections has created new challenges for information dissemination and retrieval schemes, and set new objectives in the design of such schemes.

First of all, new challenges arise because of the limitations of mobile devices. Mobile devices are energy-constrained, and have limited memory and processing capabilities. This holds true despite technological advancements, and is aggravated by the information explosion that we witness in modern networks, *i.e.*, the rapid increase in the amount and volume of published information. The effect of this information explosion is multifold: it creates congestion in the network, increases interference, consumes battery power and carries the risk of information overload. The latter is a term from cognitive psychology [Allen & Shoard, 2005] and is related to the inability to efficiently manage and exploit information when there is an abundance of data. In view of these, an information dissemination or retrieval scheme must be fast, precise, avoid redundancy (sending duplicate messages to nodes), and offer useful or meaningful information to final recipients. Because devices are connected to their human users, information should be related to the interests or preferences of each user, or at least adapt to groups of users sharing common interests, in order to be useful.

Important mechanisms that can help in mitigating information explosion and reducing the risk of information overload are information retrieval and information filtering. An information filtering system is a system that removes redundant or unwanted information from an information stream using (semi)automated or computerized methods prior to presentation to a human user, based on the profile of this user and the characteristics of the information objects. An information retrieval system refers to a search system with which the user obtains information from the knowledge resources which best help her/him in problem management. These mechanisms are interrelated [Belkin & Croft, 1992], as their underlying goal (to provide the user with useful information) is essentially the same.

The most important challenge that will preoccupy many of the schemes proposed in this thesis is the need to manage the autonomy of nodes while preserving the distributed, as well as the open nature of the system. As described in the previous section, distributed systems evolved mainly out of the need to deal with scaling problems in large and highly mobile systems. An open system further helps in improving interoperability, as any node which understands a communication protocol can communicate within a network. This in turn helps to promote innovation and provide new mechanisms and technologies that further improve performance. It is widely accepted, for example, that the open and distributed nature of the Internet is one of the reasons behind its tremendous growth[4].

However, the autonomy of nodes raises serious control problems. It is reasonable to expect an end-device to be programmed (either by its manufacturer, an operator or by the end-user controlling it) to optimize its own performance, regardless of its environment. However, this may be at the expense of other nodes, or of the overall network operation. A classic example is multi-hop forwarding in an ad-hoc network where, in order to increase the overall capacity of the network and reduce the total energy consumption, nodes must be willing to relay packets of other nodes [Gupta & Kumar, 2000, Zhao & Tong, 2005]. However, this cannot be taken for granted as relaying packets is energy consuming, and unless all nodes contribute in relaying, it may turn out unfavorable for the relaying node. In general, in many distributed networking problems (e.g., power-control mechanisms, random access in wireless media [Félegyházi & Hubaux, 2006]), nodes must be willing to cooperate in order to incur an overall benefit for the network, i.e. a social benefit. A node is called selfish if it tries to extract a benefit for itself at the expense of other nodes. Furthermore, there may be malicious nodes, which attempt to undermine quality and

[4] $http://ec.europa.eu/information\_society/policy/ecomm/doc/library/communications\_reports/netneutrality/comm-19042011.pdf$

disrupt communications without necessarily attaining a reasonable benefit (attackers). The existence of such nodes further raises security concerns, and creates a need to embed security mechanisms in distributed systems.

Cooperation in message relaying is very much relevant and must be taken into account when considering information dissemination and retrieval schemes in ad-hoc networks. Further, node cooperation comes into play when considering content storage and exchange [Chuah *et al.*, 2010, Buttyán *et al.*, 2010].

In designing distributed mechanisms in networks with autonomous nodes, the goal is to develop incentives for nodes to cooperate. These incentives may take the form of a charging/rewarding mechanism (*e.g.*, [Salem *et al.*, 2006]), of a reputation scheme for building trust relationships between nodes (*e.g.*, [Kamvar *et al.*, 2003]), or the application of reciprocal punishments between non-cooperative nodes [Marbach & Qiu, 2005]. Game theory is a key tool for modeling and analyzing cooperation problems in networks with autonomous nodes. It is used in several parts of this thesis to help identify the feasibility of sustaining cooperation between nodes in the network.

1.6 Outline of the thesis

The focus of this thesis lies on the use of cooperative mechanisms for content management in networks with autonomous nodes. The aspects of content management we examine are mainly content dissemination and retrieval, but also content storage and classification. A novel characteristic is the exploitation of social characteristics of nodes in such networks, in order to manage content more efficiently. In this way we also enter the field of social network analysis, focusing on how common interests of nodes can be exploited to improve communication, through the several aspects of content management.

Social network analysis is a key technique in modern sociology and has recently gained a significant following in information science and computer networks, along with many other disciplines [Wellman, 1996]. This has been aggravated by the emergence and tremendous growth of online social networking services or platforms (*e.g.*, Facebook, Twitter, LinkedIn, *etc.*[5]), that allow the creation of ties among individuals that share ideas,

[5] $http://en.wikipedia.org/wiki/Social\_networking\_service$

interests or activities. From a research perspective, these online platforms are a source of valuable information about how the interest profiles of users are related to ties among them and affect information exchange.

The structure of the chapters of the thesis and the main objectives in each one are as follows.

In Chapter 2 we study a content replication scheme in which autonomous nodes form a group, called a distributed replication group and cooperate in order to effectively retrieve information objects from a distant server. Each node locally replicates a subset of the server objects and can access objects stored by other nodes in the group at a smaller cost, compared to the cost of accessing them from the server. Given that nodes are autonomous and independently decide which objects to replicate, the problem is to construct efficient distributed algorithms for content replication that induce low overall average access cost. This problem becomes even more challenging when the group has to deal with churn, *i.e.*, random "join" and "leave" events of nodes in the group; churn induces instability and has a major impact on cooperation efficiency. Given a probability estimate of each node being available, we propose a distributed churn-aware object placement strategy. By considering a game-theoretic approach, we identify cases where the churn-aware strategy is individually rational for all nodes, while the churn-unaware is not. Numerical results further show that the algorithm outperforms, in most cases, its churn-unaware counterpart, and allows for a more fair treatment of nodes according to their availability frequency, thus inciting nodes to cooperate.

Based on this setting, in Chapter 3 we study the impact of the similarity in nodes' preferences or interest profiles on content replication. Our aim is to investigate and draw important conclusions and guidelines regarding the kind of content placement strategy a node participating in a distributed replication group should follow in order to increase its benefits. We define a metric that captures the similarity of nodes' interest profiles, called group tightness. Using this metric and testing with different interest profiles, we are able to show the association of the degree of interest similarity within nodes in a group with the benefits they incur by applying a cooperative or selfish replication strategy.

An important as well as anticipated conclusion from this chapter is that the higher the interest similarity between nodes, the higher the gains by cooperation in content management. It is therefore reasonable to attempt to organize nodes into communities where nodes share similar interests. However, in current "computerized" social networks users do not necessarily create ties based on common interests, but also on many other factors, such as friendship, kinship, professional relations, or even prestige. The result is a relatively small tightness of such groups, and relatively poor gains from cooperation. In Chapter 4, we propose a framework for the construction of communities based on common interests of users, by building a virtual graph where an edge between two nodes is weighted by the degree of similarity in their interests, and then using known community detection algorithms to establish communities. Testing on synthetic network scenarios shows that this framework helps to correctly identify interest communities, stressing that care must be taken on the proper choice of the similarity metric used to represent weights.

Besides common interests, a major characteristic of communities is their locality, i.e. the specific neighbourhood, venue, or spot where they are located. Some localities may constitute points of attraction or hot-spots with a higher node density. Mobile nodes also form social groups dynamically, as they move to different localities where they can establish communication with other nodes. Social groups that are examined from the viewpoint of the locality they are situated in, are termed locality-induced groups. In Chapter 5, we investigate the intermingling of interest and locality-induced social groups and propose an approach that can enhance content dissemination in the presence of such groups. We assume a setting where nodes have different interest distribution patterns over a set of information objects, and different frequencies of visiting a number of localities. Considering a new metric for the valuability of content, that takes into account both its usability and discover-ability, we explore the conditions under which a cooperative strategy can improve the content dissemination process compared to a selfish one.

We further investigate content dissemination under node mobility in Chapter 6, by considering a so-called nomadic sensor network consisting of: a) sensor nodes, that are fixed at some points and collect information about states or variables of the environment, and b) mobile nodes that collect and disseminate this information. Mobile nodes are assumed to be interested in different subsets of sensor node information. Similarly to custom multi-hop forwarding, dissemination of information content in such networks can be achieved at smaller costs if mobile nodes are cooperative and collect and carry information not only in their own interest, but also in the interest of other mobile nodes. A specific modeling scenario is considered where mobile nodes move randomly on a graph, collecting information from mobile nodes located at the vertices. We present a game-theoretic analysis to find conditions under which a cooperative equilibrium can be sustained.

In Chapter 7 we study gossip-based algorithms for content dissemination and search in large-scale networks with autonomous nodes. The dissemination or search process is carried out in rounds, where at each round multiple peers can be contacted. We develop an analytical model that allows us to evaluate the performance of the algorithm as well as the impact of several design parameters, such as the degree of cooperation of nodes, the number of peers contacted in each round, or the number of nodes where a searched content may be located. We also consider the degree of information a node has about the evolution of the gossiping process, meaning the number of nodes contacted so far, and study both the case where a node has complete information and the case of no information. The results provide significant insights on the design of such schemes.

Finally, Chapter 8 presents collectively the major conclusions of the thesis and provides high-level directions for future research.

This thesis is based on material from the following papers:

Journal articles (3)

1. E. Jaho, I. Koukoutsidis, I. Stavrakakis, I. Jaho, “Cooperative content replication in networks with autonomous nodes”, Computer Communications, Elsevier (accepted December 2011)

2. S. Tang, E. Jaho, I. Stavrakakis, I. Koukoutsidis, P. Van Mieghem, “Modeling Gossip-based Content Propagation and Search in Distributed P2P Overlay”, Computer Communications, Elsevier, Vol. 34, No. 6, pp. 765-779, May 2011

3. I. Antoniou, I. Koukoutsidis, E. Jaho, A. Pitsilidis, I. Stavrakakis, “Access Network Synthesis Game in Next Generation Networks”, Computer Networks, Elsevier, vol. 53, no. 15, pp. 2716-2726, October 2009

Conference/workshop proceedings (8)

1. M. Karaliopoulos, P. Pantazopoulos, E. Jaho, I. Stavrakakis, "Trace driven Analysis of Data Forwarding in Opportunistic Networks", Second Conference on the Analysis of Mobile Phone Datasets and Networks (NetMob 2011), MIT (Media Lab), Cambridge, MA, USA, October 10-11, 2011

2. E. Jaho, M. Karaliopoulos, I. Stavrakakis, "ISCoDe: a framework for interest similarity-based community detection in social networks", Third International Workshop on Network Science for Communication Networks (NetSciCom 2011), colocated with INFOCOM 2011, Shanghai, China, April 15th, 2011

3. E. Jaho, M. Karaliopoulos, I. Stavrakakis, "Social Similarity as a Driver for Selfish, Cooperative and Altruistic Behavior", Fourth International IEEE WoWMoM Workshop on Autonomic and Opportunistic Communications (AOC 2010), Montreal/QC Canada, June 14th, 2010

4. E. Jaho, I. Stavrakakis, "Joint Interest- and Locality-Aware Content Dissemination in Social Networks", Sixth Annual Conference on Wireless On demand Network Systems and Services (IFIP/IEEE WONS 2009), Snowbird, Utah, USA, February 2-4, 2009

5. R. Cuevas, E. Jaho, C. Guerrero, I. Stavrakakis, "OnMove: A Protocol for Content Distribution in Wireless Delay Tolerant Networks based on Social Information", Fourth International Conference on emerging Networking EXperiment and Technologies (ACM CoNEXT 2008 Student Workshop), Madrid, Spain, December 9th, 2008

6. E. Jaho, I. Koukoutsidis, I. Stavrakakis, I. Jaho, "Cooperative Replication in Content Networks with Nodes under Churn". IFIP Networking 2008, Singapore, May 5-9 2008.

7. I. Koukoutsidis, E. Jaho, I. Stavrakakis, "Cooperative Content Retrieval in Nomadic Sensor Networks", IEEE INFOCOM MObile Networking for Vehicular Environments Workshop (MOVE 2008), Phoenix, AZ, USA, April 18th, 2008

8. E. Jaho, I. Jaho, I. Stavrakakis, "Distributed Selfish Replication under Node Churn", Sixth IFIP Annual Mediterranean Ad Hoc Networking Workshop (MED-HOC-NET 2007), Poster Session, Corfu, Greece, June 12, 2007

Under review (2)

1. S.M. Allen, M.J. Chorley, G.B. Colombo, E. Jaho, M. Karaliopoulos, I. Stavrakakis, R.M. Whitaker, "Exploiting user interest similarity and social links for micro-blog forwarding in mobile opportunistic networks", Pervasive and Mobile Computing, Elsevier (submitted September 2011)

2. E. Jaho, M. Karaliopoulos, I. Stavrakakis, "Analysis of content placement strategies based on social similarity", IEEE Transactions on Parallel and Distributed Systems (submitted June 2011)

Chapter 2

Cooperative content replication

Content or data replication in a network refers to the storage of information objects (such as files and other software entities) in multiple interconnected local nodes, so that the availability of the objects increases and they can be accessed by users at a smaller cost – compared to the cost of being stored in a single source. The technique of replication is largely applied in Content Distribution Networks (CDNs), where content from a core server is replicated at multiple servers (usually placed at the edge of the provider's network) [Loukopoulos et al., 2002, Kangasharju et al., 2001]. The same principle of replication can be used to facilitate content retrieval and exchange in other network structures, such as peer-to-peer networks, mobile ad-hoc networks, social network, wireless mesh networks, etc..

2.1 Introduction

In a peer-to-peer network, the cost to reach some nodes may be significantly high, *e.g.*, because of long distances, low-speed links or congested switches. Properly distributing replicas of content in multiple nodes in the network has shown to provide smaller search and retrieval times for content, and to decrease the overall network load [Cohen & Shenker, 2002]. In mobile networks, there has been in the recent years a tremendous increase in the volume of downloaded data from the Internet, which may result in congestion in wireless access links. Replication techniques have also been proposed in this case to take advantage of device-to-device communication capabilities, reduce content retrieval times and mitigate congestion in Internet access links [La *et al.*, 2010, La *et al.*, 2011, Derhab & Badache, 2009]. Despite the fact that in a mobile network the topology dynamically changes over time, there has been significant evidence that non-random clustered mobility characterizes human movements in outdoor environments [Lim *et al.*, 2006]. That is, despite node mobility, there is a tendency for the formation of groups composed of nodes which are in geographical proximity for a relatively long period of time and have high connectivity. This is the key fact that allows replication strategies to be extended to such networks, since it allows nodes to rely on other nodes in order to retrieve content.

User nodes in the networks described above typically have very small memory capacity and experience high costs in retrieving content from distant servers at a provider's fixed network. Therefore, each user would potentially have significant gains by retrieving objects from peer nodes instead of the original server. The main issue, however, is what objects to replicate at each node so that a global benefit exists.

Further, a primary characteristic of the above network structures is that nodes are autonomous, in that they can decide independently which information to store. A "globally beneficial placement" is much harder to achieve in a network with autonomous nodes than in a network under a centralized control, as in the former nodes behave as rational entities that aim at minimizing their own access cost. Such behaviour may be at the expense of others, as the individual cost of each node depends only on its local requests for objects; nodes do not primarily "care" for storing objects to serve requests originating at other nodes, *i.e.*, for the common welfare. However, despite the selfishness of nodes, a distributed algorithm could allow for an "implicit cooperation", simply by having each node view objects replicated at other nearby nodes and replicate objects in an attempt to minimize its total cost for all its requested objects. Such a cooperative replication would lead to the creation of an "enhanced local storage space", if different nearby nodes would replicate different objects.

We address this problem in a game-theoretic context, where nodes are the players. The number of objects each node requests is typically higher than its memory capacity. Each player implements a placement strategy that consists of choosing which objects to replicate locally in its limited storage space, at one or more occasions in the game ("rounds"). The goal of each player is to minimize the total access cost for all its requested objects at the end of the game.

A node may choose not to cooperate with other nodes in deciding which objects to replicate, in which case the optimal strategy is to replicate its most highly requested objects. This is called the *selfish* or *greedy local* strategy. Naturally, selfish nodes who are not interested in the social benefit would want to cooperate with others only if they could reduce their access cost compared to their incurred cost when all nodes follow the greedy local strategy. The latter cost is the so-called security level of a player and the above requirement is the *participation* or *individual rationality constraint* of a player. A strategy which satisfies the participation constraint for a player is called individually rational for this player. Ensuring that participation constraints are satisfied for all players is key to maintaining a cooperative scheme.

We use the model introduced in [Leff *et al.*, 1993] where nodes are self-organized into what we call a "replication group", *i.e.*, a group consisting of nodes in network proximity, where the cost of each node to retrieve locally stored content from another node in the group is about the same – and small compared to the cost of retrieving it from the origin server. The low-medium cost associated with fetching an object from within the group may reflect low actual or virtual price such as lower access delay due to locality or high connectivity, or higher level of trust and reliability. We assume that nodes have established trust relationships in order to belong to the same group, and have access to each others' caches. This may be realized through various schemes (*e.g.*, see [Rahman & Hailes, 1997, Gil & Ratnakar, 2002, Shikfa *et al.*, 2009]). Requests for objects are handled in the following manner. First, a user's request is received by the local node the user is associated with. If the requested object is stored locally, it is returned to the requesting user immediately, incurring a minimum access cost. Otherwise, the requested object is searched for and fetched from another node in the group, at a higher access cost. If the object can not be located anywhere else in the group, it is retrieved from an origin server assumed to be outside the group incurring a maximum access cost. Note that unlike caching, replication refers to storage of objects for a longer term, and no replacement policy (*e.g.*, LRU) is applied for each new object request.

We consider that each node knows the placements of other nodes before playing. This information can be provided by a central coordinating entity, or if each node announces its local placement and broadcasts any changes in it to other nodes in the group. Another significant aspect of groups with autonomous nodes is "churn", *i.e.*, random changes in the set of participating nodes in the group, that may occur due to "join" and "leave" events. To account for node churn, we consider a probability estimate of each node to be available. This can be derived by monitoring the fraction of time a node is available online and allows access to objects in its memory. This estimate is called the reliability or availability of a node and is common knowledge to all the nodes in the group. Such probabilities can be obtained from a reputation measure, which can itself be derived in a distributed manner, by local interactions between nodes. (For a review of distributed algorithms for deriving reputation measures, readers may refer to [Avrachenkov *et al.*, 2007].) Apart from the placements and the reliability of other nodes, each node also has to calculate its own request frequencies for objects. Given this information is available, the algorithm runs at a low cost compared to an optimal strategy, as is shown in Section 2.4.

We may reasonably assume that the starting placement of each node is the placement resulting from the greedy local strategy. When a node joins a replication group, it makes the number of changes of replicated objects that will result in the greatest access cost reduction for it. We call this appropriately a greedy churn-aware object placement strategy. The performance of the algorithm is investigated extensively and is shown to reduce the access cost, compared to its churn-unaware counterpart (*i.e.*, when nodes erroneously consider other nodes to be always available) and the greedy local strategy.

2.2 Content placement strategies

Let $\mathcal{N} = \{1, 2, ..., N\}$ denote the set of the nodes (or players) in a replication group and let $\mathcal{M} = \{1, 2, ..., M\}$ denote the set of objects (or items) these nodes are interested in. Let R_m^n denote the preference probabilities of node n, for object m, and let $R^n = \{R_1^n, R_2^n, ..., R_M^n\}$; R_m^n can be viewed as the normalized rate of requests for object m by node n.

Let P_n denote the *placement* at node n, defined to be the set of objects stored locally at that node with storage capacity C_n. We assume without loss of generality that $|P_n| = C_n$

since a node can always gain by storing objects of interest locally, rather than having to retrieve them from an outside source. Let $\mathcal{P} = \{P_1, P_2, \ldots, P_N\}$ denote the global placement for the repliaction group and $\mathcal{P}_{-n} = \mathcal{P} \setminus P_n$ denote the set of placements for all group nodes but node n.

Let t_l, t_r and t_s denote the cost for accessing an object from the node's local memory, from another remote node within the replication group and from nodes outside the replication group or distant server, respectively; $t_l < t_r < t_s$. These costs are assumed to be the same for all nodes in order to simplify the analysis. If the nodes of the group are in proximity the access cost could either represent the additional latency incurred when fetching content or the bandwidth consumed when retrieving content, depending on the scenario of interest. Otherwise, the low-medium cost associated with fetching an object from within the group may reflect low price due to high level of trust for example.

Given an object placement $\mathcal{P}$, the mean access cost incurred to node n per unit time for accessing its requested objects is given by:

$$\mathcal{C}_n(\mathcal{P}) = \sum_{m \in P_n} R_m^n t_l + \sum_{\substack{m \notin P_n, \\ m \in \mathcal{P}_{-n}}} R_m^n t_r + \sum_{\substack{m \notin P_n, \\ m \notin \mathcal{P}_{-n}}} R_m^n t_s \ . \tag{2.2.1}$$

The first, second and third addends on the right hand side correspond to the mean cost for accessing objects locally, from other nodes of the group and from external sites if not found within the group, respectively.

In the game nodes play sequentially, not necessarily in a predetermined order. Such a dynamic game in which there is an ordering of players, and each player's moves have an effect on the utilities of players ordered before or after, is called a *Stackelberg* game, or leaders-followers game (see *e.g.*, [Basar & Olsder, 1999]). We examine the following replacement strategies.

Optimally altruistic strategy: The objects are stored in such a way that the total access cost for all nodes in the social group is minimized (*i.e.*, minimize $\sum_{n=1}^{N} \mathcal{C}_n(\mathcal{P})$). This problem can be transformed into a 0-1 integer programming problem.

Let $X_m^n = \begin{cases} 1, & \text{if } m \in P_n; \\ 0, & \text{otherwise} \end{cases}$ and $Y_m^n = \begin{cases} 1, & \text{if } m \notin P_n \text{ and } m \in \mathcal{P}_{-n}; \\ 0, & \text{otherwise.} \end{cases}$

The objective is to minimize the total access cost:

$$\sum_{n=1}^{N}\sum_{m=1}^{M}[X_m^n R_m^n t_l + Y_m^n R_m^n t_r + \prod_{j=1}^{N}(1-X_m^j)R_m^n t_s], \tag{2.2.2}$$

where

$$Y_m^n = (1-X_m^n)(1-\prod_{\substack{j=1\\ j\neq n}}^{N}(1-X_m^j)). \tag{2.2.3}$$

This is a special case of a quadratic programming problem, with zero diagonal elements, whose solution is very difficult [Luenberger, 2008]. Fortunately, it was shown in [Leff *et al.*, 1993] that this quadratic problem *reduces to* a 0-1 integer linear minimization problem (ILP) with objective function

$$f(X) = \sum_{n=1}^{N+1}\sum_{m=1}^{M} z_m^n X_m^n, \tag{2.2.4}$$

subject to $\sum_{n=1}^{N+1} X_m^n \geq 1, \quad 1 \leq m \leq M$ and $\sum_{m=1}^{M} X_m^n \leq C_n, \quad 1 \leq n \leq N$. In (2.2.4), the terms X_m^n are as above, the additional virtual node $N+1$ represents the ensemble of nodes in other social groups, and the terms z_m^n, $n \in \mathcal{N}$, are defined as
$$z_m^n = \begin{cases} R_m^n(t_r - t_l), & \text{for } 1 \leq n \leq N; \\ \sum_{j=1}^{N} R_m^j(t_r - t_s), & \text{for } n = N+1, \end{cases}$$

In this ILP formulation, there is effectively an implicit reference placement, whereby all nodes can access all objects from the caches of group nodes and aggregate access cost $\sum_{n=1}^{N}\sum_{m=1}^{M} R_m^n t_r$. The aim is then to derive object placements that improve over this reference placement. Hence, the terms z_m^n, $n \in \mathcal{N}$, express the incremental benefit resulting for each node when it stores the object locally instead of retrieving it from the group; whereas, z_m^{N+1} notes the loss all nodes incur if the object is not stored anywhere in the group.

Proposition 1. *The quadratic maximization problem described in (2.2.2) is equivalent to the minimization ILP problem in (2.2.4).*

Proof. The proof is based on argument in [Leff *et al.*, 1993]. Our starting point is the minimization objective (2.2.2). Substituting the terms Y_m^n from (2.2.3) and after some algebraic manipulations, (2.2.2) becomes:

$$\sum_{n=1}^{N}\sum_{m=1}^{M}\left[R_m^n(t_l - t_r)X_m^n + R_m^n(t_s - t_r)\prod_{j=1}^{N}(1-X_m^j) - R_m^n t_r\right] \tag{2.2.5}$$

The minimization of (2.2.5) calls for maximization of the first negative summation term and minimization of the second positive summation term; the third term is a constant corresponding to the cost of the reference placement, whereby all nodes access all objects from the caches of the group nodes. Maximization of the first term implies that all nodes should fully occupy their caches, *i.e.*, it is always better to store an additional object in the local cache rather than having vacant cache space. The product in the second term can be equivalently written as

$$X_m^{N+1} = \prod_{j=1}^{N}(1 - X_m^j). \tag{2.2.6}$$

and is equivalent to the condition

$$\sum_{n=1}^{N+1} X_m^n \geq 1, \tag{2.2.7}$$

when equality holds. Therefore, the minimization of (2.2.5) is equivalent to the minimization of the negative objective function, $-f(X)$, of the ILP

$$g(X) = -f(X) = -\sum_{n=1}^{N+1}\sum_{m=1}^{M} z_m^n X_m^n \tag{2.2.8}$$

if we let its two constraints holds as equalities, import the terms z_m^n with inverse sign, and omit the constant term $\sum_{n=1}^{N}\sum_{m=1}^{M} R_m^n t_r$.

Hence, the solution that minimizes the objective function (2.2.2) also maximizes the objective function $f(X)$ in (2.2.4). $\square$

Selfish strategy: Under the selfish strategy, the nodes store the objects *they* prefer most. Each node n ranks the objects in a decreasing order of preference and selects to store the first C_n ones. In [Laoutaris *et al.*, 2006], this strategy is also referred to as *greedy local*.

Self-aware cooperative strategy: The strategy involves two discrete steps. First, each node stores its C_n most preferable items (selfish placement). Then, nodes take turns in adjusting their placements based on the placements of the other nodes in the group. It is essentially the strategy proposed in [Laoutaris *et al.*, 2006]. During this second step and given the global placement P at its time of play, each node considers making replacements so as to minimize its current access cost according to (2.2.1) (hence the term "self-aware").

To describe the strategy, we need some further definitions. Let the *eviction loss* $L_{e,n}$ be the increase in the access cost that would incur if node n evicted object e, $e \in P_n$, from its memory. Let the *insertion gain* $G_{i,n}$ be the decrease in the access cost, that would incur if node n inserted object i, $i \notin P_n$. A node would decide to evict an object and insert another one only as long as the resulting insertion gain exceeds the eviction loss.

Under the self-aware cooperative strategy, it turns out that: (P1) a node may evict an object from its local memory only if it exists in one or more of the other nodes, (P2) a node may only insert an object that does not exist in any of the other nodes in the group. A proof of these properties is presented below along with related definitions and discussions that will be utilized later.

If a node n evicts an object $e \notin P_{-n}$, the eviction cost would be $R_e^n(t_s - t_l)$. This object must be replaced by an object $i \notin P_n$ for which, in the best case that $i \notin P_{-n}$, it would hold that $R_i^n(t_s - t_l) > R_e^n(t_s - t_l)$. Such an object i does not exist, though, since all such objects are already in P_n. Likewise, if node n inserts an object $i \in P_{-n}$, the insertion gain would be equal to $R_i^n(t_r - t_l)$. Since from the above a node only evicts an object that exists in one or more of the other nodes, i would replace an object e for which $R_i^n(t_r - t_l) > R_e^n(t_r - t_l)$. This also cannot hold, because if $R_i^n > R_e^n$ object i would have already been in P_n.

With these in mind, the eviction cost for an object e replicated at node n and at one or more of the other nodes in P_{-n} is given by

$$L_{e,n} = R_e^n(t_r - t_l) \ . \tag{2.2.9}$$

For an object i not replicated in any of the nodes in the group, the insertion gain to node n is equal to

$$G_{i,n} = R_i^n(t_s - t_l) \ . \tag{2.2.10}$$

An object $e \in P_n \cap P_{-n}$ is called an eviction candidate, whereas an object $i \notin P$ is called an insertion candidate for node n. The set of eviction candidates for node n is denoted as $\mathcal{E}_n$, where $|\mathcal{E}_n| \leq C_n$, and the set of insertion candidates as $\mathcal{I}_n$.

Index the eviction candidates for node n as $e_{1n}, e_{2n}, \ldots, e_{|\mathcal{E}_n|n}$, in order of increasing eviction cost. That is, $L_{e_1,n} \leq L_{e_2,n} \leq \cdots \leq L_{e_{|\mathcal{E}_n|},n}$ (the double subscript n is removed). Accordingly, index the insertion candidates for node n as $i_{1n}, i_{2n}, \ldots, i_{|\mathcal{I}_n|n}$ in order of decreasing insertion gain, *i.e.*, such that $G_{i_1,n} \geq G_{i_2,n} \geq \cdots \geq G_{i_{|\mathcal{I}_n|},n}$ (the double subscript n is likewise removed).

It is evident that in order to minimize its access cost node n should make changes in the following way: evict e_{1n} and insert i_{1n}, evict e_{2n} and insert i_{2n}, and so on, until the maximum number m_n such that $G_{i_{m_n},n} > L_{e_{m_n},n}$ ($m_n \leq \min(|\mathcal{E}_n|, |\mathcal{I}_n|)$). For an arbitrary eviction candidate e and insertion candidate i, we call $\Delta G_{(e,i)} \stackrel{\text{def}}{=} G_{i,n} - L_{e,n}$ the replacement gain for the eviction-insertion pair (e, i). A replacement of object e by object i is also denoted as $e \leftarrow i$.

Since each node only evicts an object that exists in one or more of the other nodes (P1) to insert an object absent from the group (P2), each node's moves do not increase the total access cost of nodes previously ordered in the play. Further, in line with (P1) and (P2), it has been shown for the self-aware cooperative placement strategy in [Laoutaris *et al.*, 2006] that the placements that result with the self-aware cooperative strategy do not give rise to node mistreatment phenomena (P3), *i.e.*, for any node n, it holds that $\mathcal{C}_n^C(\mathcal{P}) \leq \mathcal{C}_n^S(\mathcal{P})$, where $\mathcal{C}_n^C(\mathcal{P})$ and $\mathcal{C}_n^S(\mathcal{P})$ denote the mean access cost for node n under the self-aware cooperative and selfish strategy, respectively.

Moreover, no node can gain by playing again at any subsequent round in the game. Consequently, this strategy is individually rational for every player and the game ends in one round.

2.2.1 Effect of asymmetric costs on mistreatment

The three placement strategies have been discussed so far under the assumption that the replication group membership induces a distinct symmetrical level of access cost hierarchy; namely, the cost t_r^{ij} of node i to access an object from any group node j is the same irrespective of the involved nodes, *i.e.*, $t_l < t_r^{ij} = t_r < t_s, \forall i, j \in \mathcal{N}$. This assumption is rational whether the group draws on locality or other types of social context. Due to properties (P1)-P(3), the performance of the self-aware cooperative strategy has been shown to lie close to the optimal (social-cost minimizing) altruistic one. Before deriving conditions under which the two schemes yield comparable performance, we discuss the aforementioned properties (P1)-(P3) when the within-group access costs are not symmetrical.

Requestor node-based access cost

The access costs depend on the node that issues a request (requesting node) but not on the node that will serve it (serving node having the content stored at its cache); thus, $t_l < t_r^{ij} = t_r^i < t_s, \forall i, j \in \mathcal{N}$. The optimal altruistic scheme is still described by a quadratic minimization function that reduces to the ILP problem (2.2.4), only now
$$z_m^n = \begin{cases} R_m^n(t_r^n - t_l), & \text{for } 1 \leq n \leq N; \\ \sum_{j=1}^{N} R_m^j(t_r^j - t_s), & \text{for } n = N+1, \end{cases}$$

Under the self-aware cooperative strategy and as with fully symmetrical intra-group access costs, a node n may evict only objects m that are replicated elsewhere in the group in favor of lower preference objects m' *not stored* anywhere in the group, if $R_m^n \cdot t_r^n < R_{m'}^n \cdot t_s$. Therefore, the strategy again is node mistreatment-free.

Fully asymmetrical access costs

The access costs depend on both the requestor *and* the serving node, $t_l < t_r^{ij} < t_s, \forall i, j \in \mathcal{N}$. The cost minimization function no longer reduces to an ILP, at least not in a trivial manner. Under the self-aware cooperative strategy, nodes still may evict only objects m that are stored elsewhere within the group, say node k; only now they may replace them with lower preference objects *already stored* in the group, say node l since it may hold that $R_m^n \cdot t_r^{nk} < R_{m'}^n \cdot t_r^{nl}$. For the same reason, the placement decisions made by nodes succeeding n may give rise to mistreatment of some nodes.

2.3 Cooperative placement strategy under node churn

Another significant aspect of groups with autonomous nodes is "churn", that may occur due to "join" and "leave" events. To account for node churn, we consider a probability estimate of each node to be available (*i.e.*, , make its objects available to the other members in the group). This estimate is common knowledge to all the nodes in the group, and is also called the reliability of a node. Such probabilities can be obtained from a reputation measure, which can itself be derived in a distributed manner, by local interactions between nodes. (For a review of distributed algorithms for deriving reputation measures, readers may refer to [Avrachenkov *et al.*, 2007].)

Under node churn, considering the reliabilities of nodes (2.2.1) becomes:

$$\mathcal{C}_n(P) = \sum_{i \in P_n} R_i^n t_l + \sum_{\substack{i \notin P_n, \\ i \notin P_{-n}}} R_i^n t_s + \sum_{\substack{i \notin P_n, \\ i \in P_{-n}}} \left[R_i^n [t_r (1 - \prod_{\substack{k=1, \\ k \neq n, k: i \in P_k}}^{N} (1 - \pi_k)) + t_s \prod_{\substack{k=1, \\ k \neq n, k: i \in P_k}}^{N} (1 - \pi_k)] \right] . \quad (2.3.1)$$

We extend the *self-aware cooperative* strategy (also named *greedy churn-unaware* strategy here) and present a *churn-aware cooperative* strategy (or *greedy churn-aware* strategy). Under this strategy it is assumed that each node n has information about the reliability of all other nodes, expressed in terms of a distribution of the ON probabilities π_k, $k = 1, \ldots, N$, $k \neq n$. The vector $\pi \triangleq (\pi_1, \pi_2, \ldots, \pi_N)$ is assumed to be common knowledge to all nodes. This information may be derived and forwarded in a distributed manner, by local interactions between nodes. It can also be maintained at a central database that nodes inquire into prior to making their decision.[1] Each node uses this information in making placement decisions so as to minimize its expected access cost.

By following the same arguments as in Section 2.2 under the churn-unaware case, we can show again that a node may only evict an object that is present in other nodes in the group. However, a node may now insert an object that also exists in one or more of the other nodes in the group (with ON probability smaller than 1).

For an object e replicated at node n, we define the eviction loss (equal to the increase in the expected access cost if node n evicted the object) as:

$$L_{e,n} = R_e^n [(t_s - t_l) \prod_{\substack{k=1 \\ (k \neq n, k: e \in P_k)}}^{N} (1 - \pi_k) + (t_r - t_l)(1 - \prod_{\substack{k=1 \\ (k \neq n, k: e \in P_k)}}^{N} (1 - \pi_k))] \, , \quad (2.3.2)$$

The first term is the increase in access cost if the nodes that own the evicted object are not available and the second term is the increase if at least one of them is available.

For an object i not replicated at node n, we define the insertion gain (equal to the decrease in the expected access cost if node n inserted the object) as:

$$G_{i,n} = \begin{cases} R_i^n (t_s - t_l), & \text{if } i \notin P_{-n} \quad (2.3.3a) \\ \\ R_i^n [(t_s - t_l) \prod_{\substack{k=1 \\ (k \neq n, k: i \in P_k)}}^{N} (1 - \pi_k) + (t_r - t_l)(1 - \prod_{\substack{k=1 \\ (k \neq n, k: i \in P_k)}}^{N} (1 - \pi_k))], & \text{if } i \in P_{-n} \quad (2.3.3b) \end{cases}$$

[1] The cost for such an inquiry is assumed negligible compared to the cost of accessing an object and is not considered in the analysis that follows.

The first term is the decrease in access cost if the inserted object does not belong to the group and the second term is the decrease if the object exists in the group.

Notice that in case $i \in P_{-n}$, although $R_e^n > R_i^n$, the right-hand-side of (2.3.3b) may be greater than the right-hand-side of (2.3.2), which substantiates our previous claim that objects that exist in other nodes in the group may also be inserted.

Considering the eviction candidates ordered in increasing average eviction loss, and the insertion candidates ordered in decreasing insertion gain, each node n makes a maximum number of replacements m_n with positive average replacement gain, *i.e.*, $\Delta G_{(e_k,i_k)} > 0$, for all $k = 1, \dots, m_n$.

In contrast to the churn-unaware strategy, a node may benefit by playing again at a subsequent epoch in the game, since it can insert an object that is available somewhere in the group with low probability. We therefore also study the game in which all nodes apply the greedy churn-aware strategy repeatedly for many rounds, until stopping. The algorithm stops when no node can further improve their placements. As the next theorem shows, this occurs always.

Theorem 1. *The greedy churn-aware algorithm ends in a finite number of rounds, irrespective of the order of play in each round.*

Proof. At each step of the game, each player may evict an object with a certain availability (*i.e.*, probability at least one of the nodes of the group that has the object is ON), to insert another object with smaller availability. Thus, there will come a time when no further replacements would be possible, either because nodes would not have objects in common or because there would not be any gain by making replacements. □

In practice, the algorithm finishes very quickly. In all test cases considered here (see later in the numerical results) the algorithm finished in 1 to 7 rounds.

2.3.1 Individual Rationality

A rational node is incited to follow a placement strategy other than the greedy local (or selfish) one if its individual rationality constraint is met, *i.e.*, if its incurred access cost when all other nodes follow this strategy is smaller than its incurred cost when all nodes act in isolation. In this subsection, we examine in which cases this constraint can be satisfied for all nodes.

In an environment with churn, the greedy churn-unaware strategy can easily violate a participation constraint. We demonstrate this with the following example, which also compares the churn-unaware to the churn-aware strategy.

Example 1. *Consider two nodes, node* 1 *and node* 2*, (with capacities* $C_1 = 4$*,* $C_2 = 1$*) and 5 distinct objects* $\{1, 2, 3, 4, 5\}$*. Nodes* 1 *and* 2 *have corresponding request rates*[2] R^1 $=$ *{0.5, 0.4, 0.3, 0.2, 0.1} and* R^2 $=$ *{0.4, 0.3, 0.5, 0.2, 0.1}. Node* 1 *is a relatively reliable node with* $\pi_1 = 0.9$ *and node* 2 *has a variable probability between* 0 *and* 1 *for the purpose of the example. We assume that* $t_l = 1$*,* $t_r = 10$ *and* $t_s = 100$*.*

Suppose that node 1 *plays first in the game. Under the greedy local strategy, nodes* 1 *and* 2 *will have the placements* P_1 $=$ *{1, 2, 3, 4} and* P_2 $=$ *{3}, respectively. Under the churn-unaware strategy, if node* 1 *plays first, the resulting placements are* P_1 $=$ *{1, 2, 4, 5} and* P_2 $=$ *{3}. However, if node* 2 *is unreliable, this may cause the violation of the participation constraint of node* 1*. This is indeed shown in the results of Fig. 2.1, where for* $\pi_2 < 0.74$ *the access cost of node* 1 *is greater than that of the greedy local strategy. On the contrary, in the churn-aware strategy, for* $\pi_2 < 0.74$*,* P_1 $=$ *{1, 2, 3, 4} and* P_2 $=$ *{5}, resulting in a smaller cost for node* 1*. For* $\pi_2 \geq 0.74$ *we have the same placements we had under the churn-unaware strategy, and thus the two lines coincide. It is shown in Fig. 2.1 that the greedy churn-aware strategy always performs better than the greedy local one and hence does not violate the individual rationality constraint.*

◊

The fact – shown in the previous example – that the greedy churn-aware strategy (as opposed to its churn-unaware counterpart) performs at least as good as the greedy local, always holds for two nodes, as shown in the following theorem.

Theorem 2. *For games with two nodes and under node churn, the greedy churn-aware strategy is always individually rational for both nodes.*

[2]It is noted that request rates in the examples used here are not normalized.

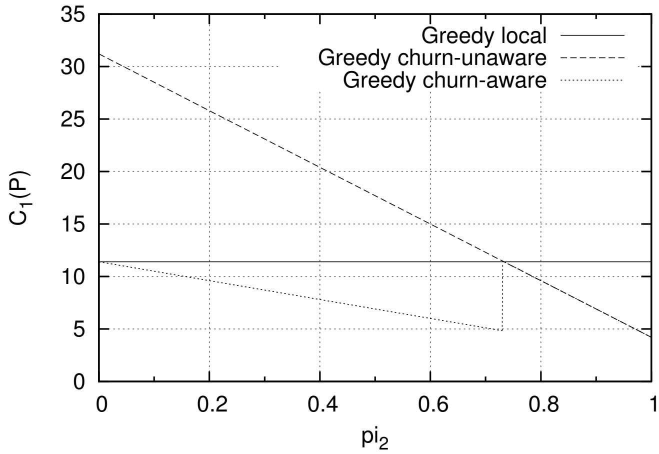


Figure 2.1: Violation of the participation constraints of node 1 for $\pi_2 < 0.74$ when using the churn-unaware strategy, in the game of Example 1.

Proof. The statement of the theorem is obvious for the last node in the game. Furthermore, since the second node only evicts an object that belongs to the first node, such action does not increase the access cost of the first node and hence the strategy is individually rational for the first node as well. □

For more than two nodes, one can easily construct examples with nodes having different request rates and probabilities of being available, in which the greedy churn-aware strategy is not individually rational for one or more of the nodes.

Example 2. *Consider three nodes, indexed* $1, 2, 3$, *and five distinct objects* $\{1, 2, 3, 4, 5\}$. *The request rate vectors are* $R^1 = \{0.5, 0.4, 0.3, 0.18, 0.1\}$, $R^2 = \{0.5, 0.3, 0.25, 0.24, 0.2\}$, *and* $R^3 = \{0.5, 0.4, 0.3, 0.2, 0.1\}$. *All nodes have the same capacity,* $C_1 = C_2 = C_3 = 3$. *The single-object access costs are* $t_\ell = 1$, $t_r = 5$, *and* $t_s = 10$. *Node 1 has the smallest reliability* $\pi_1 = 0.2$, *while* $\pi_2 = \pi_3 = 0.5$. *The order of play is according to each player's index (i.e., first node* 1, *then* 2, *and finally* 3*).*

When acting in isolation, the best strategy for all nodes is to locally replicate the first three objects (greedy local strategy). The corresponding costs, which are also the security levels for each node are: $\mathcal{C}_1 = 4$, $\mathcal{C}_2 = 5.45$, *and* $\mathcal{C}_3 = 4.2$.

When following the greedy churn-aware strategy, node 1 *inserts object* 4 *in place of* 3 *(it does* $3 \leftarrow 4$*), which yields itself an access cost reduction of* 0.045. *Nevertheless at its* *turn, node* 2 *will also do* $2 \leftarrow 4$*, as well as* $3 \leftarrow 5$*. Node* 3 *will not perform any change. The final placements are:* $P_1 = \{1, 2, 4\}$*,* $P_2 = \{1, 4, 5\}$*, and* $P_3 = \{1, 2, 3\}$*.*

It is anticipated that node 1 *will be disadvantaged by node* 2 *evicting objects* 2 *and* 3*, since it has relatively high request rates for these objects. Indeed this is the case, as shown by the final access costs of the nodes:* $\mathcal{C}_1' = 4.08$*,* $\mathcal{C}_2' = 4.915$*, and* $\mathcal{C}_3' = 3.35$*. The access costs of nodes* 1 *exceeds its security level, and hence the greedy churn-aware strategy is not individually rational for this player.*

$\Diamond$

Nevertheless, it is possible to show that in the homogeneous case (*i.e.*, when $R_i^j = R_i^{j'}$, for all $j \neq j'$, where $j, j' = 1, \dots, N$, and for all $i = 1, \dots, M$.), under additional conditions on the order of play and the eviction policy of the players, the greedy churn-aware strategy is individually rational for all players. The following definition embodies the additional condition on the eviction policy of the players.

Definition 1. The *cautious churn-aware strategy* is defined as the greedy churn-aware strategy, with the additional rule that a node can only evict an object if it is also evicted either by all, or by none of the previous nodes.

The following proposition can be proved for such a strategy.

Proposition 2. *In the homogeneous case, the cautious churn-aware strategy is individually rational for all nodes, when less reliable nodes play first and the game lasts one round.*

Proof. Consider the set of nodes $\mathcal{N} = \{1, 2, \dots, N\}$ and assume that the order of play is as dictated by the number of each player, which also corresponds to their reliability order; *i.e.*, $\pi_j \leq \pi_{j'}$ for $j < j'$, $\forall j, j' \in \mathcal{N}$. Initially, all nodes have the same lists of objects to be placed in their memory. Assume that each node starts making replacements in its list, as dictated by the churn-aware strategy. First notice from (2.3.2) that, because $\pi_j \leq \pi_{j'}$, the cost incurred to node j by evicting an object is smaller than the cost to node j', $j' > j$, by evicting the same object. Furthermore, the gain of node j', when inserting the same object is obviously smaller than that of node j, when $j' > j$. Suppose that node j makes m_j replacements when playing. It then follows that $m_1 \geq m_2 \geq \cdots \geq m_N$.

Consider any subsequence of ℓ nodes, indexed without loss of generality as $1, 2, \ldots, \ell$, where $\pi_1 \leq \pi_2 \leq \ldots \leq \pi_\ell$. Then each node in this subsequence makes at least m_l replacements, and subsequent nodes have decreasing gains when making the kth replacement, $k = 1, \ldots, m_l$. Denote the objects that node j removed by $e_1, e_2, \ldots, e_{m_j}$, indexed in increasing eviction cost. For its kth replacement, the replacement gain of node j is $G_{i_k,j} - L_{e_k,j}$. In the rest of the proof, we consider how the access cost of this node is increased by the moves that subsequent nodes make at their kth replacement step (if that exits), and show that the increase always remains smaller then $G_{i_k,j} - L_{e_k,j}$. Then, since node j makes a higher number of replacements than its subsequent nodes ($m_j \geq m_{j+1} \geq \cdots \geq m_N$), the total gain of node j by making its m_j moves can never become smaller than zero. Therefore node j incurs a smaller access cost under the final placement of all nodes compared to its greedy local placement.

The access cost of node j will be increased if subsequent nodes at their kth replacement remove one or more of the objects $e_1, e_2, \ldots, e_{m_j}$ that node j removed. (Its access cost is not affected if subsequent nodes insert the same objects, and is decreased if they insert some different objects.) However, we show next that in the cautious churn-aware strategy it always becomes lower than its cost before making this replacement.

We generally denote by $L^{(j)}_{e_k,j+n}$ the average increase in access cost incurred to node j due to node $j+n$ evicting object e_k. If node $j+n$ is the first node that evicts object e_k after j and if $L_{e_k,j+n}$ is the eviction cost of node $j+n$, it holds that $L_{e_k,j+n} = L_{e_k,j} + L^{(j)}_{e_k,j+n}$.

Let n_k be the number of nodes that make a kth replacement after node j has played, and consider the subsequence $j+1, j+2, \ldots, j+n_k$ of these nodes. The object evicted by node $j+1, j+2, \ldots, j+n_k$ at the kth replacement is denoted by $e_{k_{(j+1)}}, e_{k_{(j+2)}}, \ldots, e_{k_{(j+n_k)}}$, respectively.

Without loss of generality, we assume that at their kth replacement, *all* nodes in this subsequence evict objects evicted by node j (otherwise their induced cost to j is zero and we need not include these nodes). Due to decreasing gains of subsequent nodes for making the kth replacement, the objects evicted by a node $j+\ell$ are a subset of the objects evicted by j and the index $k_{(j+\ell)} \geq k\ \forall \ell = 1, \ldots, n_k$. We now use the fact that the objects evicted by node $j+2$ are also evicted by node $j+1$, for each pair ($j+2$,$j+1$) in the subsequence.

The replacement gain of node j after these evictions becomes at least (since it may be increased due to different inserted objects by other nodes)

$$G_{i_k,j} - L_{e_k,j} - L^{(j)}_{e_{k_{(j+1)}},j+1} - L^{(j)}_{e_{k_{(j+2)}},j+2} - \cdots - L^{(j)}_{e_{k_{(j+n_k)}},j+n_k} \tag{2.3.4a}$$

$$\geq G_{i_k,j} - L_{e_{k_{(j+1)}},j} - L^{(j)}_{e_{k_{(j+1)}},j+1} - L^{(j)}_{e_{k_{(j+2)}},j+2} - \cdots - L^{(j)}_{e_{k_{(j+n_k)}},j+n_k} \tag{2.3.4b}$$

$$= G_{i_k,j} - L_{e_{k_{(j+1)}},j+1} - L^{(j+1)}_{e_{k_{(j+2)}},j+2} - \cdots - L^{(j)}_{e_{k_{(j+n_k)}},j+n_k} \tag{2.3.4c}$$

$$\geq \cdots \geq G_{i_k,j} - L_{e_{k_{(j+n_k)}},j+n_k} > 0\ . \tag{2.3.4d}$$

(2.3.4c) follows from (2.3.4b) because it holds that $L_{e_{k_{(j+1)}},j+1} = L_{e_{k_{(j+1)}},j} + L^{(j)}_{e_{k_{(j+1)}},j+1}$ and $L^{(j+1)}_{e_{k_{(j+2)}},j+2} = L^{(j)}_{e_{k_{(j+2)}},j+2}$ (since both j, $j+1$ have evicted object $e_{k_{(j+2)}}$, the same cost is induced to both by $j+2$ evicting it.) Finally, the last implication holds because $L_{e_{k_{(j+n_k)}},j+n_k} < G_{i_{k_{(j+n_k)}},j+n_k} < G_{i_k,j}$. ($i_{k_{(j+n_k)}}$ denotes the object inserted by node $j + n_k$.) □

Remark 1. The additional condition on the eviction policy – imposed by the cautious churn-aware strategy – that an object that is allowed to be evicted by the current node is not evicted by any of the previous nodes, is intuitively anticipated. On the other hand, the fact that the greedy churn-aware strategy is still individually rational for all nodes when a node evicts an object evicted by all previous nodes, is rather surprising, since we might expect that in this case a node may incur a significant loss.

2.3.2 Potential Gain of a Node by Playing Again

A useful metric in our model is the potential gain of a node by playing again after a number of steps in the game. This gain is defined as a node's reduction in mean access cost if it were to play again after a number of steps in the game. Intuitively, a strategy that is close to being stable yields small such gains to all nodes.

A node j, $j = 1, \ldots, N-1$ may obtain a benefit by playing again if one or more subsequent nodes evict an object e evicted by j – in which case j may benefit by reinserting e –, or one or more subsequent nodes insert an object owned or also inserted by j, in which case j may benefit by evicting it and inserting another object.

Intuitively, if more reliable nodes play first, potential gains of nodes in next rounds are expected to be small, since the first nodes, who are more likely to be further away from an optimal placement, refrain from making changes when other nodes are less reliable.

In the homogeneous case, we can state the following proposition.

Proposition 3. *Consider a set of N nodes $\{1, 2, \ldots, N\}$, where $R^1 = R^2 = \cdots = R^N$, and more reliable nodes play first. Suppose that at a subsequent epoch after node j has played, $j = 1, \ldots, N-1$, a number of players evict an object e that was also evicted by node j and another number of players insert an object i also inserted by j. Say the last node that evicted object e was $j + n_1$, and the last node that inserted object i was $j + n_2$. If $n_1 = n_2$ $\forall$ (e, i), then at this epoch $L_{i,j} > G_{e,j}$ and node j does not obtain a benefit by playing again.*

Proof. We consider that players are numbered according to their order of play and $\pi_1 \geq \pi_2 \geq \ldots \geq \pi_N$ holds. The insertion gain of node j when reinserting object e equals the eviction cost of the last node after j that evicted the object. So $G_{e,j} = L_{e,j+n_1}$. If the last node that inserted object i was $j + n_2$, since $\pi_j \geq \pi_{j+n_2}$, we also have that $L_{i,j} \geq G_{i,j+n_2}$. If $n_1 = n_2$, then since we necessarily have $G_{i,j+n_1} > L_{e,j+n_1}$ it follows that $L_{i,j} > G_{e,j}$. □

We also examine this issue in numerical examples shown in the next section.

2.4 Implementation cost of placement strategies

In the following we calculate the implementation cost for the greedy local and the cooperative replication strategies, without including the cost for calculating reliability estimates. The implementation cost includes two components: the cost of *deciding* which ones to store locally and the cost of *retrieving* them and *storing* them in local memory. Throughout the analysis that follows, we assume for simplicity that all nodes have the same storage capacity C; extending the analysis to scenarios with nodes having non-uniform storage capacity is straightforward. Moreover, we assume that all communication links have sufficient capacity, and that nodes' requests for objects cannot be rejected because of congestion in other nodes or the remote server.

Under the greedy local strategy each node locally downloads its C most preferred objects. For each node, the average cost of placing the M objects in decreasing order of preference is $O(MlogM)$, with a comparison sort algorithm such as Heap sort or Merge sort [Cormen *et al.*, 2001]. Since we have N nodes, the total cost of the sorting operation is $NO(MlogM) = O(NMlogM)$. Then each node has to retrieve the first C objects from the distant source. The total cost of this operation is NCt_s. Therefore the total implementation cost of the selfish strategy is $O(NMlogM) + NCt_s = O(NMlogM)$ since Ct_s is constant and $C << M$.

The cost of a cooperative replication strategy is the cost of the greedy local *plus* the cost of object replacements the nodes make after viewing the placements of the other group nodes. Each node will do up to C replacements. Note that the objects can already be sorted in order of increasing eviction cost (in the local memory of each node) and decreasing insertion gain (in the central server), as part of the previous step of ranking objects in decreasing order of preference; hence, extra sorting costs are saved and the total cost $O(NMlogM) + NCts + O(NC) = O(NMlogM)$ remains comparable to the greedy local strategy. However, we need to also consider the additional cost of the process through which each node is informed about the placements of the other nodes. Generally, this process could run in two ways:

- In a *centralized* manner: each node uploads its placement in a central database. Then, each node downloads from the database the placements of other nodes. We have NC upload operations and $N(N-1)C$ download operations, so assuming a fixed cost for each operation the total cost is $O(N^2)$. Therefore, in the centralized case the total cost is $O(NMlogM + N^2)$, if we let aside the cost of setting up and maintaining the central database.

- In a *distributed* manner: each node transmits its placement information individually to other nodes. In the worst case, each node would transmit the information to all other nodes, which requires $N(N-1)C$ operations. Therefore, in the distributed case the total cost is again $O(NMlogM + N^2)$.

We see that the cost of a cooperative strategy remains polynomial in N, M. It is noted here that an optimal placement of objects – so that the sum of access costs of all nodes is minimized – requires a complete knowledge of the group's characteristics (demand patterns of all nodes in the group) and a solution to the associated optimization problem. The latter is a 0-1 programming problem, and is generally classified as NP-hard [Schrijver, 1998]. When all nodes are available with probability 1, it can be transformed into a linear problem (see Proposition 1), but in the general case with arbitrary ON probabilities it is a problem of degree N, which is extremely difficult to solve even with numerical methods.

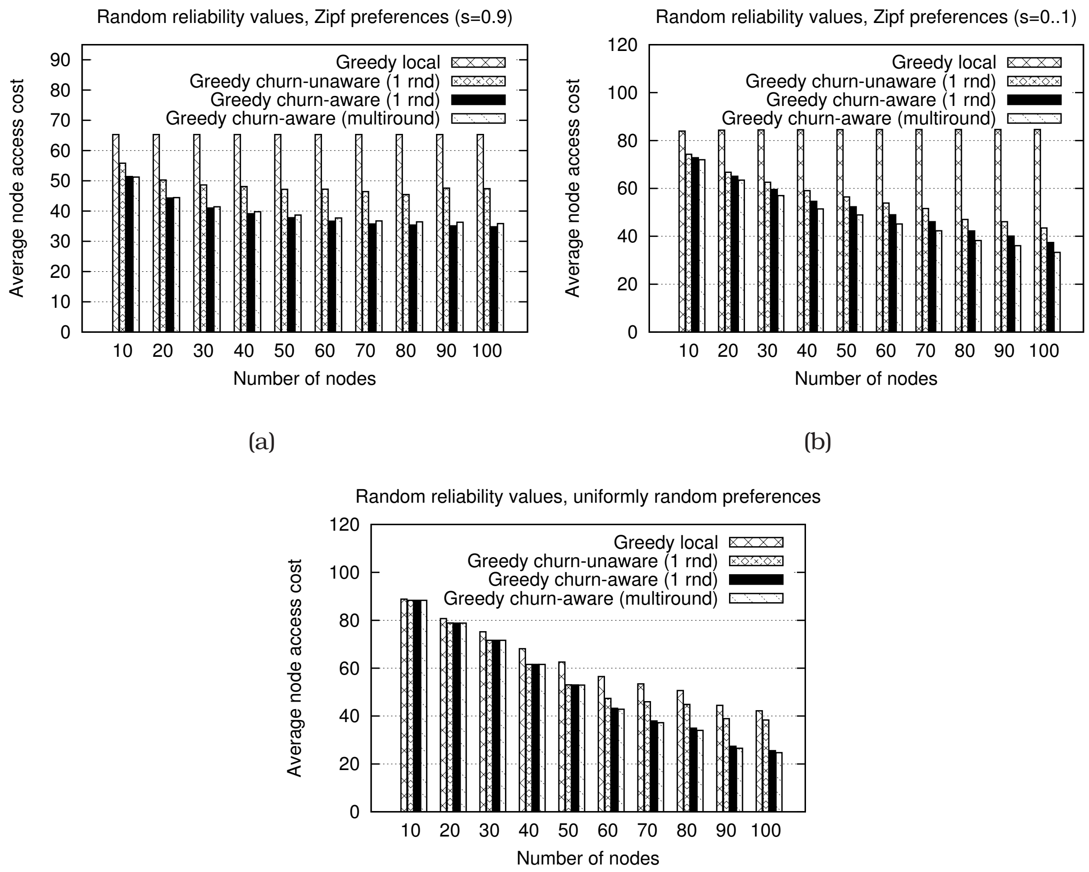
2.5 Numerical evaluation of content replication strategies under node churn

Here we present some numerical examples to show how the different object replacement strategies perform in a replication group under node churn. We are interested in analyzing cases where nodes have similar preferences for objects, so that mutual benefits emerge by cooperation. Request rates for each node are drawn from a Zipf distribution with exponent s. The probability of the object with rank k, $k = 1, ..., M$, is

$$f(k; s, M) = \frac{1/k^s}{\sum_{m=1}^{M} 1/m^s} \ .$$

We first consider the simplest case where nodes have the same request rates. Fittings based on real traces in [Breslau *et al.*, 1999] have shown the value of the exponent s of the Zipf distribution to lie between $0.8 - 0.9$. Here we take $s = 0.9$. The total number of nodes in the replication group is varied from 10 to 100. Each of the N nodes has a capacity to hold 10 objects in its local storage, and there exists a total of $M = 500$ objects. The reliability of each node is chosen uniformly in $[0, 1]$ and experiments are repeated for 100 random instances. We set $t_l = 1$, $t_r = 10$ and $t_s = 100$ cost units.

In Fig. 2.2 we show the average access cost over all nodes and all instances. As the number of nodes increases, more objects are available in the replication group. Thus, the cooperative (churn-aware and churn-unaware) replication strategies produce higher gains over the greedy local strategy. However, there is no further gain after some number N, as anticipated. As shown in Fig. 2.2(a) (with nodes having the same Zipf distribution with $s = 0.9$), this limit is close to M/C. For values of $N > M/C$ there is enough storage capacity to replicate all server objects in the group, and there is almost no further gain by adding more nodes. Furthermore, the fact that nodes rely more on other group nodes to retrieve content implies that their reliability plays a greater role. Hence, although there is a reduction in the access cost using both strategies, this is progressively more significant for the churn-aware strategy.



(a) (b)

(c)

Figure 2.2: Average node access cost for different total number of nodes.

We also took results where nodes have different values of s. We examined the case where the interests in the objects are drawn from Zipf distributions with different exponent s for each node. We let $s = 0$ for the first node (uniform interest distribution) and $s = p(n-1)$ for node n, $n \in [2, N]$, where $p = 1/(N-1)$ is an increment parameter. Results are shown in Fig. 2.2(b). Compared to the previous case of homogeneous interest distributions, in this case most nodes have smaller s parameter values and thus their interests are now more uniformly spread over the set of objects (the first-ranked objects). Thus, for small values of N we observe higher access costs (smaller cooperation gains) compared to the results in Fig. 2.2(a). Further, the churn-aware strategy does not give significantly smaller cost compared to the churn-unaware one. Only as the number of nodes increases, we arrive at gains that are comparable to the case of homogeneous request rates. Indeed, when nodes in the group have more uniform preferences, high cooperation gains exist for large numbers of nodes as is illustrated more clearly by the next set of experiments.

Here we examine the gains under node churn when nodes have uniformly random preferences. For this case, request rates for each node are drawn uniformly in $[0, 1]$, and experiments are averaged over 100 random instances. The results shown in Fig. 2.2(c) again attest that there are significant gains that can be achieved with the churn-aware strategy. Under uniformly random preferences, the similarity of interests between nodes is relatively smaller compared to the previous case of Zipf distributions. However when there is a large number of nodes, the probability that similar content exists in some other active node increases. This is why the gains by content retrieval from nodes in the group increase as the total number of nodes increases, and gains may exist even when nodes store content solely based on their own interests. The relative difference between churn-aware and churn-unaware or greedy strategies also increases as the number of nodes increases. Overall, we conclude that large groups favor cooperation to a larger extent, and this is aggravated when nodes have more uniform preferences.

We next want to study the individual node access cost, setting a smaller number $N = 10$ nodes and a total of $M = 50$ objects. We mainly examine the case where nodes play according to a random order. To assess the impact of the order of play, we also examine two additional orderings based on the reliability of nodes: a) Least Reliable First (LRF) where nodes play in increasing order of their ON probabilities, b) More Reliable First (MRF) where nodes play in decreasing order of ON probabilities.[3] The cost parameters are the same, and we consider that all nodes have the same Zipf distribution with $s = 0.9$. The (mean) access costs of all nodes under the different orderings are shown in Fig. 2.3. The vector $\pi = [\pi_1, \pi_2, \ldots, \pi_N]$ of ON probabilities for each examined order of play is shown in the title of each sub-figure. It turns out that both the churn-aware and unaware strategies outperform the greedy local, producing smaller costs for all nodes (hence participation constraints are satisfied for all nodes).

[3]In practice, a central coordinating entity is required to enforce a specific ordering, that authorizes each node to access the lists of placements of other nodes only at the specific order. Therefore, these test cases have mostly a theoretical interest. In the examples, we also consider that the same order of play is maintained in each round.

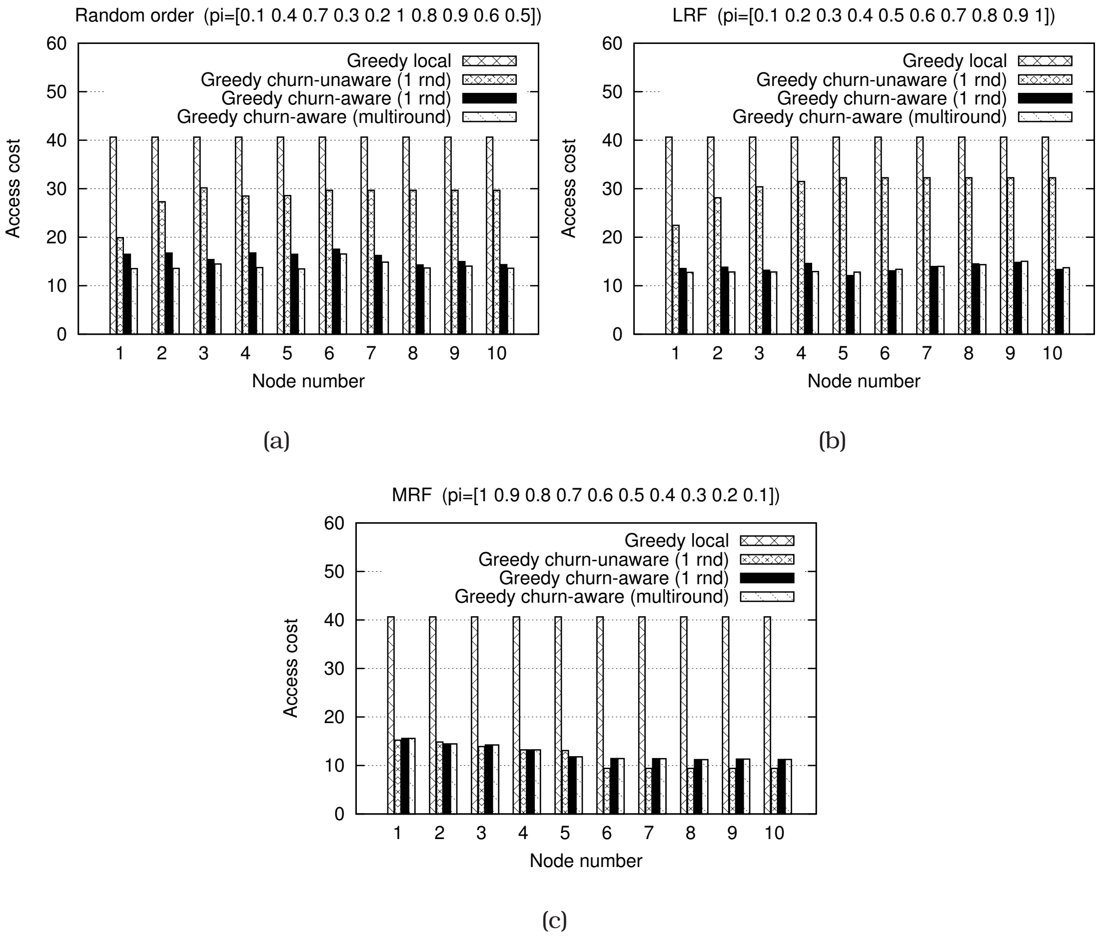


Figure 2.3: Access cost for different placement strategies with different node orderings.

In assessing the impact of the order of play, we remark that the greedy churn-aware strategy clearly outperforms the churn-unaware under the LRF ordering (Fig. 2.3(b)). The reason is that under the LRF ordering, the churn-unaware strategy results in many high request rate objects being stored at less reliable nodes (more reliable nodes falsely trust high request rate objects to be accessible from the previous nodes, while it would be better to have them stored locally). Thus, we can say that the LRF ordering is unfair under this strategy because it leads more reliable nodes to erroneous placements. Such placements occur less frequently when the order is MRF, in which case the churn-aware strategy does not present a significant advantage over the churn-unaware one and their performances are close (Fig. 2.3(c)). Finally, since the churn-aware strategy yields a significant improvement over the churn-unaware, for both the random and LRF orders, we remark that fairness problems are mitigated under the churn-aware strategy.

An important observation shown in both Fig. 2.2, 2.3 is that there is no significant access cost reduction by repeating the churn-aware algorithm for multiple rounds. In Fig. 2.3, we notice that this yields extra benefit only to some nodes; depending on the order of play, some nodes may benefit from the extra rounds, which usually occurs at the expense of others. The potential gain of each node by playing again after one round under the churn-aware algorithm and the various orderings is shown in Table 2.1. Under the MRF ordering, the algorithm stops in the first round and thus all potential gains are zero[4]. Under the other orderings, we remark that only the first few nodes attain a benefit by playing again, since they are more likely to be farther from an optimal placement. In any case, benefits are small when compared to the access cost values (see Fig. 2.3), and hence nodes have a small incentive for playing again.

Table 2.1: Potential gains of nodes by playing again after 1 round.

	Potential gain									
Node	1	2	3	4	5	6	7	8	9	10
LRF	1.52	1.61	0.83	1.93	0	0	0	0	0	0
MRF	0	0	0	0	0	0	0	0	0	0
Random order	2.81	2.73	0	2.94	2.69	0	0	0	0	0

We end this section by summarizing our findings. While the proposed algorithm we presented may not arrive at an equilibrium, we have shown that it possesses good properties: in the majority of cases, it decreases the access cost collectively for all nodes, compared to the greedy local or the churn-unaware strategy. Although individual rationality is not guaranteed, participation constraints (*i.e.*, not loosing by participating) are easier met compared to the other strategies. For the case of two nodes it is shown that the churn-aware strategy is always rational for both nodes. Furthermore, if nodes having the same request rates for objects play according to a least-reliable-first ordering, the modified version of the proposed strategy "cautious churn-aware strategy" is shown to be individually rational for all nodes. Finally, the greedy churn-aware strategy provides for a fairer treatment of nodes according to their reliability, while the churn-unaware strategy can lead to higher access costs to more reliable nodes.

[4]The analysis in Section 2.3.2 justifies this result to some extent.

Chapter 3

Impact of social similarity on content replication

Networks today can be highly personalized, in the sense that their structure and usage are shaped by the personal interests, or behavior in general, of the participating nodes. Nodes in such networks - referred to as social networks - are typically well connected, develop reciprocal trust relations, and share some attributes, such as content interests and locality. Groups of such nodes are called social groups [Scott, 2000]. Here the characteristics of the social group are exploited to address the dilemma: which strategy a node participating in a distributed replication group should follow.

3.1 Introduction

In the previous chapter we referred to the following three strategies (under no node churn):

- The *selfish* strategy requires no interaction with and guarantees no mistreatment by the other nodes; on the other hand, both the node itself and the group could benefit more by following another strategy.
- The *self-aware* cooperative strategy outperforms the selfish strategy both at the individual node and group levels, while it ensures no mistreatment of individual nodes; on the other hand, it does not maximize the group benefit, while it introduces

complexity that increases with the group size and may outweigh the benefits for individual nodes.

- The *optimally altruistic* strategy yields the maximum possible benefit for the entire group. On the other hand, it can mistreat certain nodes (with the risk of inciting them to leave the group) and requires heavier interaction with the other nodes of the group (increasing complexity); these interactions could be lighter under a centralized derivation of the optimal placement.

In view of the above it is evident that a node participating in a distributed replication group can face a dilemma as to which strategy to follow. To this end, we follow an innovative approach to the characterization of the nodes' similarity within a social group and introduce a group *tightness* metric, which explicitly accounts for the level of similarity of their content preferences. Our work highlights the impact of group tightness on the induced social and individual node benefits under the three aforementioned content placement strategies, which reflect general patterns of social behavior. On a more practical note, it draws important conclusions/guidelines regarding the kind of placement strategy a node should adopt for given levels of *tightness* in the social group.

This study has applications to social networks featuring interactions between computer devices with limited memory resources. These are typically encountered in mobile opportunistic networks that are additionally "socially aware", meaning that either the nodes or their human users are aware of the formation of social groups and the potential benefits from participation in such a group. The underlying assumption is that there are multiple groups, and we focus on the behavior regarding the exchange of information objects between nodes inside a single group. Studying content access patterns, especially between nodes in a social group is quite important to assess the viability of various networking paradigms, *e.g.*, the opportunistic wireless networking and some P2P systems.

3.2 Group tightness metric

Each one of the three placement strategies presented in Section 2.2 resolves differently the multiple tradeoff among: a) the performance of individual nodes and of the entire group; b) the possibility of individual nodes being mistreated and the respective (lack of) incentives for cooperation; c) the required communication overhead and computational complexity for each strategy realization. The obvious question then for a node-member of the social group is which strategy is the most "appropriate" to follow. In this section, we introduce a metric, which we call *tightness*, for the similarity of interests within the social group that can be of help in reaching a conclusion.

The definition of *tightness* draws on the symmetrized Kullback-Leibler (KL) divergence [Kullback, 1959], a well-known measure of divergence between two distributions. The Kullback-Leibler divergence of distribution Q from S is defined as:

$$D_{S,Q} = \sum_{m} S(m) log \frac{S(m)}{Q(m)}.$$

and its symmetrized counterpart is $D(S||Q) = D_{S,Q} + D_{Q,S}$.

The average divergence of nodes' preferences within the group can then be written as:

$$\hat{D}_R = \frac{\sum_{(i,j)} D(R^i||R^j)}{N(N-1)/2} \quad (3.2.1)$$

where the summation above is carried out over all $N(N-1)/2$ ordered node pairs (i, j). It is reminded that R^i denotes the preference distribution of node i over a set of interest classes. Finally, we define *tightness* T to be the inverse of $\hat{D}_R$:

$$T = \frac{1}{\hat{D}_R}. \quad (3.2.2)$$

We elaborate on computational aspects of the tightness metric in the following.

3.2.1 Computation of tightness metric

Brute-force computation of the tightness metric as (3.2.1) suggests, requires $W = 2MN(N-1)$ multiplications and $W - 1$ additions, noting that $alog\frac{a}{b} = aloga - alogb$. It is possible to reduce the number of elementary operations significantly if redundant operations are avoided.

There are two kinds of sums that repeatedly emerge in (3.2.1). The first one, s_c, is the sum of product terms $\sum\limits_{j \in \mathcal{M}} R^i_j log R^i_j$ over all M content objects, where R^i_j is the normalized preference of group node i, $i \in \mathcal{N}$, for content object j, $j \in \mathcal{M}$. There are N such sums, one for each group node. The second type of recurring terms are those summing the (normalized) preferences of all nodes for a given content object, $\sum_{j\in\mathcal{N}} R^i_j$. There are M such sums, as many as the content objects.

The computation of tightness can proceed by successively considering each node n, computing the respective sum $s_c(n)$ and updating the sums $s(j)$ based on the content preference distribution R^n. This computation requires $2MN$ multiplications and approximately $3MN$ additions.

Algorithm 1 Iterative computation of the tightness metric

Initialization

$for\ i = 1,2 \;\; s_c(i) \leftarrow \sum_{j\in\mathcal{M}} R^i_j log R^i_j$

$for\ j \in \mathcal{M} \;\; s(j) \leftarrow \;\; R^1_j + R^2_j$

Iterative part

for all $i \in 3..N$ **do**

$s_c(i) \leftarrow \sum_{j\in M} R^i_j log R^i_j$

for all $j \in \mathcal{M}$ **do** $s(j) \leftarrow \;\; s(j) + R^i_j$

$D \leftarrow (N-1)\sum_{i\in\mathcal{N}} s_c(i) - \sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{M}}(s(j) - R^i_j) log R^i_j$

$T \leftarrow \frac{N(N-1)}{2D}$

3.2.2 Why tightness as a metric?

In principle, various measures of distributional similarity could quantify the similarity of content preferences across the nodes of a group. For example the Spearman's rank correlation coefficient [Myers & Well, 2003], the Kolmogorov-Smirnov distance [Wang *et al.*, 2003], the proportional similarity index [Vegelius *et al.*, 1986], and the total variation distance [Denuit & Bellegem, 2001]. Compared to them, the proposed tightness metric has the following advantages:

Sensitivity to rank-preserving dissimilarity: Contrary to metrics such as Spearman's rank correlation coefficient, tightness can capture dissimilarity of interests among nodes that may rank the objects similarly, yet focus with different intensity on the top-k content objects (see rank-preserving dissimilarity case in Section 3.3).

Account of full preferences' profiles: Contrary to the the Kolmogorov-Smirnov (K-S) distance metric, which considers the supremum of the differences over all elements of a distribution, the KL divergence accounts for deviations across the whole distribution. Thus, the proposed tightness metric captures more accurately the overall distributional (dis)similarity.

Broader range of values: In contrast with the proportional similarity and total variation distance metrics [Vegelius *et al.*, 1986], which yield values in $[0, 1]$, tightness values vary in $(0, +\infty)$. Therefore it can resolve easier finer levels of distributional divergence. In the following chapter, we will show how this property of the tightness metric can benefit a different task, that of community detection, by modulating its resolution.

Finally, we should note that when some element values of one of the distributions are zero while the corresponding elements of the other distribution are not (*i.e.*, the request rate of a node for an object is zero), the KL distance value approaches infinity. In order to avoid such problems, smoothing methods such as interpolation and backing-off schemes can be used for providing reliable probability estimates. These methods have been studied in statistical language modelling in order to estimate the distribution of natural language elements as accurately as possible. In our case, non-zero request rates for objects can be discounted with different discounting methods (see [Mori, 1997]), whereas all other non-requested objects can be given a minimal ϵ probability. Here, we will consider that all nodes have probability mass (*i.e.*, positive request rate) for all content objects, so that we do not need to apply any smoothing method.

3.3 Dissimilarity patterns for performance evaluation

Tightness expresses the similarity of preferences among the nodes of the social group and is always greater than or equal to zero. In fact, $T \to \infty$ when the group nodes have very similar preferences and $T \to 0$ when they have very diverse preferences. As T is an average metric over all node pairs of a social group, it is clear that any given value of T may arise under different combinations of node level content preference distributions.

The content preferences of nodes are modeled by Zipf distributions with variable shape parameter s, *i.e.*, the normalized interest of node n for its k^{th} most interesting object is $(1/k)^s / \sum_{m=1}^{M} 1/m^s$. Zipf distributions combine modelling simplicity with flexibility in that proper manipulation of their shape parameter s, gives rise to a wide set of distributions ranging from uniform ($s = 0$) to highly skewed ones with power-law characteristics ($s >> 0$).

In order to draw more insightful conclusions in the current study, we distinguish between the following two broad patterns of dissimilarity in the preference distributions.

3.3.1 Rank-preserving dissimilarity

The rank of the objects remains the same for all group member nodes, *i.e.*, the m^{th} most popular object for all nodes is the same, $m \in [1, M]$. However, the preference distributions become more concentrated towards the highly-ranked objects as the shape parameter s increases.

More specifically, the interests R_m^n in the object m, $m \in [1, M]$, are drawn from Zipf distributions with different exponent s for each node. We let $s = 0$ for the first node (uniform interest distribution) and $s = p(n-1)$ for node n, $n \in [2, N]$, where $p \in \mathbf{R}$ is the increment parameter. As shown in Table 3.1(a), under the (object-)rank-reserving dissimilarity scenario, *tightness* is a monotonically decreasing function of p. As p increases, the content preference distributions of nodes diverge more strongly, resulting in higher pairwise KL divergence values between any two node distributions.

3.3.2 Shape-preserving dissimilarity

The preference distributions are identical in shape for all nodes, yet the ranking of a given object differs from node to node, *i.e.*, the $m^{th}, 1 \leq m \leq M$, most popular object for each node is different. The dissimilarity of nodes can be more dramatic in this case and lower tightness values are expected on average.

Contrary to the rank-preserving dissimilarity scenario, the request rates R_m^n are drawn from a Zipf distribution with the same exponent s for all nodes. The object preference rank for first node is $[1, 2..., M]$ and is shifted to the right by $k(n-1)$ positions for

Table 3.1: Example tightness values when $M = 50$ and $N = 5$.

(a) rank-preserving (b) shape-preserving

p	Tightness (T)	k	Tightness (T)
0.0	∞	0	∞
0.2	2.0861	1	0.3688
0.4	0.4614	2	0.2674
0.6	0.2398	3	0.2294
0.8	0.1697	4	0.2089
1.0	0.1362	5	0.1962

node n, $n = 1, \ldots, N$, where $k \in [0, M-1]$ is the *shift parameter*. For example, the most preferable object for node n is the one with index $u = mod(k(n-1), M)$ and its object preference rank is $[u, u+1, ..., M, 1, ...u-1]$. Table 3.1(b) lists the values of *tightness* for various values of k. Notably, *tightness* is a monotonically decreasing function of the shift parameter k for k values satisfying $(N-1)k + C < M$. As k increases in this interval, the divergence in the content preferences between any two nodes increases. Generally, the numerical values of *tightness* are smaller than in the rank-preserving dissimilarity scenario. Even for small values of k (*e.g.*, k=1), the divergence is high enough to reduce tightness below 0.37.

These two broad dissimilarity patterns let us control systematically yet simply the extent of similarity in the preferences of group nodes. Since Zipf distributions capture the content preferences of a single group member, we can differentiate the distributions (hence, the content preferences) by adjusting two properties, their ranking of objects and skewness parameter s. In other words, the parameters s and k serve as tuning knobs with predictable effect. The shape parameter s gives rise to a wide set of distributions ranging from uniform (s=0) to highly skewed ones with power-law characteristics ($s \gg 0$). The permutation (or shift) parameter k shuffles the ranking of preferences across objects. Tuning the two parameters, we can synthesize a very broad range of possible dissimilarity patterns across the social group. We exercise this flexibility in the analysis that follows in Section 3.4 and the numerical evaluation in Section 3.5.

3.4 Analytical derivation of placement cost

The expected total access cost under the altruistic strategy can be numerically computed, at least for small values of M, N, after solving the ILP problem (2.2.4) and deriving the optimal placements that minimize this cost. The expected total costs incurring from the altruistic and selfish strategies constitute lower and upper bounds, respectively, for the expected total cost incurring from the self-aware cooperative strategy. Herein, we derive analytical expressions for the expected per-node access cost under the selfish and self-aware cooperative content placement strategies for the two content preference-dissimilarity patterns described in Section 3.3.

In our analysis, the group nodes are indexed in order of increasing Zipf distribution exponent s (for the rank-preserving dissimilarity) and distribution-shift k (for the shape-preserving dissimilarity). For content items, on the other hand, two kinds of item indexing become relevant: the "global" one, enumerating all objects in decreasing preference order of node 1; and, the local node-specific ones, which index objects in decreasing preference order of the respective nodes. The two types of indexing coincide under rank-preserving dissimilarity; whereas there are N different local indexings, one per node, under shape-preserving dissimilarity.

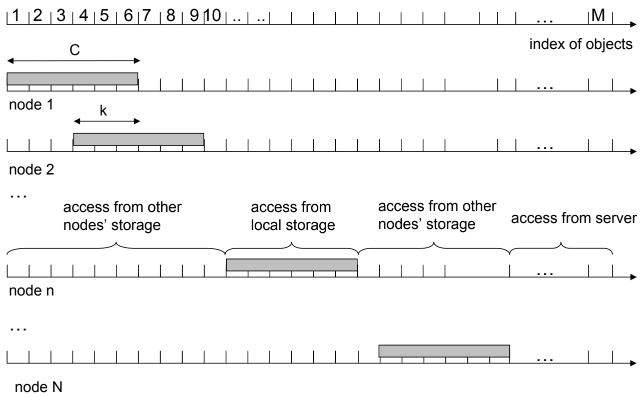
3.4.1 Selfish placements

When the nodes behave selfishly, it is possible to analytically compute the amount of content they access from the three levels of data storage, *i.e.*, locally C_l, remotely from the caches of the group member nodes C_r, and externally from server(s) or nodes in other social groups C_s, as well as the resulting access costs. We treat separately the two generic dissimilarity patterns described in Section 3.3.

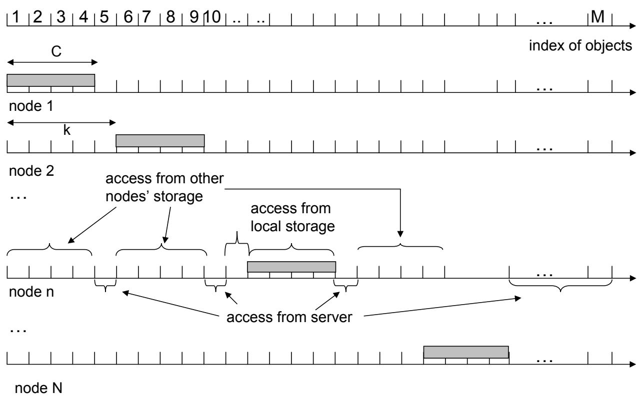
Rank-preserving dissimilarity, $p \neq 0$: All nodes store locally the *same* C content items that commonly rank top at their preferences and access the remaining $M - C$ items from external sources. What changes with p is the preference amount that is concentrated in the C items each node stores, which is controlled by the exponent $s_n = p \cdot (n-1)$ in the Zipf content preference distribution. Therefore, $C_l = C$, $C_r = 0$, and $C_s = M - C$ for all nodes and the overall access cost for node n, is given by

$$\mathcal{C}_n^S(\mathcal{P}) = \frac{\sum_{j=1}^{C} j^{-s_n}}{\sum_{j=1}^{M} j^{-s_n}} t_l + \frac{\sum_{j=C+1}^{M} j^{-s_n}}{\sum_{j=1}^{M} j^{-s_n}} t_s \tag{3.4.1.1}$$
$$= \frac{\sum_{j=1}^{C} j^{-p\cdot(n-1)}}{\sum_{j=1}^{M} j^{-p\cdot(n-1)}} t_l + \frac{\sum_{j=C+1}^{M} j^{-p\cdot(n-1)}}{\sum_{j=1}^{M} j^{-p\cdot(n-1)}} t_s.$$

Shape-preserving dissimilarity, $k \neq 0$: Now, the C items each node stores locally are, generally, different than those other nodes store. Assuming that the number of content items exceeds the cumulative group storage capacity, $M \gg N \cdot C$, we can distinguish two possibilities:



(a) $k < C$, partial overlapping in local storage



(b) $k \geq C$, each nodes stores different items

Figure 3.1: Objects stored at different group nodes under shape-preserving dissimilarity.

a) $k < C$: There is partial overlapping in the preferences of two (or more) nodes (Fig. 3.1(a)). Each node accesses $C_r = (N-1)k$ items from the storage of the other group nodes and $C_s = M - C - (N-1)k$ items from the server(s). As the shift parameter k increases, the nodes of the social group cumulatively store and can access from each others' storage more content. Yet, nodes with higher index n have to access more items ranking higher at their preferences from external sources since they are not stored by any other group member. Worst of all, the node N has to fetch the content items ranking at positions $[(C+1),(M-(N-1)k)]$ in its own preference distribution at cost t_s, whereas he can get access to content objects in positions $[M-(N-1)k+1, M]$ through the storage of the other group nodes.

The content access cost for node n can be written

$$\begin{aligned}\mathcal{C}_n^S(\mathcal{P}) = {} & \frac{\sum_{j=1}^{C} j^{-s}}{\sum_{j=1}^{M} j^{-s}} t_l + \frac{\sum_{j=C+(N-n)k+1}^{M-(n-1)k} j^{-s}}{\sum_{j=1}^{M} j^{-s}} t_s \\ & + \left[\frac{\sum_{j=C+1}^{C+(N-n)k} j^{-s}}{\sum_{j=1}^{M} j^{-s}} + \frac{\sum_{j=M-(n-1)k+1}^{M} j^{-s}}{\sum_{j=1}^{M} j^{-s}} \right] t_r. \end{aligned} \tag{3.4.1.2}$$

b) $k > C$: There is no overlapping in the items each node stores locally in its cache (Fig. 3.1(b)) so that $N \cdot C$ different content items are stored within the group. The rest of the $C_s = M - N \cdot C$ objects have to be fetched from external sources. As long as $N \cdot k + C \leq M$, the access cost increases with higher k and node index n values. The resulting content access cost for node n is given by (3.4.1.3).

$$\begin{aligned}\mathcal{C}_n^S(\mathcal{P}) = {} & \frac{\sum_{j=1}^{C} j^{-s}}{\sum_{j=1}^{M} j^{-s}} t_l + \left[\sum_{i=1}^{N-n} \frac{\sum_{j=i(C+k)+1}^{i(C+k)+C} i^{-s}}{\sum_{j=1}^{M} j^{-s}} + \sum_{i=1}^{n-1} \frac{\sum_{j=M-ik-(i-1)C+1}^{M-i(k+C)+1} j^{-s}}{\sum_{j=1}^{M} j^{-s}} \right] t_r \\ & + \left[\sum_{i=1}^{N-n} \frac{\sum_{j=iC+(i-1)k+1}^{i(C+k)} j^{-s}}{\sum_{j=1}^{M} j^{-s}} + \sum_{i=1}^{n-1} \frac{\sum_{j=M-(i-1)(k+C)}^{M-ik-(i-1)C} j^{-s}}{\sum_{j=1}^{M} j^{-s}} + \frac{\sum_{(N-n+1)(C+k)+1}^{M-(n-1)(C+k)} j^{-s}}{\sum_{j=1}^{M} j^{-s}} \right] t_s. \end{aligned} \tag{3.4.1.3}$$

Identical uniform content preferences: This represents an extreme case of zero dissimilarity in the preferences of group nodes. It results from the general rank-preserving dissimilarity scenarios when $p = 0$, and from the shape-preserving dissimilarity scenarios, when $s = 0$.

Contrary to the general dissimilarity scenarios, the distribution of content objects at the three storage levels and the corresponding access cost are no longer deterministic.

$$Pr(X \leq C + z) = \sum_{i_2=0}^{min(z,C)} h(i_2; M, C, M - C) \sum_{i_3=0}^{min(z-i_2,C)} h(i_3; M, C, M - C - i_2)... \quad (3.4.1.4)$$
$$\sum_{i_n=0}^{min(z-1-\sum_{j=2}^{n-1} i_j, C)} h(i_n; M, C, M - C - \sum_{j=2}^{n-1} i_j)... \sum_{i_N=0}^{min(z-\sum_{j=2}^{N-1} i_j, C)} h(i_N; M, C, M - C - \sum_{j=2}^{N-1} i_j).$$

The C objects stored locally at each node are randomly chosen out of the full set of M objects so that the total number of *different* objects collectively stored at the caches of the N group nodes is a random variable X, $C \leq X \leq M$. As earlier, the cost for each node consists of two components: a) the cost $(X - C) \cdot t_r/M$ of accessing the $X - C$ objects stored at the caches of the other N-1 group nodes; b) plus the cost $(M - X) \cdot t_s/M$ of accessing the remaining $M - X$ content objects from the server.

The probability distribution of the random variable X is given by (3.4.1.4), where $h(x; L, r, T)$ is the hypergeometric probability distribution. It expresses the probability that a hypergeometric experiment, *i.e.*, the random selection without replacement of an r-size sample, where the population consists of L items and T of them are classified as successes, results in exactly x successes.

Assume, without breach of generality, that the N nodes choose objects to store sequentially. The first node may choose any C items. Then each of the N-1 remaining nodes carries out a hypergeometric experiment, whose sample size C and total population of objects M remain the same, whereas the number of items that can be considered successes in the n^{th} draw declines according to the outcome of all previous n-1 experiments. Likewise, the selection of the first node could be viewed as a hypergeometric experiment, where the number of success items equals the total content item population M.

The expected value of X is

$$E[X] = \sum_{z=0}^{M-C} (C + z) \cdot [Pr(X \leq C + z) - Pr(X \leq C + z - 1)],$$

where $Pr(X \leq C - 1) = 0$, and the *expected* per-node content access cost becomes

$$\mathcal{C}_n^S(\mathcal{P}) = \frac{1}{M}[C \cdot t_l + (E[X] - C) \cdot t_r + (M - E[X]) \cdot t_s]. \quad (3.4.1.5)$$

3.4.2 Self-aware cooperative placements

The selfish placements analyzed above are the starting point for the self-aware cooperative placements. In the second step of the strategy, nodes take turn in adjusting the selfish placements of the first step evicting objects that are replicated elsewhere in the group and inserting new ones, not yet stored anywhere in the group, inline with the properties (P1)-(P3) listed in Section 2.2. In general, by the end of the second step, each node n has retained the first j_n objects of the initial selfish placement and inserted $C - j_n$ new ones, which are not replicated anywhere else in the group, according to property (P2).

Rank-preserving dissimilarity

By the end of the first step, the selfish strategy gives rise to full replication of the same C objects in the group so that candidates for insertion are objects with global indices in $[C+1, M]$. Let s_i be the exponent in the Zipf preference distribution for node i with, $s_i < s_j$ for $i < j$, and consider the first node just before executing the second step (placement adjustment). Depending on its distribution skewness, node 1 will retain j_1 objects in its cache, $1 \leq j_1 \leq C$, and remove the rest, where

$$\begin{aligned} j_1 &= min(max\{u : \frac{t_r - t_l}{u^{s_1}} > \frac{t_s - t_l}{(2C - u + 1)^{s_1}}\}, C) \\ &= min(\lfloor \frac{2C+1}{1 + (\frac{t_s - t_l}{t_r - t_l})^{1/s_1}} \rfloor, C) \end{aligned} \tag{3.4.2.1}$$

namely, the maximum item index of those stored locally in the first step, for which the benefit of retaining it in the local storage exceeds the insertion benefit from the next most preferred item among those not yet stored elsewhere in the group (Fig. 3.2).

Since nodes only insert non-represented objects in this step (P2), candidates for insertion by the second node will be the objects $[2C - j_1 + 1, 3C - j_1]$, and generalizing, by the n^{th} node, the objects $[nC - \sum_{l=1}^{n-1} j_l + 1, (n+1)C - \sum_{l=1}^{n-1} j_l]$. The number of items retained locally by node n is

$$\begin{aligned} j_n &= min(max\{u : \frac{t_r - t_l}{l^{s_n}} > \frac{t_s - t_l}{[(n+1)C - \sum_{i=1}^{n-1} j_i - u + 1]^{s_n}}\}, C) \\ &= min(\lfloor \frac{(n+1)C - \sum_{i=1}^{n-1} j_i + 1}{1 + (\frac{t_s - t_l}{t_r - t_l})^{1/s_n}} \rfloor, C) \end{aligned} \tag{3.4.2.2}$$

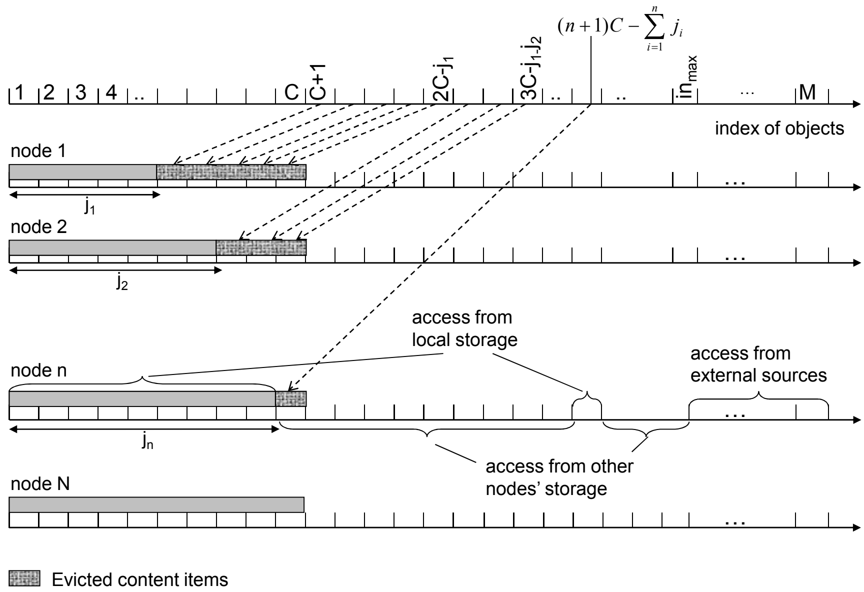


Figure 3.2: Self-aware cooperative strategy, rank-preserving dissimilarity: evicted and inserted content items per node.

Moreover, the index of the last inserted object in the group, after all N nodes have finished with their second step, is $in_{max} = NC - \sum_{i=1}^{N-1} j_i$. With these quantities at hand, we can state the following result for the level of object replication at the caches of the group nodes.

Proposition 4. *Under the self-aware cooperative placement strategy and rank-preserving dissimilarity across nodes' preferences, the number of replicas within the group is: N for objects with (global) indices in $[1, j_1]$ (full replication); $r, 1 \leq r < N$, for objects with indices in $[j_1+1, C]$; one for objects with indices in $[C+1, in_{max}]$; and, zero for objects with indices in $[in_{max}+1, M]$.*

Proof. We first prove that the number of evicted/inserted objects decreases with the node index; namely

$$j_n \leq j_{n+1} \text{ for } 1 \leq n \leq N-1 \qquad (3.4.2.3)$$

so that the items of the selfish placement retained by node n are a subset of those stored by all nodes m with $m > n$. It can be easily checked that

$$a = \frac{(n+2)C - \sum_{i=1}^{n} j_i + 1}{1 + (\frac{t_s - t_l}{t_r - t_l})^{1/s_{n+1}}} > \frac{(n+1)C - \sum_{i=1}^{n-1} j_i + 1}{1 + (\frac{t_s - t_l}{t_r - t_l})^{1/s_n}} = b \qquad (3.4.2.4)$$

since $s_{n+1} > s_n$ and $j_n \leq C$. Hence, it can only hold $j_n = j_{n-1}$, when $\lfloor a \rfloor = \lfloor b \rfloor$; or, $j_n < j_{n+1}$, in all other cases.

Therefore, the j_1 objects retained by node 1 are also retained by the remaining $N-1$ nodes, *i.e.*, all j_1 objects are fully replicated. The objects $[C+1, in_{max}]$ are those objects that are inserted during the second step of the strategy. As such, they cannot be inserted more than once in the group, due to (P2). Items with intermediate indices are surely replicated more than once but less than N times. In fact, the number of replicas is $N-1$ for objects $[j_1+1, j_2]$ when $j_1 < j_2 \leq C$. Likewise, the replication count is $N-n$ for objects $[j_n+1, j_{n+1}]$ whenever $j_n < j_{n+1} \leq C$. □

Corollary 3.4.1. *The original selfish placements of all nodes remain intact during the second step when*

$$(\frac{t_s - t_l}{t_r - t_l})^{1/s_1} < 1 + 1/C \tag{3.4.2.5}$$

Proof. The self-aware cooperative placements coincide with the selfish ones if $j_1 \geq C$. Replacing j_1 from (3.4.2.1), the condition follows directly. □

Likewise, we can derive the condition for inserting $(N-1)C$ items during the second step so that eventually NC different items, be stored within the group.

Proposition 5. *Under the self-aware cooperative placement strategy and rank-preserving dissimilarity, NC different content objects will be stored in the caches of the group nodes, each represented only once, if*

$$s_{N-1} < \frac{log(\frac{t_s - t_l}{t_r - t_l})}{log(NC)} \tag{3.4.2.6}$$

Proof. To have all objects only once replicated within the group, nodes [1,N-1] should replace *all* C items of their original selfish placements with non-represented ones. The last node, node N will always retain its original selfish placement since these most preferred objects will not be replicated anywhere else in the group, hence it cannot evict them (P1). Therefore, objects $[nC+1, (n+1)C], 1 \leq n \leq (N-2)$ should be inserted at node n; the ultimate requirement is that the object NC replaces the first, most preferred object, of node $N-1$,

$$\frac{t_s - t_l}{(NC)^{s_{N-1}}} > \frac{t_r - t_l}{1^{s_{N-1}}}. \tag{3.4.2.7}$$

Solving for s_{N-1} yields (3.4.2.6). □

With these results at hand, we can compute the per-node access cost as

$$\mathcal{C}_n^C(\mathcal{P}) = \frac{\sum_{i=1}^{j_n} i^{-s_n}}{\sum_{i=1}^{M} i^{-s_n}} t_l + \frac{\sum_{i=in_{max}}^{M} i^{-s_n}}{\sum_{i=1}^{M} i^{-s_n}} t_s + \frac{\sum_{i=j_n+1}^{in_{max}} i^{-s_n} - \sum_{i=nC-\sum_{l=1}^{n-1} j_l+1}^{(n+1)C-\sum_{l=1}^{n} j_l} i^{-s_n}}{\sum_{i=1}^{M} i^{-s_n}} t_r \tag{3.4.2.8}$$

whereby the objects that a node eventually accesses from the other group nodes' caches are the full set of non-represented objects that are inserted at the second step of the strategy (algorithm) *minus* those locally stored at that node.

Shape-preserving dissimilarity

As with selfish placements, we need to distinguish between two cases:

a) $k > C$: assuming that $N(C+k) \leq M$, the selfish placements in the first step result in the placement of NC different objects in the caches of the group nodes, each one represented only once. According to (P1), nodes do not evict objects from their caches; hence, the placements under the self-aware cooperative placement coincide with those under the selfish strategy.

b) $k < C$: the selfish placements give rise to overlaps in the contents of nodes' caches. The number of different objects that are placed in the whole group are $m_d = (n-1)k + C$ and their replication count varies in $[1, \lfloor \frac{C}{k} \rfloor]$. Moreover, for $k < \lfloor C/2 \rfloor$, the $m_d - 2k$ objects stored by the group nodes feature at least two replicas and could be evicted by one or more group nodes in the second step.

Whereas, the exact computation of the per-node access cost in this case is cumbersome, it is easier to prove the following result.

Proposition 6. *The placements under the self-aware cooperative strategy and shape-preserving dissimilarity across nodes' preferences coincide with the selfish placements when*

$$k/C > (\frac{t_s - t_l}{t_r - t_l})^{1/s} - (1 + 1/C) \tag{3.4.2.9}$$

Proof. The two placements will coincide as long as no object evictions/insertions are made during the second step. Given that the local indices of objects $[m_d + 1, M]$, hence their preference rank, increases for higher node indices, two conditions should be met so that nodes' original placements do not change.

- the $(N-1)^{th}$ node should not (find it profitable to) evict its least preferable object currently in its cache.
- the last node should not (find it profitable to) evict its least preferable object that is replicated in the group.

The first condition translates to

$$\frac{t_r - t_l}{C^s} > \frac{t_s - t_l}{(C+k+1)^s} \Rightarrow \frac{C+k+1}{C} > (\frac{t_s - t_l}{t_r - t_l})^{1/s} \tag{3.4.2.10}$$

whereas the second condition can be expressed as

$$\frac{t_r - t_l}{(C-k)^s} > \frac{t_s - t_l}{(C+1)^s} \Rightarrow \frac{C+1}{C-k} > (\frac{t_s - t_l}{t_r - t_l})^{1/s} \tag{3.4.2.11}$$

since $(C+1)/(C-k) > (C+k+1)/C$, $\forall k > 0$, the first inequality is the active constraint and (3.4.2.9) results trivially. □

Therefore, equations (3.4.2.5) and (3.4.2.9) already suggest that the placements emerging under the self-aware cooperative and selfish strategies tend to coincide as the exponents of the Zipf preference distributions ($s_1 \propto p$ for rank-preserving dissimilarity) and shift parameter k increase. In other words, since tightness decreases with s and k (see Table 3.1), the gain under cooperation fades out as the content preferences of nodes diverge. We elaborate on this result in the next section.

3.5 Evaluation results for different similarity patterns

The numerical examples in this section illustrate how group similarity, aka *tightness*, shapes the tradeoffs induced by the three behavior-based content placement strategies. Therefore, they help establish guidelines as to which behavior (strategy) would be beneficial to individual nodes and/or the entire group, under given similarity levels in the preferences of the nodes in the social group.

In the numerical examples in this paper initially $N = 5$ nodes; a small number of nodes helps us better illustrate and discuss the results regarding the content access cost for each node. The default value for node storage capacity is $C = 10$ objects, for object population $M = 50$ objects and for the costs $t_l = 0$, $t_r = 10$ and $t_s = 20$ cost units.

3.5.1 Placement strategy comparison at node level

Fig. 3.3 considers separately the two scenarios for preferences' dissimilarity in Section 3.3. It plots the per-node access cost under the three content placement strategies and for different values of *tightness*, T. We discuss these viewpoints for very high and very low tightness values.

Social groups with infinite or very high *tightness*

The first important remark out of the two plots is that, under high tightness value (*i.e.*, highly similar content preferences of group nodes), the optimally altruistic strategy outperforms the other two regarding not only the cost for the entire group (by definition), but also the cost for individual group nodes (Fig. 3.3(a)). This relation holds irrespectively of the dissimilarity scenario in question and suggests that the optimally altruistic behavior is the clear winner-behavior for every node in a very tight social group.

The second noteworthy outcome when there is high similarity in the preferences of the group nodes, relates to the performance achieved by the self-aware cooperative strategy. More specifically, the access costs for both individual nodes and the entire group: (a) are very close to (slightly higher than) those under the optimally altruistic strategy; and (b) are always lower than those under the selfish strategy, as expected given that it is mistreatment-free (see Section 2.2). Given its lower implementation complexity compared to the optimally altruistic one, it becomes an attractive alternative for node-members of tight social groups. Therefore, the self-aware cooperative strategy achieves a good tradeoff between performance and complexity.

Looking closer into the rank-preserving dissimilarity plots, when tightness approaches infinity (Fig. 3.3(a) on the left), the access cost for all group nodes under the self-aware cooperative and the optimally altruistic strategies tend to become equal. Infinite tightness in the general case ($p = 0$, $s \neq 0$ in Section 3.3) implies that a given object is requested

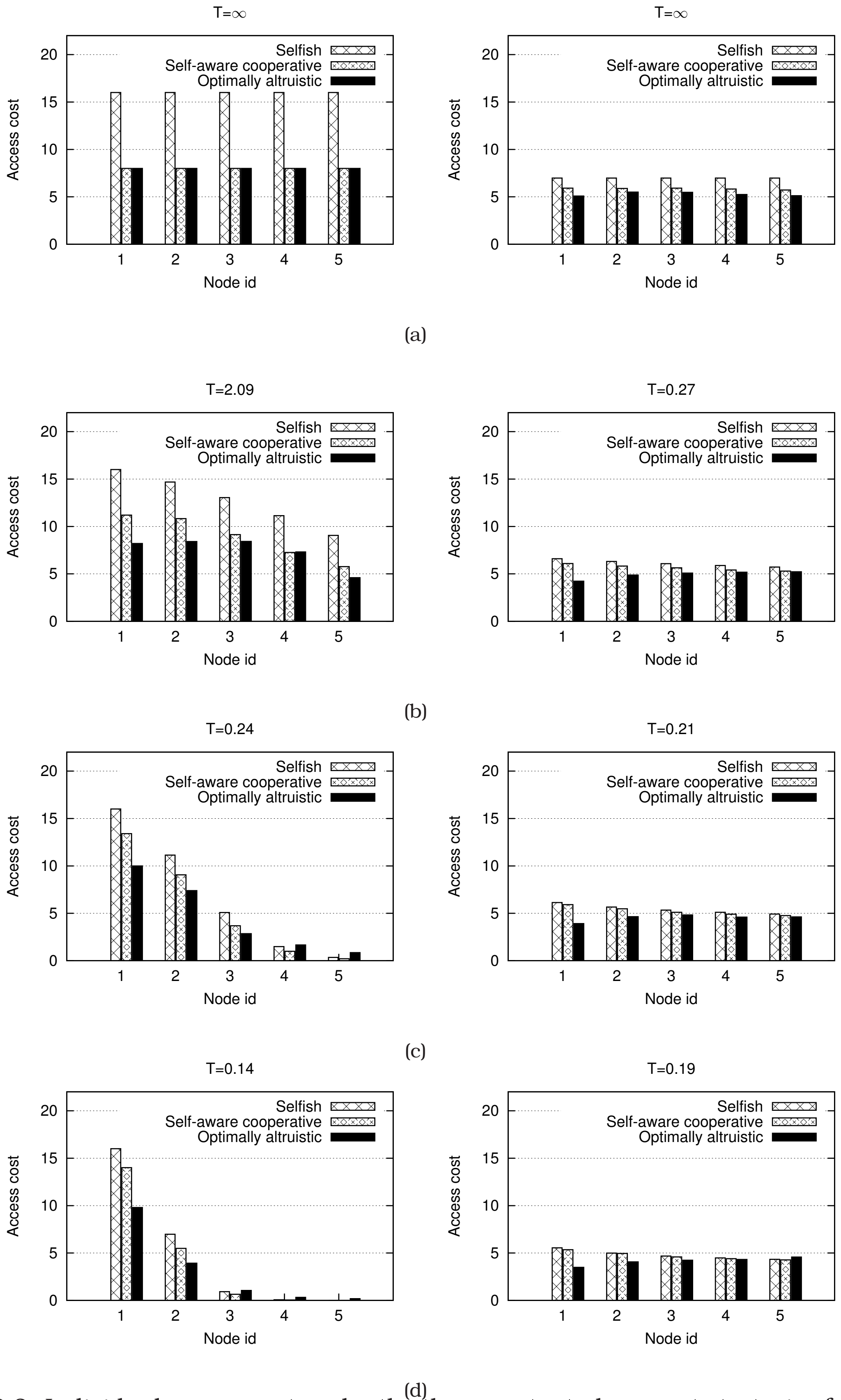


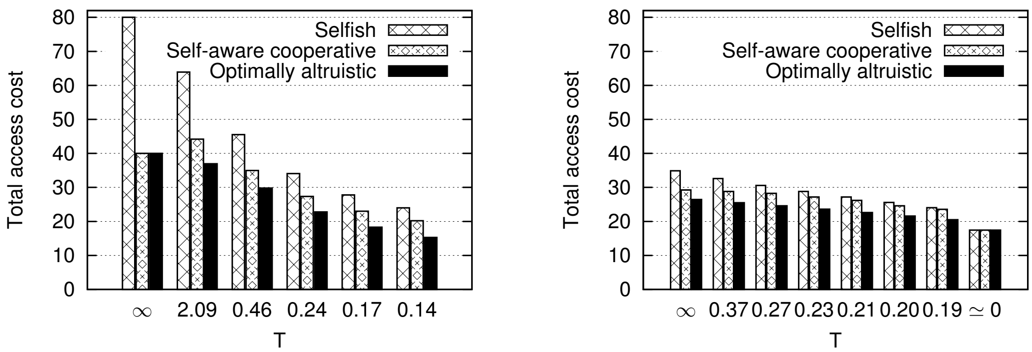
Figure 3.3: Individual access cost under the three content placement strategies for different values of *tightness* T, under rank-preserving (figures on the left) and shape-preserving (figures on the right) dissimilarity.

with the same intensity by all nodes. Both the self-aware cooperative and the optimally altruistic end up with the same object placement $\mathcal{P}$, which spreads all objects across the storage capacity of the group nodes. On the contrary, when nodes behave selfishly, they choose the same C objects for storage; namely, these C objects that attract the interest of all group nodes are replicated N times at group level. In the special case that individual node preferences are uniformly distributed ($s = 0$, $R_m^n = 1/M$, $\forall m \in \mathcal{M}$ and $n \in \mathcal{N}$), the nodes may choose any random combination of C out of M objects. The achieved diversity regarding the objects eventually stored across the whole group can vary (see Section 3.4). In the unfortunate worst-case scenario, where all nodes choose the same C objects, the access cost for each node gets its maximal value $((M-C)t_s + Ct_l)/M$. Thus, when all objects have the same request rate $1/M$, the selfish strategy results in an increase of the total access cost equal to $N(min(NC, M) - C)(t_s - t_r)/M$.

Summarizing, the tighter the social group, the more incentives the nodes have to behave in cooperative or, even, altruistic rather than selfish manner.

Social groups with low *tightness*

Whereas cooperation is the recommended behavior for nodes with highly similar interests, whether in the extreme form of altruism or the more conservative self-aware cooperative strategy, Fig. 3.3 and 3.4 suggest that it is far less attractive when the node preferences are more diverse.



(a) Rank-preserving (b) Shape-preserving

Figure 3.4: Total access cost *vs.* *tightness* T.

The first conclusion out of Fig. 3.3 concerns the way the total access cost is spread across the group nodes under the two types of dissimilarity and for low T values. The content access cost split under *rank-preserving dissimilarity* is uneven. Nodes with higher indices "pay" much less than nodes with smaller indices, irrespective of the adopted content placement strategy. This unfair cost distribution becomes more pronounced as tightness decreases. The reason behind this unfairness has to do with the demand distributions of the five nodes. Remember from Section 3.3 that the higher the node index the more skewed the node preference distribution. Hence, the preference of nodes is more concentrated around fewer top-ranked objects and there is less demand for the remaining objects that have to be accessed from remote storage, whether from the $N-1$ group nodes at cost t_r or outside the group at cost t_s.

On the contrary, under the *shape-preserving dissimilarity* scenarios, the total access cost is more uniformly split across the group nodes and smaller in absolute values. Moreover, the relative increase of access cost under the selfish strategy is smaller. There is far more diversity in the content that different nodes are interested in. Therefore, more content ends up being accessible from *within the group* at cost t_r rather than from outside the group at cost t_s, so that the nodes experience marginal benefit by committing to cooperation.

Turning into the pairwise comparison of the strategies with respect to the per-node content access cost under low T, we would note the following:

Optimal altruistic vs. selfish: Although the optimally altruistic strategy still minimizes the cost of content access at the group level, it cannot avoid mistreatment of individual nodes; for example, this is the case with Node 5 in Fig. 3.3(d), for both types of dissimilarity we consider. Two are the straightforward remarks when looking closer at Fig. 3.3: a) the number of mistreated nodes increases as the value of T drops; b) it is mainly the nodes with higher indices that are mistreated (this is more apparent for the rank-preserving dissimilarity scenarios). Both remarks can be explained by considering the way the two strategies actually operate. The optimally altruistic strategy shuffles the objects in a more radical way so that all nodes may result with objects that are different than their list of C top-ranked preferences.

In the case of rank-preserving dissimilarity, the preference distributions of nodes become more skewed for lower T values and, for given T, the skewness increases with the node index. Hence, the overall cost of storing objects other than the top-value ones is significantly less than the respective cost for higher-index nodes, whose preference is concentrated in very few objects. For this same reason, storing locally these objects under the selfish strategy, significantly reduces the cost for those nodes. Apparently, as tightness drops, more nodes find the selfish strategy more self-rewarding against the socially-optimal yet individual-node-oblivious altruistic strategy.

The explanation for the emergence of mistreatment under shape-preserving similarity scenarios is similar. The optimally altruistic strategy places objects that the majority of nodes want more. However, as tightness decreases some node preferences depart more significantly from the "mean" preferences and thus they can be mistreated.

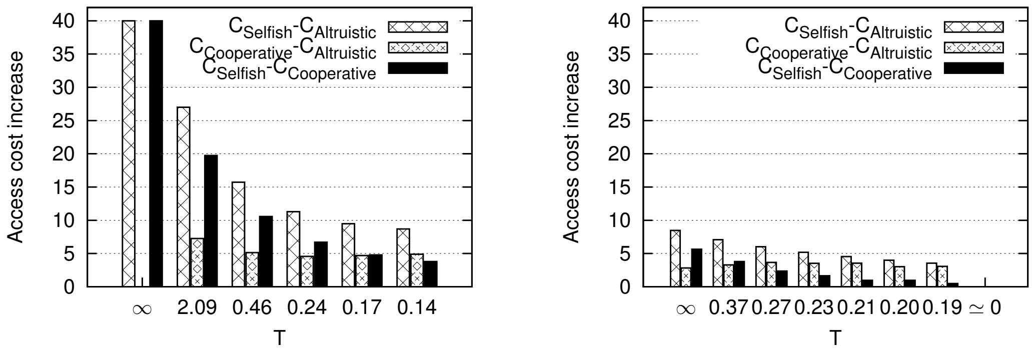
Overall, since the complexity involved in implementing the optimally altruistic strategy is larger (see Section 2.4), and the mistreatment of nodes threatens the very existence and operation of the social group as such, the selfish strategy emerges as a more attractive and stable strategy for the group nodes.

Self-aware cooperative vs. selfish: As tightness decreases we note that the access cost of self-aware cooperative strategy approaches that of the selfish. This happens because each node has its strongest interest in different objects. Hence, each node stores different objects locally, and thus the self-aware cooperative strategy achieves similar cost with that of the selfish one. This result leads to the conclusion that under dissimilarity nodes do not gain by cooperation or altruism (considering also the higher implementation cost of these strategies) and it would be better for them if they acted in isolation.

3.5.2 Placement strategy comparison at group level

Fig. 3.4 plots the absolute access cost values aggregated over all group members, whereas Fig. 3.5(a) and 3.5(b) report the access cost increase through pairwise comparisons of all strategies for different values of tightness. For example, the access cost increase between selfish and optimally altruistic strategy is $C_{Selfish} - C_{Altruistic} = \sum_{n=1}^{N} \mathcal{C}_n^S(P) - \sum_{n=1}^{N} \mathcal{C}_n^A(P)$, where $\mathcal{C}_n^S(P)$ denotes the access cost of node n for all the objects under the selfish strategy and $\mathcal{C}_n^A(P)$ denotes its access cost under the optimally altruistic one.

From Fig. 3.4 it is clear that as tightness decreases the self-aware cooperative and optimally altruistic strategies bring negligible benefits if any (the same (Fig. 3.4(b), $T \simeq 0$) or only slightly larger (Fig. 3.4(a), $T = 0.14$)) to the social group compared to the selfish one. Thus, in view of the fact that the complexity of the self-aware cooperative and optimally altruistic strategies, requiring information exchange among the nodes, is substantially larger than that under the selfish one, requiring no information exchange and lower computational complexity, it may be concluded that the selfish strategy should be preferred against the other strategies under low *tightness*.



(a) Rank-preserving (b) Shape-preserving

Figure 3.5: Pairwise access cost differences between the three strategies *vs. tightness* T.

From Fig. 3.5 the relative gain of the optimally altruistic strategy over the self-aware cooperative one is smaller than the gain over the selfish one, especially for higher values of *tightness*. Another interesting observation out of the results (Fig. 3.4(a) and 3.4(b)), seen more clearly in Fig. 3.5 is that in general the improvement of the optimally altruistic and self-aware cooperative strategies over the selfish one decreases as *tightness* decreases.

These confirm the deduction that, if the optimally altruistic strategy cannot be applied, then for small values of *tightness* it is better to use a selfish strategy, and for larger values the self-aware cooperative one.

From these results, we can also obtain a rough idea of the application ranges of each strategy. However the exact range of values of T where each strategy should be applied depends on the underlying preference distributions, as we see that these values are different for the two dissimilarity patters studied here. For example, under rank-preserving dissimilarity case the relative improvement of the optimally altruistic strategy against the other strategies is much greater, for almost all values of T.

3.5.3 Sensitivity analysis

Understanding the impact of group size, N, number of items, M, and level of cooperation would be useful in understanding how the strategies compare in a dynamic environment, where the above parameters change.

Sensitivity to number of objects (M) and group size (N)

We now explore the impact on the total number of objects, M, and the group size N on tightness as well as on the comparative performance of the selfish and self-aware cooperative strategies. The relative gain when being self-aware cooperative is quantified by the ratio of the total content access cost under the selfish strategy to that under the self-aware cooperative strategy, $R_{S/C}$. We compute it as $R_{S/C} = \frac{\sum_{n=1}^{N} \mathcal{C}_n^S(P)}{\sum_{n=1}^{N} \mathcal{C}_n^C(P)}$. where $\mathcal{C}_n^{C(S)}(P)$, denotes the cost of accessing all objects for node n under the self-aware cooperative (selfish) strategy.

In the set of experiments with variable M, we fix $N = 50$ and $C = 10$, whereas we fix $M = 500$ and $C = 10$, when N is varied. Moreover, we consider $t_s = 100$, to emphasize more the cooperation benefits.

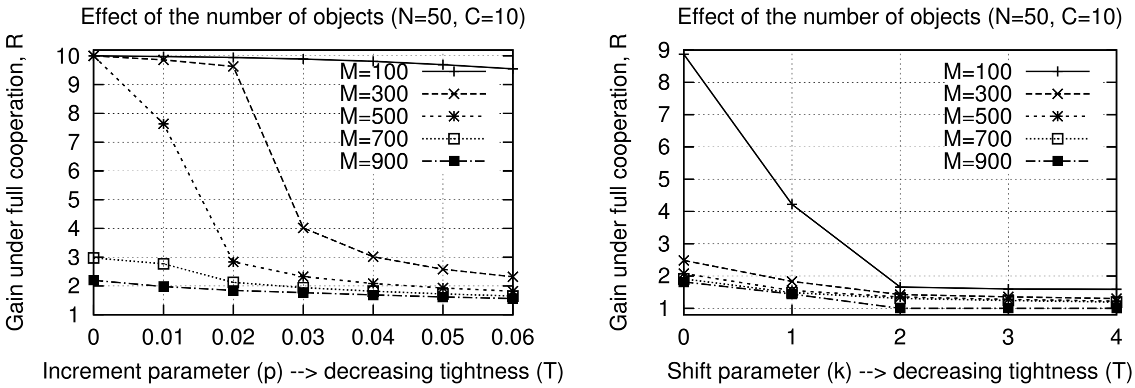
Table 3.2: Tightness values T for various M,$N = 50$, $C = 10$.

(a) rank-preserving (b) shape-preserving

p	Tightness (T)			k	Tightness (T)		
	M=100	M=500	M=900		M=100	M=500	M=900
0.00	∞	∞	∞	0	∞	∞	∞
0.01	9.731	7.771	7.392	1	0.172	0.162	0.161
0.02	1.831	1.251	1.125	2	0.163	0.135	0.133
0.03	0.722	0.457	0.400	3	0.155	0.123	0.121
0.04	0.421	0.268	0.235	4	0.143	0.116	0.114

Impact of number of content objects, M: For a given group of size N, more content means that the nodes' preferences are spread over more items. Therefore, additional non-zero terms are added to the pairwise D_{R^i,R^j} terms in (3.2.1) and T decreases, as Table 3.2 suggests. When content preferences are already highly dissimilar, the tightness decrease with content volume is smaller. Exceptions to this trend are the two extreme cases, where the increment (shift) parameter $p(k)$ for the rank(shape)-preserving dissimilarity scenarios is zero: adding more items with exactly the same number of nodes does not affect the group infinite tightness.

The relative gain $R_{S/C}$ grows in both directions that increase tightness values: a) with more *similar* content preferences within the group, inline with what was discussed in Section 3.5.2 and reading Fig. 3.6 from left to right; b) with more *focused* content preferences within the group over fewer content objects, inline with Table 3.2 and reading Fig. 3.6 from top to bottom. As the node preferences concentrate on fewer objects, chances are higher that a self-aware cooperative node will deem it self-beneficial to fetch a new content object from the external server. Hence, this object is made available to the other group nodes at lower cost than if they had to fetch it from external server(s) and the aggregate content access cost under the self-aware cooperative strategy is reduced.



(a) Rank-preserving

(b) Shape-preserving

Figure 3.6: Comparison of total content access cost under selfish and self-aware cooperative strategies for different M.

In more detail, under rank-preserving dissimilarity (Fig. 3.6(a)), increasing the content volume does not change the access cost ratio as long as it does not exceed the group storage capacity ($M < N \cdot C$). On the other side, increasing M beyond $N \cdot C$, implies that, irrespective of the strategy, we can no longer store all objects within the group. A significant contribution to the overall cost comes from objects that are stored outside the group, hence the self-aware cooperative has lower gain margins over the selfish strategy. The trend is the same under shape-preserving dissimilarity. Only now, even for $k = 0$, there is visible, yet lower in absolute terms, gain differentiation as long as $M \leq N \cdot C$.

Impact of group size N: In this set of experiments, we gradually increase the number of nodes N in the social group, assigning to them content preference distributions inline with the two dissimilarity scenarios described in Section 3.3. The corresponding tightness values are listed in Table 3.3 and suggest that tightness decreases monotonically with N. This is due to both the content preferences of superimposed nodes and the growing factor $N(N-1)$ in (3.2.1).

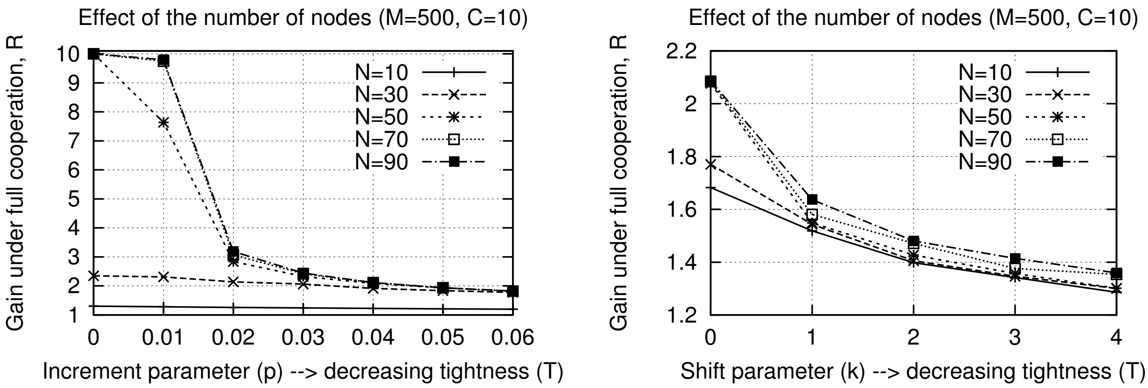
Table 3.3: Tightness values T for various N,$M = 500$, $C = 10$.

(a) rank-preserving (b) shape-preserving

p	Tightness (T)			k	Tightness (T)		
	N=10	N=50	N=90		N=10	N=50	N=90
0.00	∞	∞	∞	0	∞	∞	∞
0.01	264.838	7.771	1.668	1	0.273	0.162	0.140
0.02	60.917	1.251	0.322	2	0.207	0.135	0.123
0.03	24.837	0.457	0.185	3	0.180	0.123	0.111
0.04	12.786	0.268	0.144	4	0.164	0.116	0.107

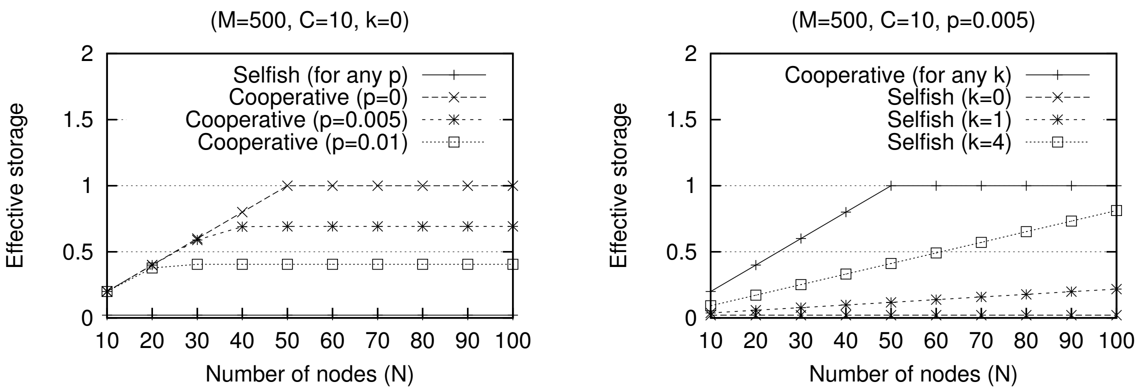
Whereas the impact of M and N on tightness is similar (*i.e.*, tightness decreases as they increase), the way they affect the relative gain of the self-aware cooperative strategy differs. Fig. 3.7(a) shows that this gain *increases* with larger N, hence lower tightness values. More nodes of diverse interests within the group result in an increase of the effective group storage, *i.e.*, the number of items that are collectively stored across all group nodes. Fig. 3.8 plots the normalized effective group storage, defined as the ratio of the number of different objects stored in the group to the total number of objects, M.

Under rank-preserving similarity, Fig. 3.7(a) suggests that under a uniform distribution of content preferences ($p = 0$), an increase in N has an impact on the access cost as long as $N < M/C$. The total access cost of the self-aware cooperative strategy can become up to half that under the selfish strategy as the normalized effective group storage grows to unity (Fig. 3.8). Having $N > M/C$ does not yield extra benefit because all objects in the network are already stored in the group. Thus, M/C can be seen as a threshold that shows the highest size of the social group that is needed in order to have an important benefit by employing the self-aware cooperative strategy. As p increases (and tightness decreases), the size of the group plays a less significant role in the content access cost and effective group storage. This is because the nodes' preference are more narrowly focused on the first few objects and, thus, less diversity of objects can be achieved within the social group.



(a) Rank-preserving (b) Shape-preserving

Figure 3.7: Comparison of total content access cost under selfish and self-aware cooperative strategies for different N.



(a) Rank-preserving (b) Shape-preserving

Figure 3.8: Effective group storage under selfish and self-aware cooperative strategies for different N.

Under shape-preserving dissimilarity and $k = 0$ (*i.e.*, nodes requesting the objects with the same preference order) an increase of N, when $N < M/C$, has an impact on the access cost (Fig. 3.7(b)) and the effective storage (Fig. 3.8(b)). However, for higher k

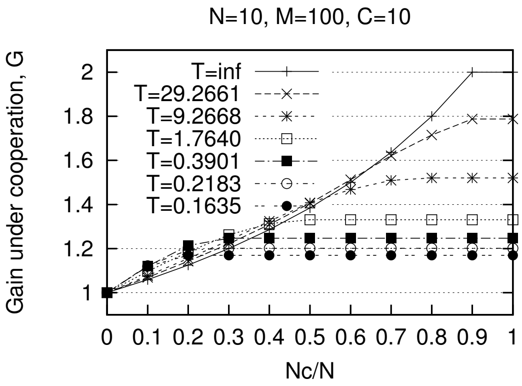
an increase of N brings less benefit to the cooperative strategy. This is because shifting preference orders creates a shuffling of interests between nodes which in itself brings diversity of content placement even under a selfish strategy. Whereas, the cooperative strategy can generate enough diversity irrespective of k, selfish strategies become competitive with larger k and diminish the achievable gain.

Overall, there is a negative correlation between tightness (social similarity) and the gain achievable through cooperation when we vary the number of objects that attract the interest of the group nodes, at least under the two dissimilarity patterns considered in this work. The gain over selfish behaviour grows as the preferences of a given group are concentrated on a smaller number of objects $M > C$, since the excess number of objects that are not stored locally or in the group is also smaller. On the other hand, as the number N of nodes in the group increases and given that the total capacity of nodes does not exceed the number of objects available (*i.e.*, $NC < M$), tightness can increase the cooperation gain. As the number of nodes increases, groups with higher value of tightness report greater gain under cooperation than those with lower tightness. This is due to the greater diversity of objects in the placements that result under a cooperative strategy (*i.e.*, the total number of different objects in the group is greater). However, the drawback of this improvement lies in the increased complexity of the cooperative strategy as N increases.

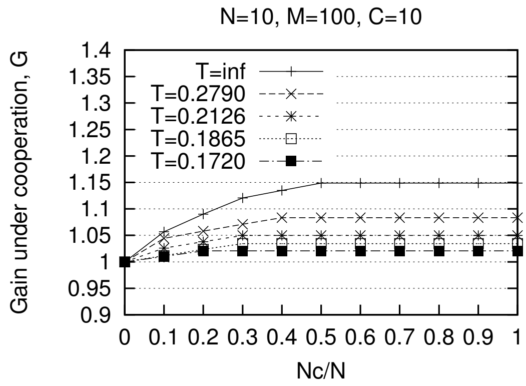
Sensitivity to cooperation

We now consider a mixed social group with both cooperative and selfish nodes and study how the performance of the self-aware cooperative strategy varies with the number of cooperative nodes. Our metric in this respect is the ratio of the aggregate content access cost for the mixed social group over that for the same group when all nodes behave selfishly, $G = \frac{\sum_{n=1}^{N} \mathcal{C}_n^S(P)}{\sum_{n=1}^{N} \mathcal{C}_n^{S,C}(P)}$, where $\mathcal{C}_n^{S,C}(P)$ denotes the access cost for node n averaged over all the objects when both selfish and cooperative nodes exist in the group. Equivalently, the ratio G measures the performance gain achieved by the self-aware cooperative strategy in the presence of nodes that *misbehave* and do not cooperate.

Fig. 3.9 shows that the performance gain G reaches a constant value as the fraction of cooperative nodes N_C/N in the group increases. The convergence is faster for smaller



(a) Rank-preserving



(b) Shape-preserving

Figure 3.9: A mixed social group scenario with both cooperative and selfish nodes.

values of *tightness*, but the gain G is smaller. On the other hand, for larger values of *tightness* a higher gain is achieved when more nodes become cooperative. In other words, under rank-preserving dissimilarity there is a tradeoff between the achievable gain and the fraction of cooperative nodes in the group. Fig. 3.9(a) shows that in social groups with highly similar preferences all nodes should be cooperative in order to have the highest access cost reduction. For less similar groups, the achievable gain is smaller but the cooperation of 40% of nodes suffices to obtain it.

On the contrary, this tradeoff does not emerge in the shape-preserving dissimilarity scenario, where anyway the tightness values and the achievable gains are smaller, as shown in Fig. 3.9(b). As we have discussed in earlier sections, there is already sufficient "mixing" of objects even under strictly selfish placement decisions.

3.5.4 Responsiveness of the self-aware cooperative strategy to changes in user preferences

In this set of simulation experiments we study how fast the self-aware cooperative strategy responds to changes in the content preferences of users. Initially the nodes' caches are empty. Nodes are let learn the content preferences of users over time and every time they receive a number of requests, say R, they run one iteration of the self-aware cooperative strategy; namely, each node n updates its cache with the C_n most requested content items (selfish first step) and then takes turn into adjusting its placement through evictions and insertions of new items, based on its running estimates for the user's interest in them. We let $N = 10$, $M = 10000$ items and $C_n = C = 10$ for all nodes.

In the course of the simulation, we change the content preference distributions of users twice. In the beginning, all nodes receive requests for content following the Zipf distribution with exponent $s = 1$, *i.e.*, there is high similarity in the content preference across the group nodes and the cooperation benefits are maximized. After 2000 requests are received by all nodes, we reverse the content preference distributions so that $R'^n_m = R^n_{M-m+1}, \forall m \in \mathcal{M}$ and $n \in \mathcal{N}$ (*rank-reversing*). In Fig. 3.10, we see that temporarily, till nodes gradually inform their caches and converge to the right placements, the nodes spend much more than the average expected cost for accessing content. Each plotted point is the cost averaged over the last 50 requests received by all nodes and 100 simulation runs. The system gradually improves its performance and drops the running average access cost to within $a = 6\%$ of the expected access cost (113.93) after $r(a) = 1719$ requests. Note that the total expected access cost remains the same in this case since the distributions are symmetric. After further 18000 requests, the preference distributions change again, this time following the shape-preserving dissimilarity pattern with shift value $k = C/2$ (*shift-preference*). This time the overshoot of the running aggregate access cost is milder than the first change; yet it takes the algorithm 4232 requests before the cost converges to $a = 6\%$ of the steady-state expected aggregate access cost.

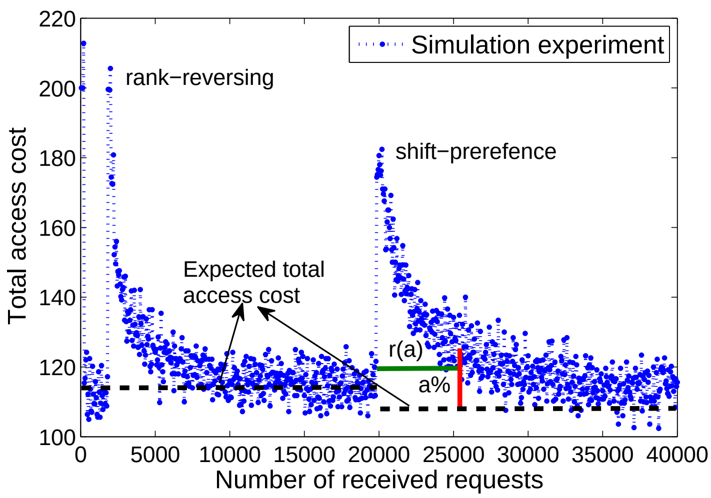


Figure 3.10: Performance of self-aware cooperative as function of number of received requests.

The frequency of placement adjustments (*i.e.*, iterations of the self-aware cooperative algorithm), induces a tradeoff between the excess access cost in the transient phase, during which the nodes fill their caches in response to their estimates about the nodes' preferences, and the overhead of the algorithm execution, both in terms of computations and network resources.

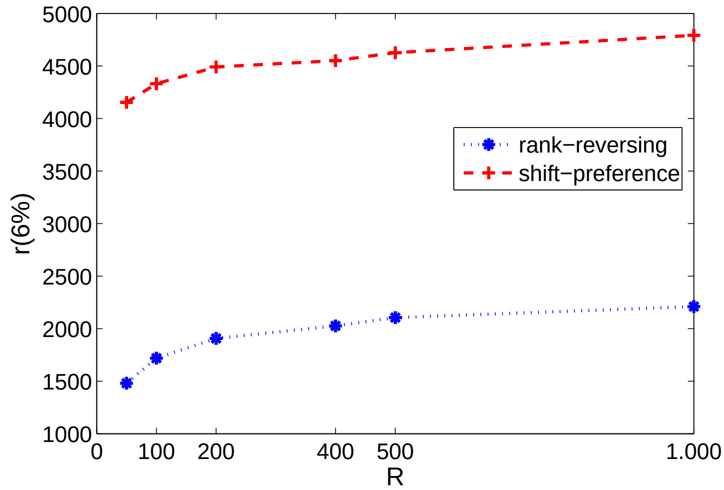


Figure 3.11: Access cost convergence time *vs.*period of placement algorithm execution, measured in number of requests.

Fig. 3.11 plots the access cost convergence time $r(6\%)$ against the period of algorithm execution R, both measured in number of requests, for the *rank-reversing* and *shift-preference* types of content preference changes. Although frequent invocations of the algorithm can accelerate the convergence of the algorithm upon disruptive changes in the users' preferences, the performance gap between lower and higher R values is rather moderate. Considering that these scenarios represent extreme cases of content preference changes across the social group population, we conclude that the self-aware cooperative strategy can respond rather fast to stochastic changes of users' content preferences even at moderate execution frequencies.

Chapter 4

Community detection in social networks

The term community denotes a social group of people that have one or more things in common. Whether this is residence, geographical neighborhood, traditions, or interests and ideals, communities have been long attracting the interest of sociologists and psychologists thanks to their potential to motivate and shape human behavior. On the contrary, virtual communities have emerged more recently and, almost always, transcend distance barriers. Empowered by the Internet, users in online communities socialize in virtual spaces provided by social networking sites. A major research question is then how could the dynamics of these virtual communities be exploited for more efficient design of networked communication protocols, that will increase user benefits.

4.1 Introduction

As we saw in the previous chapter, higher similarity in the interests/preferences of online social group members favors collaborative, and even altruistic, behavior in content replication. This was also demonstrated for content dissemination scenarios [Allen *et al.*, 2010]. But is such similarity present in social networks, where users tend to select their friends/followers with very different criteria, including acquaintance, social status, educational and family background? To answer this question, we need to devise mechanisms and tools that can assess the similarity of interests among social group members and leverage the structure this similarity embeds in their social network.

Our work addresses this requirement by poring over the interest-based community detection problem. We propose a framework, which we call "*ISCoDe*", for assessing the similarity in online social communities (Fig. 4.1). Input to *ISCoDe* are the interests of the communities' member nodes in certain thematic areas, hereafter called "interest classes", such as music, sports, art. Each interest class could further be split into subcategories (*tags*). In Section 4.4, we give an example of how the end-user interests can be inferred out of a real social network application. *ISCoDe* then proceeds in two steps. First, it quantifies the interest similarity between node pairs through the use of interest similarity metrics. Outcome of this step is a weighted graph representation of the social network, with edge weights corresponding to the similarity metric values. In a second step, *ISCoDe* can invoke standard community detection algorithms for weighted graphs (for example, [Newman, 2004b], [Clauset *et al.*, 2004]) to group nodes into disjoint clusters, connected internally by high-weight edges and to other subsets' nodes with small- or zero-weight edges. These algorithms also assess, in the same time, the quality of this grouping through the modularity metric [Newman & Girvan, 2004].

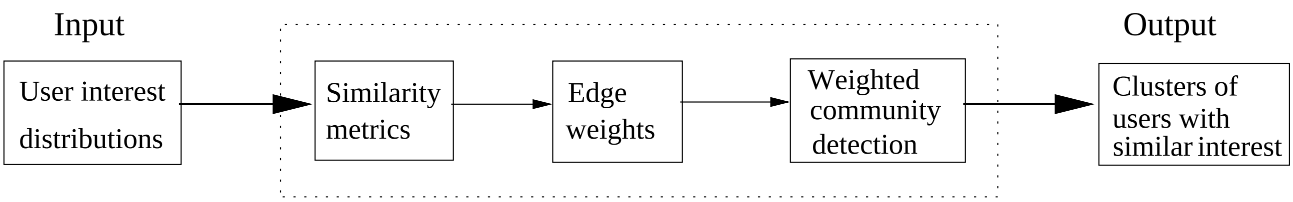


Figure 4.1: The *ISCoDe* framework.

We call *ISCoDe* a framework since there are more than one options for its two main processing steps, the derivation of the weighted graph edges and the community detection algorithm. Part of our work, hence, is devoted to the analysis and assessment of these options. For the derivation of graph edge weights, we consider two metrics: the Proportional Similarity [Vegelius *et al.*, 1986] and the inverse of the symmetrized Kullback-Leibler divergence [Kullback, 1959]. Effectively, each metric could be seen as a different *transformation* from one data set (distribution of user interests over interest classes) to another (graph edge weights). Comparing the outcomes of *ISCoDe* under synthetic user interest distributions, we show that the choice of the similarity metric affects both the sensitivity and the resolution properties of our framework.

In the literature, algorithms for detecting community structure have largely been applied to a given network structure, usually modeled as a graph. The most prominent algorithm thereof is that of Girvan and Newman [Newman, 2004c], which is highly efficient and overcomes many shortcomings of previously proposed algorithms, such as graph partitioning (*e.g.*, spectral bisection [Pothen *et al.*, 1990], Kernighan-Lin algorithm [Kernighan & Lin, 1970]) and hierarchical methods (*e.g.*, Euclidean distance single linkage clustering) [Fernández & Gómez, 2008]. These methods are not ideal for analyzing general network data since usually it is not known in advance in how many communities the network should be split into and which is the best division.

Newman further proposed a simple mapping from a weighted network to an unweighted multigraph and proposed an algorithm for detecting communities in weighted networks [Newman, 2004b]. The graph edge weights introduce another set of variables in the community detection process and it is shown in [Fan *et al.*, 2007] that they can have big influence on the resulting community structure, especially on dense networks. Note that the similarity metrics we consider are not related to the similarity indices that capture structural equivalence, *i.e.*, same profile of relations to all other nodes in the network, such as the Pearson correlation and the Jaccard coefficient.

Current practice in community detection consists in applying modularity-maximizing clustering algorithms over *given* (weighted) graphs. Our work has a different starting point: the network structure, *e.g.*, edge weight set, is not given beforehand. It is rather generated by *ISCoDe* out of the distributions of user interests over different thematic areas, the only information we assume known and given to us. Since our framework uses the interest distributions as its input for community detection, in the same time it becomes a means of assessing the effectiveness of similarity metrics that carry out the *mapping* of interest distribution differences. Our work, hence, explores how effectively different *mappings* facilitate the detection of the underlying interest similarity structure when the “commodity” community detection algorithms are applied on their *images*, *i.e.*, the weighted edge sets they generate. Through the application of the framework and the presented results, the effectiveness of the proposed framework that advocates projecting distributional differences to a weighted graph and using commodity approaches for identifying communities thereof, is assessed and established.

4.2 Detailed implementation of the *ISCoDe* framework

ISCoDe consists of two main phases. First, there is a *mapping* from the interest distribution space of the network nodes to pairwise similarity indices. These indices become the edge weights in the weighted graph. Then, the edge weight set is fed into a commodity community detection (graph clustering) algorithm that extracts possible community structure embedded in the network.

In general, we want *ISCoDe* to satisfy three main requirements:

Correctness: The framework should be able to distinguish correctly existing community structure. Whereas it may not always be possible to conclude whether such structure really exists, the outcome of the framework should at least agree with our intuition in certain benchmark scenarios, where this structure is evident.

Sensitivity: The framework should be able to adapt to changes of the user interest distributions and reflect the strength of the community structure.

Resolution: The framework should be able to identify important community structure irrespective of its scale and the overall network size.

We evaluate *ISCoDe* along these lines in Section 4.3. In the rest of this section, we detail the two processing steps of the framework and present baseline choices for populating them.

4.2.1 From interest distributions to a weighted graph

Let $\mathcal{N} = \{1, 2, \ldots, N\}$ be the set of the network nodes (online social network users) and $\mathcal{M} = \{1, 2, \ldots, M\}$ the set of interest areas (classes). We assume that for each node n we can have an estimate of R^n, the probability distribution of its preferences over the M interest areas, which takes discrete values $R_1^n, R_2^n, \ldots, R_M^n$ with $\sum\limits_{m \in \mathcal{M}} R_m^n = 1$. Practically, R_m^n could be measured through the normalized request rate of node n for data objects (content) of type m or some other form of interest expression in a certain area (*e.g.*, subscription to this category's tags). In Section 4.4, we describe this process for a particular online social application.

From the node interest distributions, we can then compute the pairwise similarity in the interests of two nodes drawing on measures of distributional similarity. The pairwise similarity is then used as weight of the a connecting the pair of nodes. In this way, a weighted graph is constructed. Hereafter, we describe and focus on two of the possible choices: a) the Proportional Similarity (PS) metric, which is shown in [Vegelius *et al.*, 1986] to satisfy 11 criteria suggested as suitable for a measure of similarity between distributions; and b) the inverse of Kullback-Leibler symmetrized divergence (InvKL) [Kullback, 1959]. InvKL projects the difference between two interest distributions to a significantly broader range of values compared to the PS metric, *i.e.*, $(0, +\infty)$ *vs.* $[0, 1]$, thus shaping the resolution properties of the framework, as we will see later in Section 4.3.

Proportional Similarity (PS) metric

With the PS metric, the interest similarity PS_{R^i,R^j} between two nodes i and j, with interest distributions R^i and R^j, equals [Vegelius *et al.*, 1986]:

$$PS_{R^i,R^j} = 1 - D_{R^i,R^j} = 1 - \frac{1}{2}\sum_{m=1}^{M} \left| R^i_m - R^j_m \right|, \qquad (4.2.1.1)$$

where $D_{R^i,R^j} = \frac{1}{2}\sum_{m=1}^{M} |R^i_m - R^j_m|$ is the total variation distance.

Inverse KL (InvKL) symmetrized divergence

Our second metric is the inverse of the Kullback-Leibler (KL) symmetrized divergence [1], a metric capturing the distance between two distributions

$$InvKL_{R^i,R^j} = \Big(\sum_{m=1}^{M} R^i_m log\frac{R^i_m}{R^j_m} + R^j_m log\frac{R^j_m}{R^i_m} \Big)^{-1}.$$

The InvKL metric takes values in $(0, +\infty)$. The KL divergence goes to infinity in cases where there is zero interest in an interest class from one node, and nonzero interest in it from another. In order to avoid such problems, different methods can be used as mentioned in the the previous chapter (see Section 3.2.2). Here we will consider that all nodes have positive interest for all interest classes.

[1] In the previous chapter we used *tightness*, defined to be the inverse of the average KL symmetrized divergence between nodes in a social group.

4.2.2 From a weighted graph to communities

Out of the full population of clustering algorithms, relevant to our objectives are those carrying out density-based graph clustering [Brandes & Erlebach, 2005]. Namely, they take as input a graph and partition it in a way that some notion of density (in our case: the weights of intra-cluster edges) is significantly higher within a partition than across different partitions (inter-cluster edges). Within the complex networking community the de-facto criterion for assessing the quality of the partitioning is modularity [Newman, 2004b], [Newman & Girvan, 2004]. Modularity sums across all partition clusters the fraction of within-cluster edges minus the expected fraction of edges that would fall within the same cluster were they selected at random. For a given partition of a weighted graph $G(V, E)$, where V is the set of network nodes and E the edge set capturing pairwise interest similarities, modularity Q equals [Newman, 2004b]

$$Q = \sum_{c=1}^{C} \left[\frac{l_c}{L} - \left(\frac{d_c}{2L} \right)^2 \right], \tag{4.2.2.1}$$

where the sum is over the C communities of the partition, L is the sum of the weights of all edges in the graph, l_c is the sum of weights over edges lying fully within community c, and d_c the respective sum over the full set of edges incident to nodes in c. Modularity takes values in the interval $[-1/2, 1]$ [Brandes *et al.*, 2008]. It becomes zero for community structures that do not differ than what one would get by random chance, whereas values above $0.3 - 0.4$ suggest strong community structure.

Our framework lends to the use of different modularity-maximization algorithms. One example is the divisive clustering algorithm Newman proposed in [Newman, 2004b] for weighted graphs. The algorithm iteratively removes from the original graph the edge with the highest "edge betweenness" (defined as the number of shortest paths between pairs of nodes traversing the edge) and recalculates modularity and edge betweenness values till modularity does not increase any further. The complexity of the algorithm is $O(|E|^2|V|)$, which for dense graphs yields $O(|V|^5)$.

More generally, the problem of finding a partition that maximizes modularity in general graphs has been formulated as an Integer Linear Program (ILP) and shown to be NP-hard [Brandes *et al.*, 2008]. Proposed heuristic algorithms for modularity maximization draw on simulated annealing [Reichardt & Bornholdt, 2006] or extremal optimization

[Duch & Arenas, 2005]. More commonly used and computationally friendlier, however, is the greedy agglomerative clustering algorithm of Clauset *et al.* [Clauset *et al.*, 2004, Newman, 2004a]. We simply extend it to weighted graphs by directly relating it with the definition of modularity in weighted graphs in (4.2.2.1). Initially each vertex is viewed as a discrete cluster of size one. The algorithm then iteratively merges the two clusters that yield the largest modularity increase. The algorithm completes in at most $|V| - 1$ steps and has an implementation cost of $O(|V|^2 log|V|)$ [Brandes *et al.*, 2008] permitting scalability for large graph sizes. We retain the greedy algorithm as the baseline for the assessment of *ISCoDe* in Section 4.3.1.

4.3 *ISCoDe* evaluation setting

We work with synthetic networks of N member nodes with *controllably* similar interests in order to evaluate the correctness, sensitivity, and resolution properties of the framework. With modularity as the fitness metric of the detected community structure, structures featuring tighter communities with cleaner separation from each other should see higher Q values than equinumerous yet "looser" structures. Moreover, with respect to *ISCoDe*'s resolution, we recall the remarks by Fortunato and Barthèlemy in [Fortunato & Barthélemy, 2007] that algorithms seeking to maximize modularity may fail to identify important structures smaller than a scale. In concluding whether the identification of further distinct communities within a single one is meaningful, we adopt the weak "community" condition by Radicchi [Radicchi *et al.*, 2004], *i.e.*, a community c is correctly identified as one if

$$\frac{l_c}{L} - \left(\frac{d_c}{2L}\right)^2 > 0. \tag{4.3.0.1}$$

The study was for unweighted graphs. We could derive similar results for weighted graphs.

Note that in *ISCoDe* the resulting modularity values are significantly affected by the choice of the similarity metric. Contrary to other studies in literature, where community detection algorithms maximizing modularity are studied on given complex weighted graphs, *ISCoDe* adds the additional transformation step of interests to graph edge weights. Therefore, another requirement from the evaluation process is to show how the two interest similarity metrics affect the three framework requirements.

In the general setting, the network population is organized into k groups. Each group is interested in M interest classes. We form k equal-size groups of N/k users: nodes $1..N/k$ are assigned to group 1, nodes $N/k+1..2N/k$ to group 2, and so on (for the sake of the example, we take N/k to be an integer). We then control the similarity within and across the k groups as follows:

Interest similarity *across* groups. This is controlled in two ways. Firstly, through the number of common interest areas between groups, which may take any value r in $[0, M]$. Secondly, and this relates to the way the similarity *within* a single group is controlled, through the way the interests overlap. We consider two scenarios for the distribution of common interests between two groups: a) the r common interest areas are simultaneously the r least interesting for group g and the r most interesting for group $g+1$, $0 < g < k$ (*L(ast)-F(irst)*, Table 4.1(a)); b) the r common interest areas are the r most interesting for the users of all k groups (*F(irst)-F(irst)*, Table 4.1(b)).

Table 4.1: Example of the two interest-overlap scenarios for $k = 3$, $M = 5$. The order of interest classes marks also the order of nodes' interests within a group.

(a) L-F with a single overlap interest class ($r = 1$)

Group 1	Group 2	Group 3
1	5	9
2	6	10
3	7	11
4	8	12
5	9	13

(b) F-F with two overlap interest classes ($r = 2$)

Group 1	Group 2	Group 3
1	1	1
2	2	2
3	6	9
4	7	10
5	8	11

Interest similarity *within* groups. The interests of nodes within a group are spread over the ordered M interest classes according to a Zipf distribution. The skewness parameter s of the distribution differs for each group node. The interest of the first node of each group are uniformly distributed ($s_1 = 0$) and s increases with constant step p so that for node n, $s_n = p(n-1), p \in \mathcal{R}$. Higher p values increase the skewness in the interest distribution and concentrate the node interests' mass in the higher-order interest classes.

Interestingly, changes of p also affect the similarity of interests between nodes belonging to different groups depending on the overlap scenario (Table 4.1): higher p values result in weaker (stronger) inter-group similarity in the $L - F$ ($F - F$) overlap scenario.

In summary, by calibrating p, the overlap scenario and the number of common interest classes, we can produce networks with community structures of variable discernibility.

4.3.1 Experimental results and discussion

We show and discuss representative results from our experimentation with *ISCoDe* on synthetic networks that outline the main behavior of the framework. All experiments are carried out with the greedy agglomeration algorithm in [Clauset *et al.*, 2004] since it yields significantly faster run times than its competitors[2].

Correctness and sensitivity experiments

In this set of experiments, $N = 80$ and $k = 4$. The impact of M was found to be minimal, thus we present herein results only for $M = 20$. We vary the interest overlap scenarios (as in Table 4.1), the number of common interest classes, one (Tables 4.2(a), 4.2(c)) or half of them (Tables 4.2(b), 4.2(d)), and the skewness of the interest distributions, smaller p values representing more uniform distributions within a single group nodes.

The first remark is that both metrics produce the same intuitive community partitions in the presence of strong community structure, as in Tables 4.2(a) and 4.2(c) for low p values. On the contrary, when such structure is not evident, the two metrics result in considerably different partitions.

The second remark has to do with the higher sensitivity of the framework when the PS metric is used in its first processing step. The modularity of the resulting partitions under the PS metric decreases when the interests of nodes are more randomly diffused over the different interest classes and goes down to zero when the similarity structure tends to disappear, as in Table 4.2(d). On the contrary, the resulting modularity values under

[2] We run experiments also with the divisive clustering algorithm in [Newman, 2004b] but we had to restrict to small group sizes. In these cases, we obtained similar results with respect to community structure and modularity values.

Table 4.2: Correctness and sensitivity experiments: Modularity and communities formed for different values of p, $N = 80$, $M = 20$.

(a) L-F, 1 common object

	PS			InvKL		
	Q	C	partition	Q	C	partition
$p = 0.02$	0.6849	4	{1..20}...{61..80}	0.7498	4	{1..20}...{61..80}
$p = 0.04$	0.6925	4	{1..20}...{61..80}	0.7493	4	{1..20}...{61..80}
$p = 0.06$	0.6992	4	{1..20}...{61..80}	0.7483	4	{1..20}...{61..80}
$p = 0.08$	0.7048	4	{1..20}...{61..80}	0.7698	8	{1..10}...{71..80}
$p = 0.10$	0.7095	4	{1..20}...{61..80}	0.7745	8	{1..10}...{71..80}

(b) L-F, $M/2$ common objects

	PS			InvKL		
	Q	C	partition	Q	C	partition
$p = 0.02$	0.3594	2	{1..40} {41..80}	0.7667	4	{1..20}...{61..80}
$p = 0.04$	0.3669	2	{1..40} {41..80}	0.7490	4	{1..20}...{61..80}
$p = 0.06$	0.3756	2	{1..40} {41..80}	0.7475	4	{1..20}...{61..80}
$p = 0.08$	0.3938	3	{1..20} {21..40} {41..80}	0.7687	8	{1..10}...{71..80}
$p = 0.10$	0.4142	3	{1..20} {21..40} {41..80}	0.7730	8	{1..10}...{71..80}

(c) F-F, 1 common object

	PS			InvKL		
	Q	C	partition	Q	C	partition
$p = 0.02$	0.5711	4	{1..20}...{61..80}	0.7498	4	{1..20}...{61..80}
$p = 0.04$	0.5146	4	{1..20}...{61..80}	0.7492	4	{1..20}...{61..80}
$p = 0.06$	0.4465	4	{1..20}...{61..80}	0.7480	4	{1..20}...{61..80}
$p = 0.08$	0.3734	4	{1..20}...{61..80}	0.7692	8	{1..10}...{71..80}
$p = 0.10$	0.3038	4	{1..20}...{61..80}	0.7730	8	{1..10}...{71..80}

(d) F-F, $M/2$ common objects

	PS			InvKL		
	Q	C	partition	Q	C	partition
$p = 0.02$	0.1103	4	{1..20}...{61..80}	0.7496	4	{1..20}...{61..80}
$p = 0.04$	0.0841	4	{1..20}...{61..80}	0.7481	4	{1..20}...{61..80}
$p = 0.06$	0.0610	4	{1..20}...{61..80}	0.7444	4	{1..20}...{61..80}
$p = 0.08$	0.0422	4	{1..20}...{61..80}	0.7611	8	{1..10}...{71..80}
$p = 0.10$	0.0485	5	{1..15}...{61..75} {16..20,36..40,56..60,76..80}	0.7549	8	{1..10}...{71..80}

Table 4.3: Resolution experiments: Modularity and communities formed, $N = 80$, $M = 20$.

(a) Similar nodes

PS			InvKL		
Q	C	partition	Q	C	partition
0.0215	2	{1..38} {39..80}	0.6740	5	{1..14} {15..28} {29..44} {45..61} {62..80}

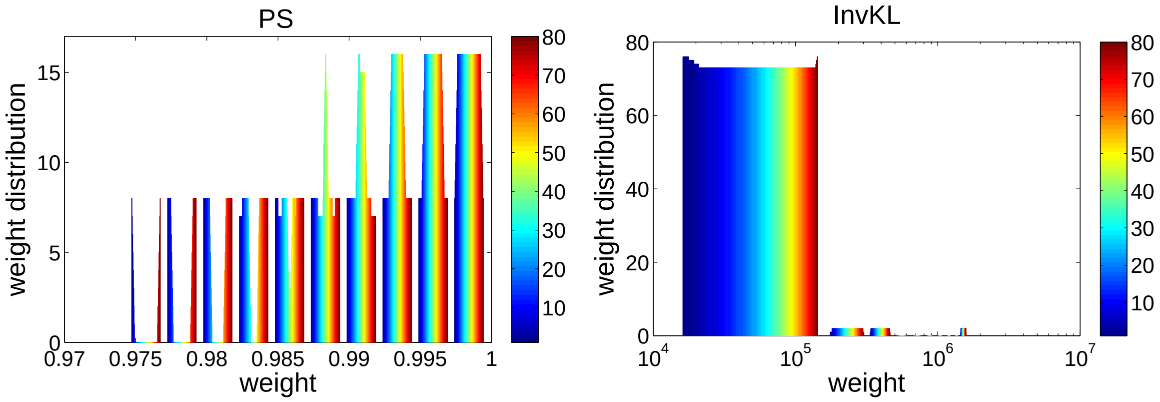
(b) Dissimilar nodes

PS			InvKL		
Q	C	partition	Q	C	partition
0.7860	10	{1..8}..{73..80}	0	1	{1..80}

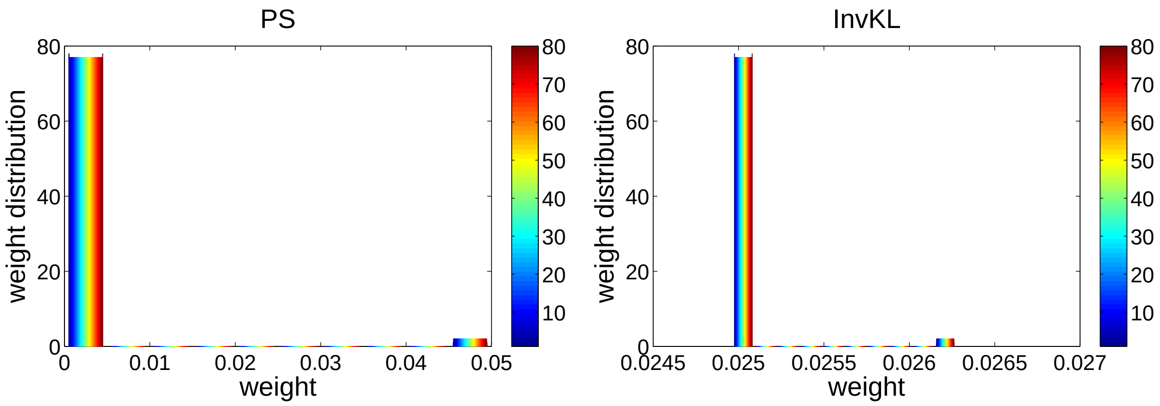
the InvKL metric are almost insensitive to the changes in the input interest distributions. With InvKL the modularity values are dominated by the edge weights between individual node pairs; these tend to be very high ($\gg 1$) for highly similar nodes and very low for highly dissimilar nodes. Finally, as p increases, the interest distributions of most nodes tend to be more concentrated on the first group objects, and the interest distributions become less uniform. For cases shown in Tables 4.2(a) and 4.2(b), this results in increasing modularity under the PS metric, thanks to the decreasing weights of inter-group edges, *i.e.*, nodes initially assigned to different groups. It has the opposite effect for cases shown in Tables 4.2(c) and 4.2(d), where increasing p leads to stronger ties between nodes in different groups. On the contrary, InvKL does not adapt to any of these changes.

Resolution experiments

We run two additional experiments focusing on the impact of the two similarity metrics upon the overall framework resolution. The first experiment involves nodes with *highly similar interests*. All nodes are interested in the same M objects, in the same order. They differentiate only slightly in how their interests are spread over the M interest classes, modelled by Zipf(s) distributions with s varying from 0 to its maximum value in steps of $p = 0.01$. The second experiment involves nodes with *highly dissimilar interests*; there is a single common interest class between successively ordered nodes. The experiment resembles the L-F overlap scenario shown in Table 4.1(a), if each group contained only one node. The results from the two experiments are reported in Table 4.3 and clearly demonstrate the capacity of the two metrics to illuminate better different parts of the interest similarity range.



(a) Similar nodes



(b) Dissimilar nodes

Figure 4.2: Resolution experiments: Edge weight distributions.

Mapping highly similar interest distributions to a much broader edge weight value range (Fig. 4.2(a)), InvKL can resolve more communities than PS in the first experiment, all of which satisfy Radicchi's weak community condition of (4.3.0.1). On the contrary, PS tends to group small communities together. Notably, the communities produced by both metrics do not satisfy the inequality

$$l_c < \sqrt{2L}, \tag{4.3.1.1}$$

which according to [Fortunato & Barthélemy, 2007] suggests that community c may be the combination of two or more smaller communities that cannot simply be detected when pursuing the optimization of modularity due to their small size.

The situation is reversed in the second experiment: it is now PS that can recognize smaller communities, as shown in Table 4.3(b). Moreover, (4.3.1.1) is satisfied, implying that there are more non-detected communities. InvKL, on the contrary, *cannot* since it squeezes all edge weight values that result from the first processing step within an interval of 0.012 width (Fig. 4.2(b)).

However, an important question is regarding the level of resolution, *i.e.*, in which cases communities should be more resolved. Intuitively, it seems more important to identify finer community structure in a network with more similar nodes, than in case of dissimilar ones. Hence, the resolution advantage of InvKL at high similarity scenarios may overweigh its disadvantage at low similarity ones.

4.4 Application to a real network

We apply our framework, *ISCoDe*, to data traces extracted from the Delicious website (www.delicious.com)[3]. Delicious is a social bookmarking application where users can save all their web bookmarks (annotated with tags) online, share them with other users, and track what other users are bookmarking themselves. Each user forms a network with other users that have subscribed to see their bookmarks. We use the organization of users into networks and interests into tags to generate the user interest distributions and feed *ISCoDe* with them.

From user interest profiles to interest distributions. Let M be the set of most popular tags used by a set of N Delicious users. Let B_m^n be the number of bookmarks tagged with m ($1 \leq m \leq M$) by user n ($1 \leq n \leq N$). Then the (normalized) interest of node n in tag m is given by the ratio of the number of bookmarks tagged with m by node n over the total number of bookmarks of this user:

$$F_m^n = \frac{B_m^n}{\sum_{m=1}^{M} B_m^n}.$$

Experimentation set-up. The Delicious network is crawled in two ways. The first method starts from four randomly selected user accounts and extracts, using a breadth-first search, 29 users that follow each one of the 4 users. Only bookmarks that contain the

[3]We thank Walter Colombo and Martin Chorley from Cardiff University for kindly providing us with traces they collected from the Delicious network.

99 most popular tags are considered for each user. The PS and InvKL metrics result in modularity values 0.0672 and 0.1130, respectively. The second procedure is similar to the first one; only now the four users are selected *among the most recent bookmarks* and the 30 tags of the highest preference values are kept for each user. The PS and InvKL metrics for this case result in modularity values 0.0754 and 0.1971, respectively.

Our results suggest that modularity values are higher in the second case, as expected, since many users are interested in the same tags. Overall, however, the modularity values of the respective users' partitions are low, implying that Delicious user networks do not display interest similarity structure. Users do not follow other users based on similarity of their tagged bookmarks. We argue that the satisfaction of users from the Delicious network would greatly increase if users formed communities taking higher account of their interests' matching. The framework proposed in this paper could be valuable in this respect.

Chapter 5

Exploiting locality and interest information to improve content dissemination

Social groups are typically formed by nodes that share common interests (interest-induced social groups), usually not considering the geographic location of these nodes. In addition to such groups, mobile nodes form groups also as they move around and come to a locality where they can establish communication with other nodes (locality-induced social groups). The exploitation of both interest-induced social groups and locality-induced social groups in order to improve information dissemination, has recently attracted the attention of researchers.

Hence, it is challenging to investigate the intermingling of these distinct types of social groups and study how content dissemination is enhanced in the presence of such groups.

5.1 Introduction

In environments where the information exchange is possible *only through opportunistic encounters*, the benefits of interest-based social grouping can be harvested only by exploiting such encounters. The higher the chance for a node to encounter other nodes that are strongly associated with a certain interest, the higher the probability for a node to acquire a certain content of interest.

Node encounters may occur in two ways:

- Locality-induced encounters: these are encounters that occur in well defined localities which nodes have a good chance of visiting, based on their behaviour. Such localities will be considered to define locality-induced (as opposed to interest-induced) social groups. Such groups may be, for example, a coffee-break place, a train station, *etc.*.

- Random encounters: these are encounters not influenced by a particular locality (physical area) in which the node is found. For example, this may include encounters a node may have when moving between different locality-induced social groups.

In this chapter, we explore how locality-induced node encounters and the nodes' own (content) interests can be jointly exploited to improve information dissemination in social networks. This has also been the focus of many researchers. For instance, a dynamic scheme for deciding which objects (content) of a certain content type to replicate locally based on the encounters with other nodes is introduced in [Boldrini *et al.*, 2008]. In that work each node appends a value to each object that is a function of its access probability and its availability in a locality, its size and the weight of the locality; this weight represents the relationship between the node and the locality (*e.g.*, how often a node visits this locality).

In addition to its membership to interest-induced social groups, a node may be attributed membership to locality-induced social groups. Such memberships are expected to boost drastically the *discoverability* of (probability of acquiring) a desirable content. That is, the locality-induced social networking structure could direct the content dissemination process to target nodes which are likely to be encountered by nodes desiring the specific content. For example, nodes that are interested in music of the 80's may be members of the interest-induced social group "music of the 80's", but they could also be members of the locality-induced social group "discotheque XYZ", where they may frequently meet and thus, exchange more contents of the same interest. Such locality-induced social groups could be useful in enhancing content dissemination. We explore how mobility and cooperation can enhance content dissemination by exploiting the joint association of nodes with interest- and locality-induced social groups.

Interest- and locality-induced social grouping may be taken into account when a node decides on which type of content to store in its memory. Clearly, the type of content that each node stores in its memory and exchanges with other nodes, shapes the content dissemination process. The effectiveness of this process can be measured in terms of a quantity (to be referred to as *valuability*) that captures both the *usability* and *discoverability* of the content; usability refers to the extent to which content is (still) useful (*e.g.*, just created vs outdated and irrelevant any more, *etc.*), while discoverability refers to the chance of succeeding in acquiring certain content by nodes that want it. This quantity will be clearly shaped by the adopted content dissemination strategy. The higher the value of this quantity, the more effective the adopted content dissemination strategy would be considered to be.

In our model, the associations of each node with the interest- and locality-induced social groups are described through probability distributions over different content types (interests) and localities, which express the likelihood of a node to be interested in a certain content-type, or to visit a certain locality. Content exchanges are assumed to occur between a node visiting a locality-induced social group and the *entire* group, as a visit to a locality implies the ability to communicate with any member of that locality; as a result, content exchanges are considered to be possible between the visiting node and other nodes in the group. The proposed content storage (and thus, dissemination) strategies operate on the content-type (or interest/content class) level and not on individual object level (as in [Boldrini *et al.*, 2008]). The proposed cooperative strategy takes into consideration the interests of the locality-induced social groups the node is likely to visit in the future, thus aiming at serving as a bridge to distinct social groups and enhance content dissemination; this seemingly *altruistic* behaviour benefits the particular node as well, as is indicated by comparing the results with the case of selfish node behaviour. The proposed strategies are evaluated analytically with respect to a newly introduced performance metric called *valuability*; this quantity captures jointly how probable a certain content-type is to be found and how useful or usable it is (its *usability*). Among other factors, the *usability* may capture how fresh or novel an object is for a certain node (*e.g.*, latest software update). Contents that reside outside a node's storage are considered to have high *usability* for this node, as such contents most likely have not been available to (or used by) that node in the past. After receiving such contents upon an encounter with another node, these contents are considered to become "old" as they are processed or utilized (if desirable) by the receiving node and, thus, their *usability* for the node that stores and carries them is considered to be decreased.

5.2 Attributes of social groups

Consider a social network that consists of N mobile nodes, C interest-induced social groups (or, equivalently, content classes) and L locality-induced social groups; small letters n, c and l will be used in the sequel to represent an element from sets N, C and L, respectively.

Network nodes are considered to belong to one or more interest-induced social groups. The degree of their association with each such group – that represents the distribution of content class preferences of the node – is captured by the node's self-interest factor. Let I_c^n, denote the self-interest factor of node n in content class c, where $0 \leqslant I_c^n \leqslant 1$ and $1 \leqslant c \leqslant C$. Let the self-interest vector of node n be the collection of all self-interest factors of this node associated with all the content classes, denoted by $I^n = (I_1^n, I_2^n, \ldots, I_C^n)$, where $\sum_{c=1}^{C} I_c^n = 1$.

In this work it is assumed that network nodes are mobile and exchange information only through encounters with other nodes. At any point of time these nodes may be found in a random location or in a non-random, well-defined and somewhat popular locality; the latter are localities that are visited by a number of nodes over certain periods and are, thus, considered to attract a locality-induced social group. Let the self-movement factor of node n, M_l^n, be equal to the probability that node n is found in locality l at a random point in time, *i.e.*, belongs to the locality-induced social group l, $0 \leqslant M_l^n \leqslant 1$. Let $M^n = (M_1^n, M_2^n, \ldots, M_L^n)$, denote the self-movement vector of node n, containing the probabilities that node n is found in the various locality-induced social groups, where $\sum_{l=1}^{L} M_l^n \triangleq 1 - r^n$. r^n is the probability not to be in any of the locality-induced social groups. Without loss of generality and in order not to burden the presentation, we will consider that node encounters occur only within the defined localities (and not at random locations) and thus, $\sum_{l=1}^{L} M_l^n = 1$ or $r^n = 0$.

Let $A^n = (I^n; M^n)$ denote the attributes of node n. As it will become clear later, the proposed cooperative content storage strategy will take into consideration certain attributes associated with the locality-induced social groups. Such attributes are defined in the sequel.

Let g_c^l denote the probability that a random member of group l belongs to content class c. This probability is given by

$$g_c^l = \sum_{n=1}^{N} P_l^n I_c^n, \tag{5.2.0.1}$$

where P_l^n denote the probability that a randomly selected node from locality-induced social group l is node n and is given by

$$P_l^n = \frac{M_l^n}{\sum_{j=1}^{N} M_l^j}.$$

Let w^l denote the weight of group l, defined to be equal to the average population of locality-induced social group l (mean number of nodes found in this group at a random inspection time), given by

$$w^l = \sum_{n=1}^{N} M_l^n. \tag{5.2.0.2}$$

Notice that w^l captures the ability of locality-induced social group l to diffuse content, as a higher value of w^l would suggest a higher potential for content dissemination within group l, since all the nodes within a locality-induced social group are considered to communicate with each other. For instance, if g_c^l is the same for all the locality-induced social groups l, it would be more effective to try to forward contents of class c to the group with the largest population. An effective content storage (and, thus, content dissemination) strategy should aim at making content of class c available to locality-induced social groups that are well-populated by nodes utilizing this class of content as well as have a relatively large population.

5.3 Content storage strategies

The above introduced framework for describing interest- and locality-induced social groups is employed in this section in order to define effective content storage strategies. These strategies will dictate the contents that the nodes will exchange upon encounters, so that (interest- and locality-induced) social grouping structure results in disseminating content effectively. Roughly speaking, an effective content dissemination strategy would increase the likelihood that a certain content type is made available to a node upon request. A specific metric for assessing the effectiveness of these strategies is presented in the next section.

In this work – and in order to facilitate the presentation of the framework and show its potential for effectiveness – we restrict the consideration to the content (or interest) classes without getting into the fine resolution of the different objects within each content class. Consequently, the objective here is to devise a strategy that offers nodes the type of contents that are likely to be requested and not specific objects from a content class. In other words, the discussion is at the content-type level and not the object level. This content-type level treatment could be almost directly applicable in an environment where there is an one to one equivalence between content-types and objects, while it is expected that the conclusions drawn here and the efficiencies achieved also apply to more refined strategies at an object level.

As the nodes are assumed here to encounter other nodes within the specific localities and not in random locations, at each visit to a locality a node is in communication range with all nodes of the locality-induced social group. Consequently, the visiting "sees" a (large) pool of contents of different types stored in the nodes of this group. By assuming the group population to be relatively large, it is reasonable that all content types are in principle available, although not at the same frequency. Thus, the visiting node can find and place in its storage (that is considered to be extremely small compared to the total storage of all the nodes in the group) any type of content (although not necessarily any object) it desires. Which content-type to actually select is dictated by the employed content storage strategy.

Let IPI_c^n denote the *interest priority index* (IPI) of a node n for content class c, to be defined below. The content storage strategies considered here use this index to determine the portion of the node's storage that will be allocated for storing content of the different content classes when the node is confronted with such a decision (upon entering a locality-induced social group). The proportion of storage each node n devotes to contents of class c is given by the normalized value of its IPI index,

$$\overline{IPI}_c^n = \frac{IPI_c^n}{\sum_{c=1}^{C} IPI_c^n}.$$

The interest priority index of node n for content class c is defined as

$$IPI_c^n = I_c^n. \tag{5.3.0.1}$$

Under the selfish strategy, the nodes store objects solely according to this index. Thus the nodes seek to store content-types that match completely their own content-type (self-interests), with no provision whatsoever for the interests of other nodes they will encounter in the future.

The second content storage strategy considered here (to be referred to as the *cooperative strategy*) utilizes the following IPI:

$$IPI_c^n = \sum_{l=1}^{L} M_l^n w^l g_c^l = \sum_{l=1}^{L} M_l^n \sum_{k=1}^{N} M_l^k I_c^k. \tag{5.3.0.2}$$

That is, under this content storage strategy the nodes seek to take into account the interests of the nodes they will most likely encounter in the future, aiming at maximizing the average benefit they can generate through such cooperative behaviour. The latter is achieved by considering the locality-induced social groups the node is expected to visit (term M_l^n), the number of nodes within those groups (term w^l), and the interests of the nodes expected to visit those groups (term g_c^l).

5.4 Measuring the effectiveness of the content dissemination strategies

As indicated earlier and in order to show the potential effectiveness of the cooperative strategy without the exploding complexities of dealing with and keeping track of a realistically immense universe of individual objects, the discussions, definitions and overall evaluation will be confined to the content-type (as opposed to individual content) level. In view of the earlier discussions it is clear that the adopted content storage strategy will shape the effectiveness of the content dissemination process. The effectiveness of this process will be measured in this work in terms of a quantity (to be referred to as *valuability*) that captures both the *discoverability* and the *usability* of the content.

Discoverability refers to the availability of a requested content-type c. This quantity will be clearly shaped by the adopted content dissemination strategy. The higher the value of this quantity, the more effective the adopted content dissemination strategy would *potentially* be, provided that the content made available to the node (discovered) is of high potential use, or usability.

Usability refers to the potential for making use of content that is, or becomes, available to a node (or ultimately, how useful a certain content is to the node). It is reasonable to assume that the content that a node finds in the storage of other nodes (upon entering a locality-induced social group) is of higher potential usage than the content that the node has been carrying in its own storage, as the latter content may be considered to have already been used by the node in the past, or be somewhat outdated.

Notice that the *discoverability* of a certain content may be high, but its *usability* be low and, consequently, a strategy that ignores the contents' *usability* would not be effective; such a strategy could end up providing contents to the nodes with high probability but these objects could be of small value to them. Clearly, if *discoverability* were the only criterion, the nodes would tend to bring to their own storage, contents of their own interests only (since the *discoverability* of contents stored locally is high), making them following in essence a "myopic" (or selfish) content storage behaviour. The latter would impact negatively on both the node itself (as it will likely end up carrying stale and largely useless content) and on a wider content dissemination (as it will not contribute to a wider circulation and refreshing of the content, benefiting all nodes). Through the introduced notion of *usability* we basically obtain a mechanism which, on one hand helps capture realistic aspects of the environment and avoid the above inefficiencies and, on the other hand, gives us a tool for steering the behaviour of the nodes across the entire spectrum, from selfish (or even "pathologically" selfish) to cooperative (or even totally altruistic), depending on the *usability* values assigned to the contents. An effective content dissemination strategy would be the strategy that yields high content *valuability*, that is, taking into consideration both the *discoverability* and *usability* metrics.

Each node is assumed to assign a *usability* value to its contents of interest. It assigns a (relatively low) value v_l $(0 \leq v_l \leq 1)$, to contents of its interest that are stored in its own storage, and it assigns a (relatively high) value v_r $(0 \leq v_r \leq 1)$, to contents of its interest that are stored in other nodes. To keep the analysis simple, it is assumed that v_l and v_r are the same for each node. The normalized *usability* values are given by $\overline{v_l} = \frac{v_l}{v_l+v_r}$ and $\overline{v_r} = \frac{v_r}{v_l+v_r}$, for v_l and v_r respectively.

The *discoverability* of a content-type can be expressed as the probability of acquiring content of this type. The probability that a node n finds contents of class c in its own (local) storage is given by

$$Pl_c^n = \overline{IPI}_c^n.$$

The probability that a node n finds contents of class c in the storage of other nodes (remote storage) is given by

$$Pr_c^n = \sum_{l=1}^{L} M_l^n [1 - \prod_{\substack{k=1, \\ k \neq n}}^{N} [M_l^k (1 - \overline{IPI}_c^k) + (1 - M_l^k)]].$$

That is, it is given by the sum for each locality-induced social group l of the probability that node n is in the group l, multiplied by the probability that at least one other node from this group has contents of class c.

The mean *valuability* of content class c for node n can now be defined by combining the above probabilities (capturing the *discoverability* of content) with the *usability* of contents, stored locally or remotely, as follows, for the case in which $\overline{v_r} \geq \overline{v_l}$:

$$V_c^n = \overline{v_r} Pr_c^n + \overline{v_l}(1 - Pr_c^n) Pl_c^n \tag{5.4.0.1}$$

When $\overline{v_r} \geq \overline{v_l}$, a node is primarily interested in acquiring contents that other nodes have (as they are considered to be more fresh and not exploited yet) with *usability* $\overline{v_r}$. Otherwise, if the contents of its interest are not found in other nodes, they are acquired from its local memory with *usability* $\overline{v_l}$. That is, when $\overline{v_r} \geq \overline{v_l}$ node n first searches for contents of class c in other nodes (of higher *usability*) and if none is found, it uses the one in its local memory.

By weighting (5.4.0.1) with the self-interest factors I_c^n of a node n for each content class $c = 1, ..., C$, the mean *valuability* of contents of its interest for node n, is derived by

$$V^n = \sum_{c=1}^{C} I_c^n V_c^n.$$

Finally, the mean *valuability* of contents of its interest for a random node, is given by

$$V = \frac{\sum_{n=1}^{N} V^n}{N}. \tag{5.4.0.2}$$

In the next section we will investigate the effectiveness of the selfish and cooperative content storage (or dissemination) strategies.

5.5 Numerical evaluation

In this section we derive results on the mean *valuability* of contents of interest for a random node (as expressed in (5.4.0.2)) under the selfish and cooperative content dissemination strategies introduced earlier. In line with the considered, the higher the value of *valuability*, the more effective the associated content dissemination strategy. The focus of the study is on cases where nodes have different preferences for content classes, so that the mutual benefit of cooperation be revealed. In all cases considered here, all nodes are either selfish or all are cooperative.

The self-interest vector $I^n = (I_1^n, I_2^n, \dots, I_C^n)$ is drawn from a Zipf distribution with exponent $s_C = 5$. The probability that each node is interested in contents of class c, $c = 1, ..., C$, which is the self-interest factor of this node in content class c, is given by

$$I_c^n = f(c; s_C, C) = \frac{1/c^{s_C}}{\sum_{k=1}^{C} 1/k^{s_C}}. \tag{5.5.0.1}$$

The drawn probabilities are the same for all nodes and they result in a preference ranking for the C content classes, which are accordingly ordered as $[1, 2, ..., C]$. Since we are interested in considering the case where each node has different ordering of preferences for each content class, node 1 is assigned the preference order $[1, 2, ..., C]$ and this order set is shifted to the left by $n - 1$ positions to generate the preference ranking for node n ($n = 2, ..., N$). Thus, if the preference ranking of node 1 is $[1, 2, ..., C]$, then the ranking of node 2 is $[2, 3, ..., C, 1]$, the ranking of node 3 is $[3, 4, ..., C, 1, 2]$, and so on.

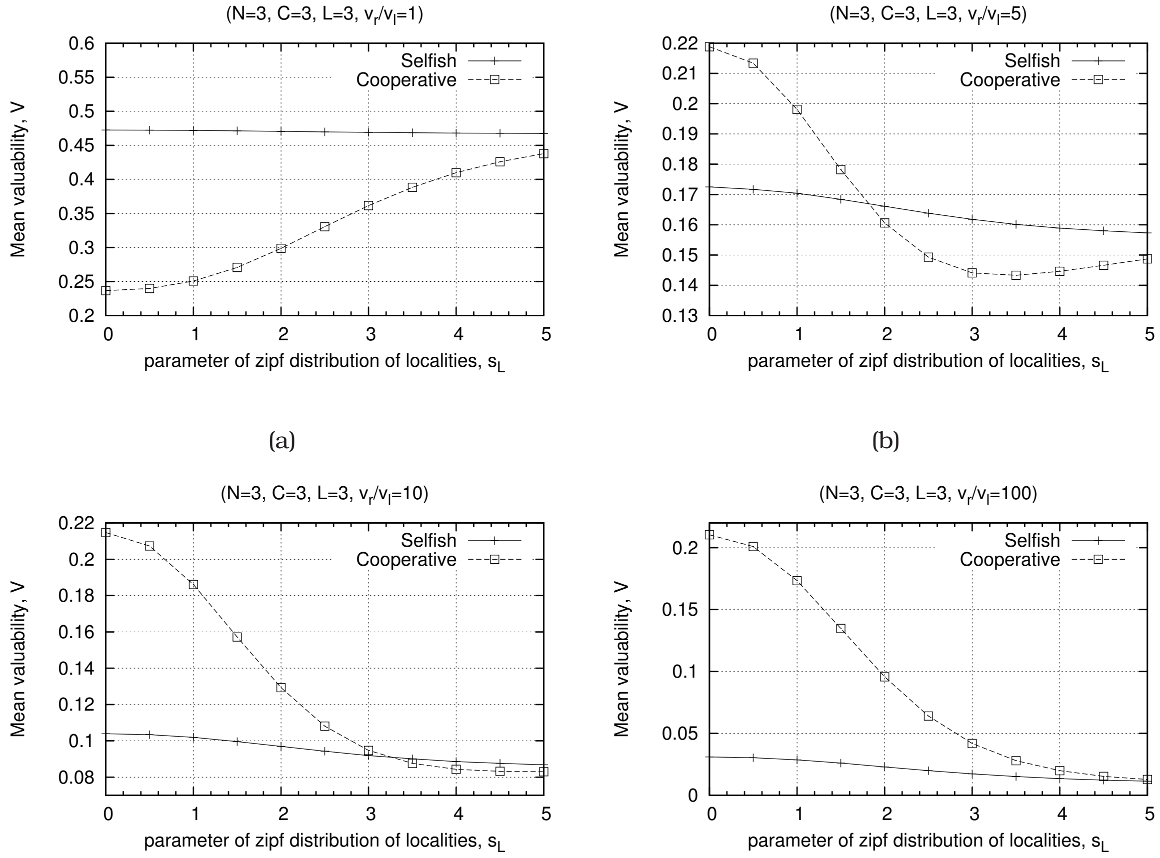
The self-movement vector $M^n = (M_1^n, M_2^n, \ldots, M_L^n)$ is drawn from a Zipf distribution (that is, from (5.5.0.1) by substituting I_c^n with M_l^n, s_C with s_L and C with L) with exponent s_L varying from $s_L = 0$ (in which case the distribution of localities is uniform and the mobility of the nodes is considered to be high as the nodes visit with the same probability all the locality-induced social groups) to $s_L = 5$ (the mobility of the nodes is considered to be low). Again, we consider the case where the preference ranking of node 1 is $[1, 2, ..., L]$ for the L localities, the ranking of node 2 is $[2, 3, ..., L, 1]$, the ranking of node 3 is $[3, 4, ..., L, 1, 2]$, and so on.

The values of g_c^l and w^l are calculated from (5.2.0.1) and (5.2.0.2), respectively.

5.5.1 Results for various self-movement vectors of a node

We first consider the case for $N = C = L = 3$. In this case and in view of the earlier assumptions on the preference ranking, it turns out that all three nodes have different largest preferences for content classes and localities (for $s_C = 5$ and $s_L > 0$). To show the impact of *usability* on the mean *valuability* of contents of interest for a random node (and, thus, on the effectiveness of the content dissemination process), we consider the cases where $\frac{v_r}{v_l} = 1, 5, 10$, or 100.

Fig. 5.1 presents results for the different values of $\frac{v_r}{v_l}$. These results show that the selfish strategy outperforms the cooperative one for any value of the parameter s_L when all the contents are considered to have the same *usability*, or equivalently, when the nodes ignore it (as shown in Fig. 5.1(a)). When mobility is high ($s_L = 0$), the cooperative strategy is very inefficient, since the nodes have different preferences for content classes and thus, the probability of finding contents of interest in other nodes is low. When the nodes have high probability of visiting certain localities only and much smaller probability of visiting others ($s_L = 5$) the performance of the cooperative strategy approaches that of the selfish. This is due to the fact that the IPI value under the cooperative strategy (as calculated from (5.3.0.2)) approaches that under the selfish strategy (as calculated from (5.3.0.1)), and thus, the probability of finding contents of interest in other nodes and locally becomes almost the same.



(a) (b)

(c) (d)

Figure 5.1: Mean *valuability* of contents of interest for a random node for different values of parameter s_L.

Fig. 5.1(b), 5.1(c) and 5.1(d) show results for various values of $\frac{v_r}{v_l} > 1$. Under high node mobility ($s_L = 0$), which can be expressed with a uniform distribution of localities ($s_L = 0$), the cooperative strategy outperforms the selfish one, clearly indicating that mobility bridges nodes with different preferences for content classes, as long as the *usability* of the remote contents is larger than that of the locally stored ones. As s_L increases (and the nodes have a higher probability of staying at a certain locality), the difference between the cooperative and the selfish strategy decreases. The cooperative strategy performs generally well, since it outperforms the selfish one for high mobility and it approaches the selfish one for very low mobility. Under medium mobility, depending on the magnitude of the ratio $\frac{v_r}{v_l} > 1$, the cooperative strategy may outperform the selfish one (for very high values of this ratio, *e.g.*, $\frac{v_r}{v_l} = 100$) or not (*e.g.*, for $\frac{v_r}{v_l} = 10$).

The mean *valuability* under the selfish strategy decreases as s_L increases, since the nodes tend to stay in different localities and thus, the probability of finding contents of interest in other nodes decreases.

As the value of $\frac{v_r}{v_l}$ increases, suggesting that contents of other nodes are more important, the cooperative strategy outperforms the selfish one under a wider ranges of values of s_L. More specifically, in Fig. 5.1(d), where $\frac{v_r}{v_l} = 100$, the cooperative strategy outperforms the selfish one for all the values of s_L as compared to the Fig. 5.1(b) and 5.1(c) where the value of $\frac{v_r}{v_l}$ is lower and thus, the cooperative strategy outperforms the selfish one only for some values of s_L. Clearly, there is a threshold of the ratio v_r/v_l above which the higher *usability* value of the remote contents more than compensates for the lower *discoverability* of the remote contents, making the *valuability* of the cooperative strategy (which stores remote contents) higher than that of the selfish one.

5.5.2 Results for identical self-movement factors

Fig. 5.1 clearly demonstrates that under high mobility ($s_L \simeq 0$), both the selfish and the cooperative strategies perform well. In addition, for a high ratio of $\frac{v_r}{v_l}$, for example $\frac{v_r}{v_l} = 100$, when $0 \leq s_L \leq 5$, the cooperative strategy always outperforms the selfish one. For this reason, we focus in this subsection on the case of high mobility; notice that in this case, the nodes have equal self-movement factors, or equivalently, a uniform distribution of locality preferences. Specifically, the case of $s_L = 0$ and $\frac{v_r}{v_l} = 100$ is considered in this subsection and the performance of the two strategies is investigated as the values of C, L and N vary; the parameter s_C is fixed at $s_C = 5$. Results are shown in Fig. 5.2.

Fig. 5.2(a) and 5.2(b) show that when each node visits each locality with the same probability, the selfish and cooperative strategies perform similarly for small values of C and L. This suggests that mobility can help in the dissemination of contents, even when nodes are selfish, provided that the number of content and localities is small.

Fig. 5.2(a) results for the case of $N = 10$, $L = 3$ and C varying from 1 to 10. When there is only one content class ($C = 1$), V is the same for both strategies and almost equal to 1. This result is anticipated, since each node has contents of only one class and thus, a cooperative behaviour would also serve a node's own interests. As C increases the difference between selfish and cooperative strategies becomes more pronounced, with

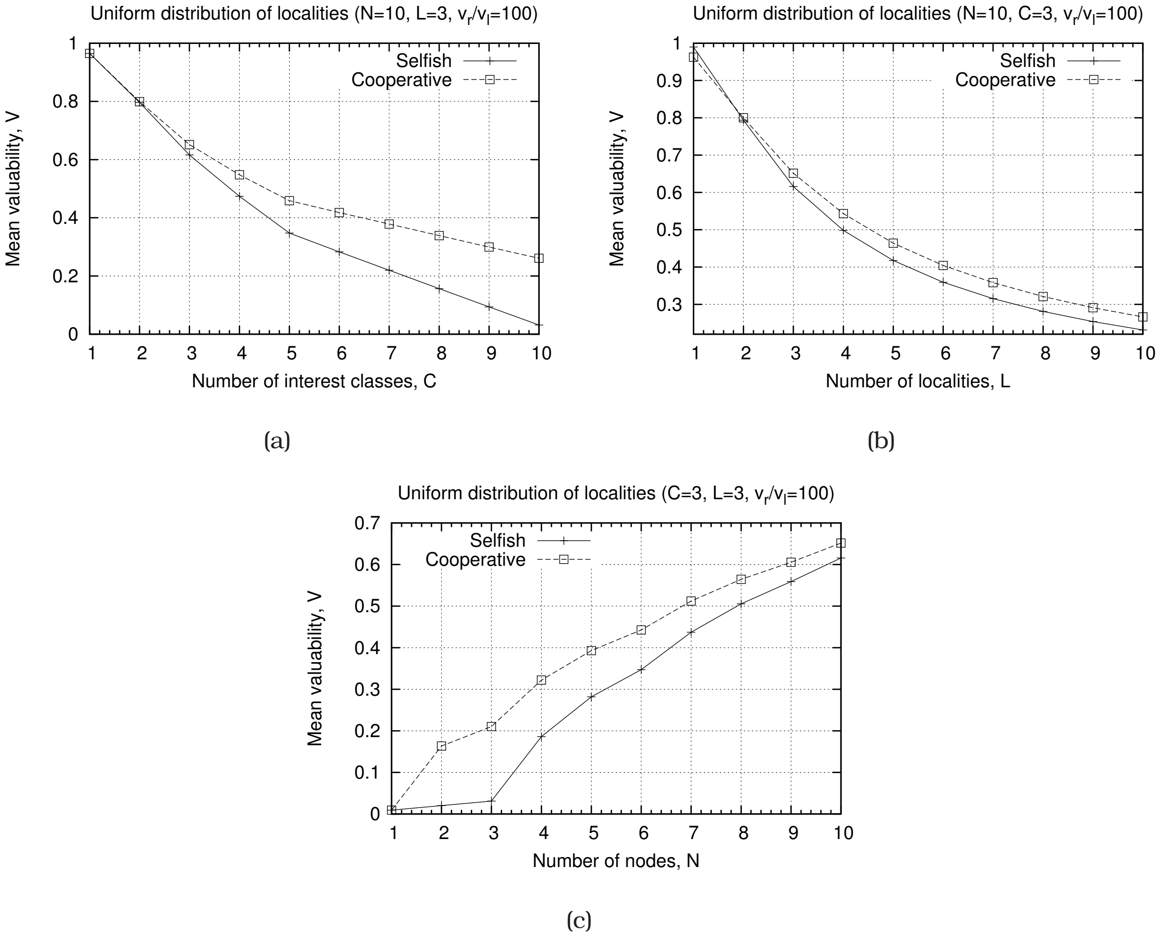


Figure 5.2: Mean *valuability* of contents of interest for a random node for uniform distribution of localities.

the latter outperforming the former. This shows that cooperation is more beneficial than selfishness if there exist more than one content classes. Moreover, V decreases for both the selfish and cooperative strategies as C increases. This is attributed to the fact that the nodes have different preference rankings for content classes and, thus, the more the content classes, the lower the probability for a node of finding contents of its interest in other nodes, and thus, the lower the value of V.

Fig. 5.2(b) shows the same mean *valuability* V as a function of the total number of locality-induced social groups L, which varies from 1 to 10. There are $N = 10$ nodes and $C = 3$ content classes. Since each node n ($1 \leq n \leq 10$) has a different preference ranking for the three content classes, by shifting the preference order set $[1, 2, 3]$ to the left by $n - 1$ positions, there would be four nodes with the same preference ranking $[1, 2, 3]$, three nodes with the ranking [2,3,1], and three other nodes with the ranking $[3, 1, 2]$. If there is only one locality ($L = 1$), a selfish behaviour would be better than a cooperative one, since in the same locality it is highly probable that there can be other nodes with the same preference ranking for the content classes. However, as L increases, cooperation clearly outperforms selfishness.

Fig. 5.2(c) shows V as a function of the total number of nodes, N. It is anticipated that as N increases, the mean *valuability* V increases as well, since the probability of finding contents of interest in other nodes increases. Also, as $N \leq C$ the selfish strategy for all nodes is much worse than the cooperative one. That is due to the fact that because the nodes' preference rankings for content classes do not coincide and thus, selfishness is in such a case detrimental to content dissemination. As N increases (for $N > C$) – keeping the same number of localities L and content classes C – the difference in performance between the cooperative and selfish strategies is much smaller, since there are nodes in the same localities with the same preference ranking for the content classes.

Chapter 6

Cooperative content retrieval in nomadic sensor networks

A nomadic sensor network consists of: a) sensor nodes, that are fixed at some points and collect information about states or variables of the environment, and b) mobile nodes that collect and disseminate this information. Mobile nodes usually belong to different classes, and are thus interested in different subsets of sensor node information. In such networks, dissemination of information content at smaller costs can be achieved if mobile nodes are cooperative and collect and carry information not only in their own interest, but also in the interest of other mobile nodes. A specific modeling scenario is considered here where the network has the form of a graph; sensor nodes are located on the vertices of the graph and mobile nodes move along the edges according to a random waypoint model. We present a game-theoretic analysis to find conditions under which a cooperative equilibrium can be sustained.

6.1 Introduction

A *nomadic sensor network* is a networking paradigm that was recently introduced in [Carreras *et al.*, 2005]. It consists of: a) simple, tiny sensor devices (T-nodes) fixed at some points, whose purpose is to collect information about states or variables of the environment and b) more complex mobile devices that are carried by users (U-nodes), that collect and disseminate this information. Compared to traditional sensor networks where communication to end-users is realized in a multi-hop fashion, this paradigm exploits user mobility to conserve limited sensor energy, prolonging the lifetime of the network and making it more cost-efficient.

A U-node can collect sensor data either from the source T-node or from an encountered U-node who has previously acquired the data. Collecting data from other U-nodes may incur a smaller access cost (in time or energy), especially if the source T-node is far-away. Exploiting mobility in this way to reduce content retrieval costs requires each U-node to show some kind of cooperative behavior. However, U-nodes are highly autonomous and intelligent devices; different U-nodes may belong to different classes, and thus be interested in different sensor node information. Acquiring unwanted information incurs a cost in time or energy, thus a U-node would be cooperative and collect information of potential interest to other U-nodes only in anticipation of the same behavior by other U-nodes. The purpose of this work is to identify requirements and conditions under which cooperative behavior may emerge in such a network. We present a game-theoretic analysis to find conditions under which a cooperative equilibrium exists.

Previous applications of game-theoretic methods in examining cooperation between mobile nodes have focused mainly on traditional ad-hoc networks, where cooperation consists of each node acting as a relay and forwarding packets of other nodes, at the expense of an increased processing and energy cost. This kind of cooperation is the main subject of the papers in [Srinivasan *et al.*, 2003, Crowcroft *et al.*, 2003, Seredynski *et al.*, 2006, Félegyházi *et al.*, 2006].

A work with a similar subject to ours is [Buttyán *et al.*, 2007]. Therein, the authors consider a general delay-tolerant network where information is disseminated in a store-carry-and-forward manner. The considered model is very similar to ours: information generating nodes are static, and mobile nodes are either interested in certain information or not, but may collect it anyway in order to exchange it with information of interest (barter exchange). The authors additionally consider depreciation of information content over time. The objective for each node is to decide which messages to collect from the information-generating nodes, based on their actual value, or the value they could have as a trade object (barter value). The authors find, by the means of simulations, an equilibrium strategy. In contrast to the above-mentioned approach, here we have analytically

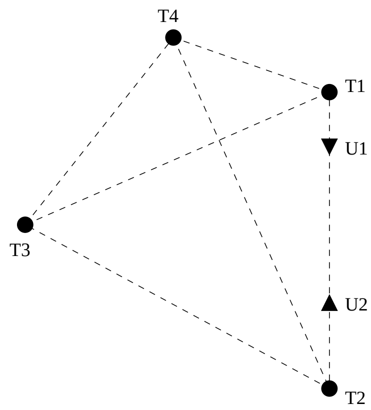


Figure 6.1: Network graph.

derived an equilibrium strategy and conditions considering the mobility of nodes in the network.

Finally, it is worth mentioning that game theory has been applied in several other problems in wireless networks, such as multiple access schemes or conflicts between transmitters-jammers. A good overview can be found in [Félegyházi & Hubaux, 2006].

The general modeling scenario that we study is as follows. We consider a graph model of the network (see Fig. 6.1). T-nodes are located at the vertices of the graph and U-nodes move randomly along the edges of the graph collecting information when they reach a T-node or upon meeting another U-node somewhere on the graph. Nodes move according to a Markovian waypoint model on the graph, with a constant speed v. Waypoints are set to be the vertices of the graph (T-nodes). No pause times at waypoints are considered. Transitions from one waypoint to the next are governed by a Markov chain, *i.e.*, a U-node moves from waypoint T_i to waypoint T_j with probability $p(T_i, T_j)$. This is a special case of the so-called "space graph" model in [Le Boudec & Vojnovic, 2005].

In order to simplify the analysis, instead of studying the network as a whole, we decompose the problem by studying possibilities of cooperation on each leg (edge) of the graph. We say that two U-nodes coming from opposite directions and meeting somewhere on a leg (T_i, T_j) are cooperative on (T_i, T_j), if and only if each one copies the content of its origin T-node (*e.g.*, in Fig. 6.1, U_1 would copy T_1 and U_2 would copy T_2), even if they are not interested in it. We shall find conditions under which the following strategy of each U-node results in an equilibrium. This strategy is made up of two actions: *Initially, a U-node is cooperative and copies unwanted content. However, if it meets a selfish U-node* *somewhere on a leg, it will only transmit its acquired content with a certain probability.* This strategy may easily be applied, provided that upon meeting each other, U-nodes exchange messages that contain the list of information objects stored in their memory.

The equilibrium conditions depend mainly on the probability of meeting other U-nodes on the leg. Under the setting we discuss in this work, satisfying equilibrium conditions for each leg and between each pair of U-nodes means that a cooperative equilibrium is achieved for the whole network. Note that in making this decomposition, we are implicitly assuming that U-nodes base their decision only on their interest for content that is in distance of one leg. The analysis becomes more complicated when this "decision horizon" is more than one, and is not attempted here. (For example in Fig. 6.1, if node U_1 that starts from T_1 is interested in both T_2 and T_3 and is about to follow the path $\{T_1, T_2, T_3\}$, it should consider the probability of meeting U-nodes that previously acquired data from T_2 or T_3, on all segments of this path.)

Upon reaching a T-node in whose content it is not interested and after selecting its next destination, a U-node must decide whether to copy the content or not, *i.e.*, to be cooperative to other U-nodes, or not. The following parameters play a major role in the game:

- *The storage capacity of U-nodes.* U-nodes may either have infinite (in practise this means sufficiently large) capacity, or finite storage space. If U-nodes have finite storage space, then they must consider possible replacement policies (for example, throwing away their less valuable content) and respective costs.

- *The depreciation of information content in time or space.* Depending on its nature, information content stored in T-nodes may be either depreciated in time or space, or not. For instance, statistical samples (*e.g.*, measurements of physical quantities) can be considered as data not depreciated in time or space. On the other hand, information with limited temporal or spatial scope, or software modules that are subject to updates are depreciated in time or space. If information content is depreciated in time or space, a U-node must include in its decision the (possibly subjective) value of each information object at the moment of its acquisition.

- *The decision horizon.* A U-node may base its decision on its interest for content that is in distance of one or more legs. This is called the decision horizon. For example in Fig. 6.1, if a U-node that started from T_1 is about to follow the path $\{T_1, T_2, T_3\}$ and the decision horizon is 2, it should consider its cost of retrieving information from both T_2 and T_3 and the possibility of another U-node to carry information from T_2 and T_3 on this path.

On what concerns the knowledge each U-node has, the following assumptions are made. Each U-node knows the topology of the network, the number of other U-nodes and the distances between each pair of T-nodes. They are also aware of the information content at each T-node. Furthermore, U-nodes know that other U-nodes move also according to the random waypoint model on the graph. On the other hand, the (instantaneous) rate at which the information content on each T-node is updated is a function of time unknown to the U-nodes. (Thus a U-node may return to an already visited T-node to get updated information.) In addition, each U-node does not know the interests of other U-nodes for sensor data and has no memory of previous encounters with them.

Several hypotheses that facilitate the analysis are also made. First it is assumed that U-nodes are homogeneous devices, have the same processing and communication costs and have the same movement parameters. Secondly, U-nodes have infinite (in practice this means sufficiently large) storage space. Thus they do not have to consider possible replacement policies (for example, throwing away their less valuable content) and respective costs. Thirdly, that each T-node generates a single type of information content, and information content is not depreciated in time or space. If it was depreciated, a U-node should have to include in its decision the (possibly subjective) value of each information object at the moment of its acquisition. Examples of non-depreciated content include statistical samples (*e.g.*, measurements of physical quantities). On the other hand, information with limited temporal or spatial scope, or software modules that are subject to updates are considered to be depreciating in time or space. We also consider that U-nodes make decisions for their best interest, but are not malicious, *i.e.*, they don't perform actions from which they have no material gain, only to hurt others. Finally, we assume that data exchanged between U-nodes or between a U and T-node consist only of a few bits; thus they are transmitted within an infinitesimal time interval, which is not considered in the analysis.

6.2 Analytical Model

In the analysis that follows, we consider a number N of U-nodes, and calculate the expected cost for a U-node to follow a certain strategy on an arbitrary leg (T_i, T_j). Because of the symmetry of our model, the cost is the same for all U-nodes. A strategy consists of a sequence of actions, concerning the decisions to collect, carry and transmit information that is not of interest to a U-node. Actions are either of a cooperative or selfish nature. A strategy is itself called cooperative if all the actions it is composed of are cooperative.

We consider that the game is played between a U-node starting from the origin sensor T_i and $N-1$ other U-nodes it may meet before reaching the destination sensor T_j. Our working hypothesis is that a U-node is not interested in the content of the origin sensor, but only in that of the destination. A priori, it assumes the same for the other U-nodes. This will produce conditions for cooperation in a worst-case scenario, since otherwise if a U-node is also interested in the content of the origin sensor, it has greater incentive to cooperate. The analogous assumption for the other U-nodes can be partially justified by the lack of any information about the identities of the encountered nodes and their interests for sensor information.

Costs can be expressed in time or energy units. Here we interpret this cost as the delay to retrieve information content. The length of a leg (T_i, T_j) is denoted by $d(T_i, T_j)$. The communication and processing cost of a U-node to acquire content from a T-node and then transmit it to a U-node is a constant c, translated in time units. Consistently with our assumption of infinite storage, we do not consider any cost for carrying the information. Finally, not to complicate the analysis, the transmission ranges of both U-nodes and T-nodes are set to be zero.

Consider the process $X(t)$ of the position of the U-node on the graph at each time instant t. We find its stationary distribution as follows. First consider the embedded Markov chain of waypoints visited sequentially by a U-node. Under the stationary distribution of this chain, the fraction of transitions to waypoint T_i is denoted by π_i. Then if we have constant velocity, the probability that the U-node is on any segment of length x on leg (T_i, T_j) in direction from T_i to T_j is

$$\frac{\pi_i p(T_i, T_j) x}{\sum_{T_i} \sum_{T_j \neq T_i} \pi_i p(T_i, T_j) d(T_i, T_j)} \, . \tag{6.2.0.1}$$

That is, it is the fraction of time the U-node spends on the segment of length x while moving from T_i to T_j.

Suppose that the U-node, hereafter called U_i, is at waypoint T_i at $t = 0$ and decides to go towards waypoint T_j. Confine the strategy of each player to take values in the set $\mathcal{S} = \{C, S\}$ (C: cooperative, S: selfish). By choosing strategy C, U_i copies, carries and transmits the content of T_i to an encountered U-node that is interested in it, whereas by following strategy S it ignores it. In order to calculate the expected cost by following a certain strategy, U_i should estimate the probability of meeting another cooperative U-node coming from T_j towards T_i, at a certain distance x from T_i. Suppose there are k other $(k < N)$ cooperative U-nodes. To meet another U-node within a distance x, the latter must be at a distance of at most $2x$ at $t = 0$ and be headed towards T_i. Since the move processes of the U-nodes are mutually independent as well as jointly stationary, the instant $t = 0$ is an arbitrary instant at which U_i at T_i observes the position of the other U-nodes. Therefore it observes the other U-nodes in their stationary distribution. Consequently, if $d(T_i, T_j) \geq 2x$ the probability of a meeting with at least one cooperative U-node within a distance x is

$$F_k(x) = 1 - (1 - \frac{\pi_j p(T_j, T_i) 2x}{\sum_{T_i} \sum_{T_j \neq T_i} \pi_i p(T_i, T_j) d(T_i, T_j)})^k \, . \tag{6.2.0.2}$$

If $d(T_i, T_j) < 2x$, we must also include the event that another U-node is on a leg or direction different from (T_j, T_i) at $t = 0$ but can meet with U_i in time less than x/v. Consider all such legs m and the starting points T_m of the U-nodes on these legs. Let $\{T_m, \ldots, T_j\}$ denote all the paths that may be followed to reach T_j before the meeting, and for which $p(T_m, \cdot) \cdots p(\cdot, T_j) > 0$, where $\cdot$ denote intermediate states. Then this meeting probability equals

$$1 - (1 - \frac{\sum_m \pi_m p(T_m, \cdot) \cdots p(\cdot, T_j) p(T_j, T_i) 2x}{\sum_{T_i} \sum_{T_j \neq T_i} \pi_i p(T_i, T_j) d(T_i, T_j)})^k \, .$$

The summation in the numerator is over all possible paths that can be followed. This probability again equals (6.2.0.2), since $\sum_k \pi_k p(T_k, \cdot) \cdots p(\cdot, T_j) = \pi_j$ from the balance equations in the embedded Markov chain.

Hence the distribution function $F_k(x)$ gives the probability that a meeting with another cooperative U-node takes place at distance $\leq x$, for any x such that $0 < x < d(T_i, T_j)$.

Suppose that there are only two nodes in the network, U_i and U_j, and that U_i meets with U_j at distance x from T_i ($x < d(T_i, T_j)$). If they follow strategies s_i, s_j respectively ($s_i, s_j \in \mathcal{S}$), then the cost of U_i, denoted as a function $\mathcal{C}_i^{(T_i,T_j)}(s_i, s_j, x)$ of U_i, where $\mathcal{C}_i^{(T_i,T_j)} : \mathcal{S} \times \mathcal{S} \times [0, d(T_i, T_j)] \to R$, is

$$\begin{aligned}
\mathcal{C}_i^{(T_i,T_j)}(C, C, x) &= c + x/v \\
\mathcal{C}_i^{(T_i,T_j)}(S, C, x) &= x/v \\
\mathcal{C}_i^{(T_i,T_j)}(C, S, x) &= c + d(T_i, T_j)/v \\
\mathcal{C}_i^{(T_i,T_j)}(S, S, x) &= d(T_i, T_j)/v \ .
\end{aligned} \tag{6.2.0.3}$$

We denote by $\alpha(T_j, T_i)$ the probability that U_i meets with a *specific* other U-node before meeting T_j, *i.e.*, :

$$\alpha(T_j, T_i) \triangleq \frac{2\pi_j p(T_j, T_i) d(T_i, T_j)}{\sum_{T_i} \sum_{T_j \neq T_i} \pi_i p(T_i, T_j) d(T_i, T_j)} \ . \tag{6.2.0.4}$$

As it can be seen, this meeting probability increases with the length of a leg. Additionally, U_i has a higher meeting probability for greater $\pi_j p(T_j, T_i)$, that is if the destination node T_j is more frequently visited, or if U-nodes at T_j have an increased probability of heading towards T_i.

We will derive the expected cost of U_i to take action s_i, when k other U-nodes are cooperative ($0 \leq k \leq N-1$). We denote this as $\mathcal{C}_i^{(T_i,T_j)}(s_i|k)$. For $1 \leq k \leq N-1$, we have that

$$\begin{aligned}
\mathcal{C}_i^{(T_i,T_j)}(s_i|k) = &\int_0^{d(T_i,T_j)} \mathcal{C}_i^{(T_i,T_j)}(s_i, C, x) dF_k(x) \\
&+ (1 - \alpha(T_j, T_i))^k \mathcal{C}_i^{(T_i,T_j)}(s_i, C, d(T_i, T_j)) \ .
\end{aligned} \tag{6.2.0.5}$$

We arrive at the following expressions for different combinations of followed strategies:

$$\begin{aligned}
\mathcal{C}_i^{(T_i,T_j)}(C|k) &= c + \frac{d(T_i, T_j)}{v} \frac{1 - (1 - \alpha(T_j, T_i))^k}{(k+1)\alpha(T_j, T_i)} \\
\mathcal{C}_i^{(T_i,T_j)}(S|k) &= \frac{d(T_i, T_j)}{v} \frac{1 - (1 - \alpha(T_j, T_i))^k}{(k+1)\alpha(T_j, T_i)} \\
\mathcal{C}_i^{(T_i,T_j)}(C|0) &= c + d(T_i, T_j)/v \\
\mathcal{C}_i^{(T_i,T_j)}(S|0) &= d(T_i, T_j)/v \ .
\end{aligned} \tag{6.2.0.6}$$

6.3 Game-Theoretic Analysis

We observe from (6.2.0.6) that $\mathcal{C}_i^{(T_i,T_j)}(C|k) > \mathcal{C}_i^{(T_i,T_j)}(S|k) \quad \forall i, k$. Furthermore it can be shown that $\mathcal{C}_i^{(T_i,T_j)}(C|k)$, $\mathcal{C}_i^{(T_i,T_j)}(S|k)$ are decreasing functions of k. The game is a Bayesian analog of the N-person prisoner's dilemma (see [Nishihara, 1997]). (Since U_i does not know the number or identity of other U-nodes it may meet on (T_i, T_j).) When seen as a noncooperative game, it is evident from the above expressions that there exists only one equilibrium, in which every node is selfish. (Since S strongly dominates C for every player.) However, it may not be the best solution; player U_i can have a benefit by cooperating on (T_i, T_j), when k other U-nodes are cooperative, if

$$\mathcal{C}_i^{(T_i,T_j)}(C|k) < \mathcal{C}_i^{(T_i,T_j)}(S|0) \; . \tag{6.3.0.1}$$

That is, if the cost for U_i when k other U-nodes are cooperative is smaller than its cost when all U-nodes are selfish. This condition also complies with individual rationality of U_i, since $\mathcal{C}_i^{(T_i,T_j)}(S|0)$ is the minimax value U_i can guarantee for itself.

From (6.2.0.6), this inequality is satisfied if

$$1 - \frac{1-(1-\alpha(T_j,T_i))^k}{(k+1)\alpha(T_j,T_i)} > \frac{cv}{d(T_i,T_j)} \; . \tag{6.3.0.2}$$

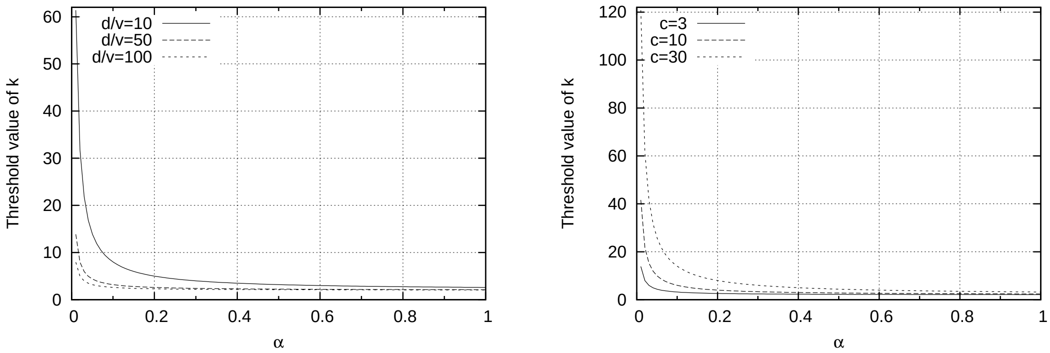
It must always hold that $c < \frac{d(T_i,T_j)}{v}$; that is, our initial assumption must be that the cost (expressed in time units) for a U-node to acquire and transmit unwanted content must be smaller than the time to reach T_j starting from T_i.

We find an approximate condition for the above inequality to be satisfied. The Taylor polynomial of $(1-\alpha)^k$ at $\alpha = 0$ is, up to a second order approximation, $1 - k\alpha + \frac{k(k-1)\alpha^2}{2}$ ($k > 1$). Substituting this approximate expression in inequality (6.3.0.2), we get the condition

$$k > \frac{2cv}{d(T_i,T_j)\alpha(T_j,T_i)} - \frac{2(1+\alpha(T_j,T_i))}{\alpha(T_j,T_i)(k+1)} + 2 \; ,$$

which is satisfied when

$$k > \frac{2cv}{d(T_i,T_j)\alpha(T_j,T_i)} + 2 \; . \tag{6.3.0.3}$$



(a) $c = 3$ (b) $d/v = 50$

Figure 6.2: Approximate threshold values of the number of other cooperative nodes k above which cooperation on a directed leg is better for U_i than full selfishness, for different values of α and d/v, c.

Based on (6.3.0.3), in Fig. 6.2 we show approximate threshold values of the number of other cooperative nodes k above which cooperation for U_i on (T_i, T_j) is more beneficial than the case when all U-nodes are selfish.

These confirm that from the point of view of U_i, a smaller number of cooperative nodes is required on long-distance routes or when the destination node T_j is visited more often. On the other hand, a higher number is required when U-nodes are moving at a high speed or if the cost c is higher.

The situation in which each U-node is cooperative on both directions of a leg (T_i, T_j) will be identified as "full cooperation" on (T_i, T_j). The inverse situation in which each U-node is selfish, will be called "full selfishness". When full cooperation is achieved as a strategic equilibrium in a leg of the graph, then we will say that a cooperative equilibrium exists on this leg. If this can be achieved on all legs, then a cooperative equilibrium exists in the whole network. We next proceed to find a strategy of each U-node and conditions under which a cooperative equilibrium can be achieved.

First pay attention to the fact that even if full cooperation is beneficial for U-nodes in the network, this is not enough to sustain an equilibrium. For an equilibrium to exist, there must either be some punishment to selfish nodes or some form of contract, in which all U-nodes would agree to be cooperative, threatening to be selfish if such contract is not signed by everyone [Myerson, 1997]. Given that the arrival to such an agreement is difficult in an unstructured network with autonomous nodes, we consider the following scheme that is easily applicable.

A requirement of the scheme is that U-nodes, upon meeting each other, first exchange the lists of information objects in their memory. These lists contain metadata regarding the type of information and the source T-node. Then each U-node executes this strategy: initially it is generous and collects and carries unwanted content; however if on a certain leg of the network it meets a selfish U-node, it will only transmit this content with a probability p, called the cooperation probability ($p < 1$). (If a U-node does not communicate its list of objects, it can be considered selfish and the same strategy applies.)

We proceed to write a condition under which this strategy is preferable for U_i on (T_i, T_j), and thus can lead to an equilibrium. Given that there are $\binom{N-1}{k}$ different combinations of U-nodes where exactly k other U-nodes are cooperative, this condition is

$$\mathcal{C}_i^{(T_i,T_j)}(C|N-1) \leq \sum_{k=0}^{N-1} p^k (1-p)^{N-k-1} \binom{N-1}{k} \mathcal{C}_i^{(T_i,T_j)}(S|k) \ . \tag{6.3.0.4}$$

That is, the expected cost for U_i when all U-nodes are cooperative must be smaller or equal to the expected cost when U_i is selfish and k other U-nodes are cooperative with probability p, $k = 0, \ldots, N-1$. It is evident that (6.3.0.1) is a special case of this condition, where $p = 0$ and $k = 0$ (admitting $0^0 = 1$).

Substituting from (6.2.0.6), we have that (in the following we omit the parameters in $d(\cdot)$, $\alpha(\cdot)$ for notational convenience)

$$\begin{aligned} c+\frac{d}{v}\frac{1-(1-\alpha)^{N-1}}{N\alpha} \leq & \ \frac{d}{v}\Big\{(1-p)^{N-1} \\ & + \sum_{k=1}^{N-1} p^k (1-p)^{N-1-k} \binom{N-1}{k} \frac{1-(1-\alpha)^k}{(k+1)\alpha}\Big\} \ . \end{aligned} \tag{6.3.0.5}$$

The right-hand side (rhs) of this inequality becomes

$$\begin{aligned} & \frac{d}{v}(1-p)^{N-1}\Big\{1+\frac{1}{a}\Big[\sum_{k=1}^{N-1}\Big(\frac{p}{1-p}\Big)^k \binom{N-1}{k}\frac{1}{k+1} \\ & - \sum_{k=1}^{N-1}\Big(\frac{p(1-\alpha)}{1-p}\Big)^k \binom{N-1}{k}\Big]\Big\} \ . \end{aligned}$$

It can be derived that $\sum_{k=1}^{N-1}\left(\frac{p}{1-p}\right)^k \binom{N-1}{k}\frac{1}{k+1} = \sum_{k=1}^{N-1}\left(\frac{p}{1-p}\right)^k \frac{1}{N}\binom{N}{k+1} = \frac{1}{Np(1-p)^{N-1}} - \frac{1-p}{Np} - 1$

and $\sum_{k=1}^{N-1}\left(\frac{p(1-\alpha)}{1-p}\right)^k\binom{N-1}{k}=\left(\frac{1-\alpha p}{1-p}\right)^{N-1}-1$. Therefore (6.3.0.5) becomes

$$c+\frac{d}{v}\frac{1-(1-\alpha)^{N-1}}{N\alpha}\leq\frac{d}{v}(1-p)^{N-1}\Big\{1+\frac{1}{\alpha}\Big[\frac{1}{Np(1-p)^{N-1}}-\frac{1-p}{Np}-\left(\frac{1-\alpha p}{1-p}\right)^{N-1}\Big]\Big\}\,.$$

Applying the second order Taylor polynomial approximation to $(1-\alpha)^N$ at $\alpha=0$, we finally obtain the condition

$$N\geq 1+\frac{2}{\alpha}+\frac{2(1-\alpha)}{\alpha}\Big\{\frac{1}{\alpha N}-(1-p)^{N-1}-\frac{1}{aNp}\Big[1-(1-p)^N\Big]+(1-\alpha p)^{N-1}+\frac{cv}{d}\Big\}\,. \tag{6.3.0.6}$$

In Fig. 6.3 we show graphically the required number of nodes in the network that would satisfy this condition, for different values of the meeting probability α and d/v. (We draw the rhs of (6.3.0.6), called f_{rhs} and the diagonal N, called f_{lhs}.) It can be deduced from the graphs that a cooperative equilibrium on a leg is easier to achieve (*i.e.*, we need a smaller number of U-nodes) for smaller values of c and larger values of d/v. The behavior with respect to α is less intuitive: when the cooperation probability of other U-nodes is small, a higher meeting probability leads a U-node actually having less incentive to cooperate and appearing more selfish, whereas if the cooperation probability of other U-nodes is high, it leads a U-node to actually be cooperative. Finally, as the probability of cooperation decreases, a smaller number N of nodes in the network is required for an equilibrium to be achieved. We can expect this result, since a cooperative U-node "punishes" more a selfish U-node by giving its acquired content with a smaller probability. Therefore U-nodes refrain from being selfish.

Note that in the model we developed we do not have to consider a "stricter" condition for a cooperative equilibrium to exist (*i.e.*, a condition that would call for a higher N). This is because we have assumed in the beginning that each U-node thinks, a priori, that all the other U-nodes are interested in the content it collects; thus we have excluded the case where a U-node would not collect unwanted information because it might think that other U-nodes would also not be interested in it (and hence also might not collect data at their respective origin points). Therefore if such cooperative equilibrium conditions are satisfied in all directed legs simultaneously, a cooperative equilibrium exists in the whole network.

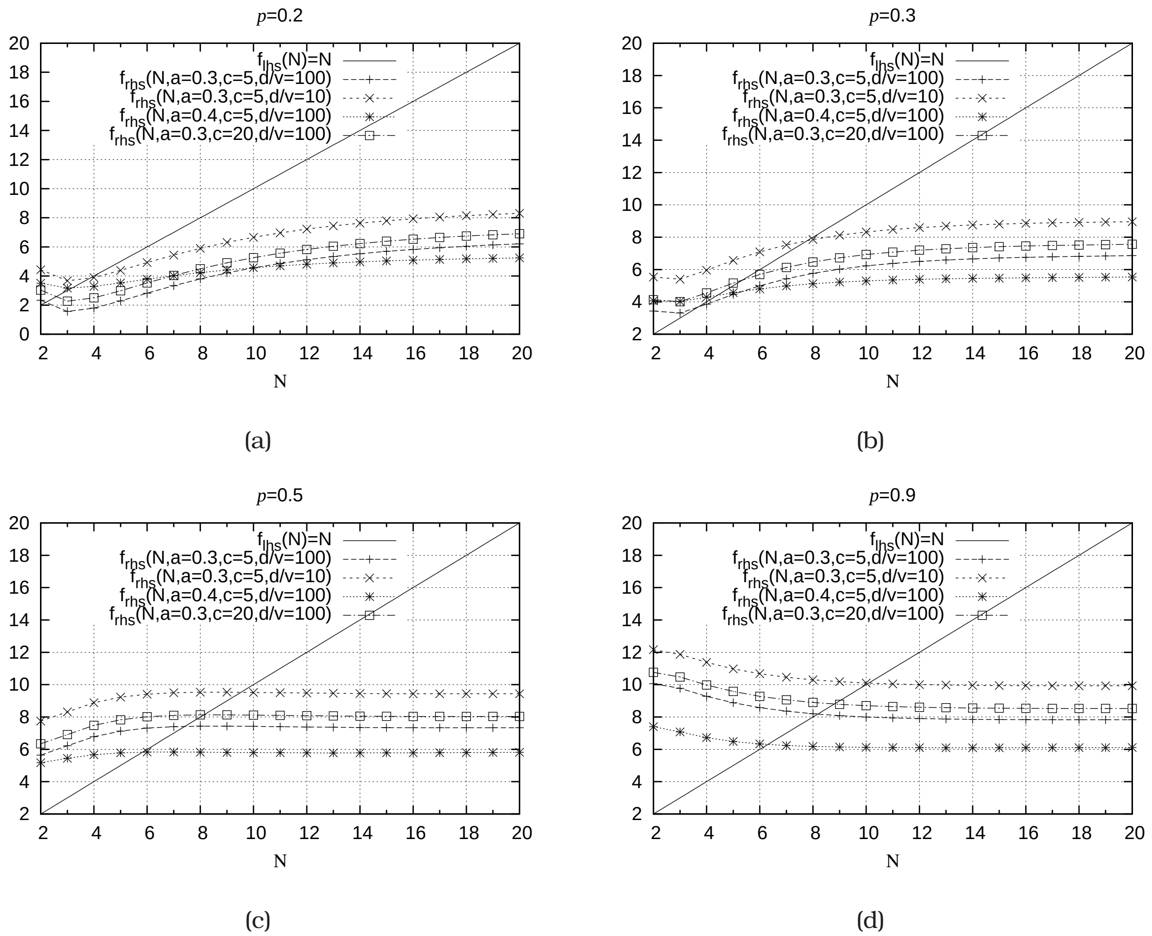


Figure 6.3: Graphical illustration of cooperative equilibrium conditions, for different values of the cooperation probability p and model parameters: the crossover points of f_{rhs} with the diagonal N show approximate threshold values of N above which a U-node is cooperative in a directed leg in our model.

6.4 Extension of the analysis: Variable content utility

We have also extended the analysis presented in the previous sections to account for the utility each node attains from content retrieval, which depends on the distance from the content's source. Specifically, we will assume that the closer a node is to the content's source, the higher the utility attained from it (since the content may be more fresh, or geographically more relevant). This is traded-off with the cost for the node to travel close to the destination, as in the previous sections.

Again we consider what happens when two nodes U_i and U_j meet on a leg (T_i, T_j)

of the network, which has a length $d(T_i, T_j)$. Suppose that the two nodes meet at a normalized distance x from T_i (*i.e.*, $x \in [0, 1]$). The cost is assumed to increase linearly with the normalized distance x, raised by a constant factor if a node is cooperative (*i.e.*, retrieves content from T_i). The utility function is denoted by $U(x)$. If the nodes U_i and U_j follow strategies s_i, s_j respectively ($s_i, s_j \in \mathcal{S}$), then the cost of U_i, denoted as a function $\mathcal{C}_i^{(T_i,T_j)}(s_i, s_j, x)$ of U_i, where $\mathcal{C}_i^{(T_i,T_j)} : \mathcal{S} \times \mathcal{S} \times [0, 1] \to R$, is

$$\begin{aligned}
\mathcal{C}_i^{(T_i,T_j)}(C, C, x) &= w_c(c + x) - w_u U(x) \\
\mathcal{C}_i^{(T_i,T_j)}(S, C, x) &= w_c x - w_u U(x) \\
\mathcal{C}_i^{(T_i,T_j)}(C, S, x) &= w_c(c + 1) - w_u U(1) \\
\mathcal{C}_i^{(T_i,T_j)}(S, S, x) &= w_c - w_u U(1) \ .
\end{aligned} \tag{6.4.0.1}$$

The parameters w_c, w_u are weighting factors such that $w_c + w_u = 1$.

We consider different forms of the utility function:

$$U_1(x) = 1 - e^{-x/R} \tag{6.4.0.2}$$

$$U_2(x) = \begin{cases} x/x_\theta, & x \le x_\theta \\ 1, & x > x_\theta \end{cases} \tag{6.4.0.3}$$

$$U_3(x) = \begin{cases} 0, & x \le x_\theta \\ 1, & x > x_\theta \end{cases} \tag{6.4.0.4}$$

Plots of the utility functions are shown in Fig. 6.4.

To simplify the analysis, we use an approximate form of the probability of a meeting with at least one cooperative U-node within a (now assumed normalized) distance x from T_i (see (6.2.0.2)):

$$F_k(x) = k \frac{\pi_j p(T_j, T_i) 2x}{\sum_{T_i} \sum_{T_j \neq T_i} \pi_i p(T_i, T_j)} \ . \tag{6.4.0.5}$$

Assuming normalized distances, the probability $\alpha(T_i, T_j)$ that U_i meets with a *specific* other U-node before meeting T_j, is:

$$\alpha(T_j, T_i) \triangleq \frac{2\pi_j p(T_j, T_i)}{\sum_{T_i} \sum_{T_j \neq T_i} \pi_i p(T_i, T_j)} \ . \tag{6.4.0.6}$$

Therefore (6.4.0.5) becomes (for simplicity, $\alpha(T_j, T_i) \triangleq \alpha$):

$$F_k(x) = k\alpha x \ .$$

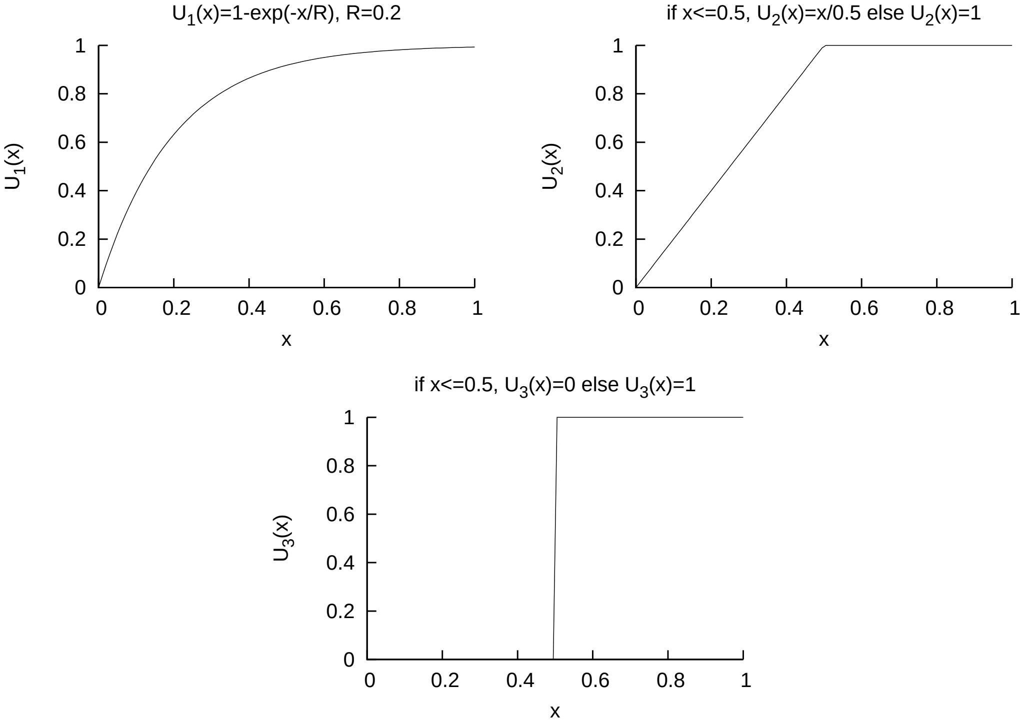


Figure 6.4: Plots of different utility functions examined.

The expected cost of node U_i to take action s_i, when k other nodes are cooperative ($0 \leq k \leq N-1$) is derived from (6.2.0.5). We will always assume that the utility function is normalized such that $U(1) = 1$.

For $U(x) = 1 - e^{-x/R}$, we derive

$$C_i^{(T_i,T_j)}(C|k) = k\alpha \left[w_c(c+1/2) - w_u(1-R) \right] + (1-\alpha)^k \left[w_c(c+1) - w_u \right] \ ,$$

which, using the approximation $(1-\alpha)^k \approx 1 - k\alpha$, is simplified to

$$C_i^{(T_i,T_j)}(C|k) = w_c(c+1) - w_u - k\alpha(w_c/2 - w_u R) \ . \tag{6.4.0.7}$$

Similarly, we derive

$$C_i^{(T_i,T_j)}(S|k) = w_c - w_u - k\alpha(w_c/2 - w_u R) \ . \tag{6.4.0.8}$$

For $k = 0$, we have

$$C_i^{(T_i,T_j)}(C|0) = w_c(c+1) - w_u \ , \tag{6.4.0.9}$$

$$C_i^{(T_i,T_j)}(S|0) = w_c - w_u \ . \tag{6.4.0.10}$$

In a similar manner we derive expected costs for different forms of the utility function. For $U = U_2$, we get

$$C_i^{(T_i,T_j)}(C|k) = w_c(c+1) - w_u - k\alpha(w_c/2 - w_u x_\theta/2) \; , \tag{6.4.0.11}$$

$$C_i^{(T_i,T_j)}(S|k) = w_c - w_u - k\alpha(w_c/2 - w_u x_\theta/2) \; , \tag{6.4.0.12}$$

$$C_i^{(T_i,T_j)}(C|0) = w_c(c+1) - w_u \; , \tag{6.4.0.13}$$

$$C_i^{(T_i,T_j)}(S|0) = w_c - w_u \; . \tag{6.4.0.14}$$

For $U = U_3$, we get

$$C_i^{(T_i,T_j)}(C|k) = w_c(c+1) - w_u - k\alpha(w_c/2 - w_u x_\theta) \; , \tag{6.4.0.15}$$

$$C_i^{(T_i,T_j)}(S|k) = w_c - w_u - k\alpha(w_c/2 - w_u x_\theta) \; , \tag{6.4.0.16}$$

$$C_i^{(T_i,T_j)}(C|0) = w_c(c+1) - w_u \; , \tag{6.4.0.17}$$

$$C_i^{(T_i,T_j)}(S|0) = w_c - w_u \; . \tag{6.4.0.18}$$

In all the above cases, it holds that $C_i^{(T_i,T_j)}(C|k) > C_i^{(T_i,T_j)}(S|k)$. However, because of the increasing utility by receiving an object close to its source, $C_i^{(T_i,T_j)}(C|k)$, $C_i^{(T_i,T_j)}(S|k)$ may not always be decreasing functions of k.

From (6.4.0.7),(6.4.0.8), if $U(x) = 1 - e^{-x/R}$, the condition for these to be decreasing functions of k is

$$\frac{w_c}{w_u} > 2R \; . \tag{6.4.0.19}$$

For $U = U_2$, the condition becomes

$$\frac{w_c}{w_u} > x_\theta \; , \tag{6.4.0.20}$$

and for $U = U_3$,

$$\frac{w_c}{w_u} > 2x_\theta \; . \tag{6.4.0.21}$$

The intuitive explanation behind these conditions is that, if the cost for retrieving an object is relatively greater than the benefit acquired from it, then cooperation between nodes can help in decreasing the mean overall costs. However, if these conditions are not satisfied, and $C_i^{(T_i,T_j)}(C|k)$, $C_i^{(T_i,T_j)}(S|k)$ are increasing functions of k, there is no initial incentive for cooperation and the best solution for each node, as well as an equilibrium for the overall game, is for each node to be selfish.

Now provided there is an initial incentive for cooperation (*i.e.*, $C_i^{(T_i,T_j)}(C|k)$, $C_i^{(T_i,T_j)}(S|k)$ are decreasing functions of k), condition

$$\mathcal{C}_i^{(T_i,T_j)}(C|k) < \mathcal{C}_i^{(T_i,T_j)}(S|0)$$

((6.3.0.1)) is used to derive the minimum number of cooperative nodes so that player U_i can have a benefit by cooperating on (T_i, T_j).

For $U = U_1$, using (6.4.0.7),(6.4.0.10) we get the condition

$$k > \frac{w_c c}{\alpha(w_c/2 - w_u R)} \ . \tag{6.4.0.22}$$

Similarly, for $U = U_2$, we get the condition

$$k > \frac{w_c c}{\alpha(w_c/2 - w_u x_\theta/2)} \ , \tag{6.4.0.23}$$

and for $U = U_3$,

$$k > \frac{w_c c}{\alpha(w_c/2 - w_u x_\theta)} \ . \tag{6.4.0.24}$$

From (6.4.0.22)-(6.4.0.24), we can deduce the following dependency relationships. First, a higher meeting probability α means that a smaller number of cooperative other U-nodes is required for a U-node to have a benefit by cooperation. Increasing the cost to acquire the content (either the constant cost c or the weight w_c) leads to a higher required number of cooperative nodes. Finally, if we decrease the parameters R or x_θ (which means that the content is less depreciated with the distance from its source), the required number of cooperative nodes decreases. Again, these conclusions hold provided $C_i^{(T_i,T_j)}(C|k)$ is a decreasing function of k, *i.e.*, nodes have an initial incentive to cooperate.

Despite the gains that may occur from cooperation, there is still a single equilibrium in the system in which all U-nodes are selfish. In the previous section, we also derived a strategy for each node that can lead to a cooperative equilibrium between all nodes, *i.e.*, an equilibrium in which all nodes are cooperative. In this strategy, each U-node transmits its acquired content with a probability p if the encountered U-node turns out selfish.

Copying (6.3.0.4), the condition under which this strategy is preferable for U_i on (T_i, T_j), is:

$$\mathcal{C}_i^{(T_i,T_j)}(C|N-1) \leq \sum_{k=0}^{N-1} p^k (1-p)^{N-k-1} \binom{N-1}{k} \mathcal{C}_i^{(T_i,T_j)}(S|k) \ .$$

For $U(x) = 1 - e^{-x/R}$, using (6.4.0.7),(6.4.0.8), we have

$$w_c(c+1) - w_u - (N-1)\alpha(w_c/2 - w_u R) \leq \sum_{k=0}^{N-1} p^k (1-p)^{N-k-1} \binom{N-1}{k} \left[w_c - w_u - k\alpha(w_c/2 - w_u R)\right] \ . \quad (6.4.0.25)$$

It is straightforward that

$$\sum_{k=0}^{N-1} p^k (1-p)^{N-k-1} \binom{N-1}{k} = 1 \ .$$

Further, it can be proved that

$$\sum_{k=0}^{N-1} p^k (1-p)^{N-k-1} \binom{N-1}{k} k = p(N-1) \ .$$

With these in mind, from (6.4.0.25) we derive the condition

$$N \geq 1 + \frac{w_c c}{\alpha(w_c/2 - w_u R)(1-p)} \ . \quad (6.4.0.26)$$

A similar analysis for $U = U_2$ leads to the condition

$$N \geq 1 + \frac{w_c c}{\alpha(w_c/2 - w_u x_\theta/2)(1-p)} \ , \quad (6.4.0.27)$$

while for $U = U_3$ we get

$$N \geq 1 + \frac{w_c c}{\alpha(w_c/2 - w_u x_\theta)(1-p)} \ . \quad (6.4.0.28)$$

Expressing (6.4.0.26)–(6.4.0.28) as conditions on the number $N-1$ of the remaining nodes and comparing with conditions (6.4.0.22)–(6.4.0.24), we observe that these are the same, except for the existence of the factor $(1-p)$. This has a nice mathematical explanation since $1/(1-p)$ is the number of encounters a U-node has to make in order to meet a cooperative U-node. Considering that the k given in (6.4.0.22)–(6.4.0.24) is the necessary number of other cooperative U-nodes in order for each U-node to have a benefit by cooperation, the second term in the rhs of (6.4.0.26)–(6.4.0.28) gives the minimum required number of cooperative U-nodes in the system for a benefit to occur.

Chapter 7

Gossip-based content dissemination and search

The problem of disseminating and searching for content in large-scale, distributed and unstructured networks - such as typical peer-to-peer (P2P) and ad-hoc networks - is challenging. Content dissemination can be realized in two ways: either the content itself is disseminated or, instead, an advertisement message indicating its availability and location is spread. Searching for content is typically achieved through the dissemination of a query looking for the content itself or for the information about its location. In both cases, a message needs to be disseminated. Consequently, a scheme that effectively disseminates the message, would be applicable to all the aforementioned problems. Such a scheme, based on gossiping algorithms, is the focus of this chapter. We present two analytical modeling approaches, an exact and an approximate one. We also evaluate the performance of the gossip-based dissemination or search schemes and examine the impact of the design parameters.

7.1 Introduction

The term "gossiping algorithm" encompasses any communication algorithm where messages between two nodes are exchanged opportunistically, with the intervention of other nodes that act as betweeners or forwarders of the message. It is inspired from the social sciences, in the same way as epidemic protocols were inspired from the spreading of infectious diseases [Eugster *et al.*, 2004b]. These two communication paradigms are very much similar, with the differences focusing on the different ways that gossiping nodes and infected nodes could behave: gossiping nodes adopt human-like characteristics, while the behaviour of infected nodes is governed by the dynamics of the virus or disease. Attractive characteristics of gossiping algorithms include simplicity, scalability and robustness to failures, as well as a speed of dissemination that is easily configurable. Gossiping can be identified with the spreading of rumors in a network, the dynamics of which are investigated in [Pittel, 1987, Nekovee *et al.*, 2007]. Additionally, gossiping protocols have been used for the computation of aggregate network quantities, such as sums, averages, or quantiles of certain node values [Kempe *et al.*, 2003].

Gossiping algorithms are suitable for communication in distributed systems, such as ad-hoc networks and generally systems with peer-to-peer communication. The process of communication consists of one or more rounds, in which a number of nodes that carry the message contact their peers, until the message reaches the intended recipient(s). Pittel [Pittel, 1987], Karp *et al.*[Karp *et al.*, 2000] and Kempre *et al.*[Kempe *et al.*, 2003] studied parallel communication in which each peer selects a single neighbour in the network to communicate with at every round. As most modern communication networks - either fixed or mobile - evolve, a node may be able to communicate with multiple peers, maintaining a short-time connection with each one. Therefore, we are motivated to analyze the network performance under the circumstance that a node selects more than one peer to communicate with in each round.

We consider gossip-based information dissemination and searching under the fundamental assumption of uniform neighbour selection over the entire network. As mentioned in [Eugster *et al.*, 2004a], the assumption of uniform peer selection guarantees the reliability of information dissemination, *i.e.*, that all nodes will ultimately be notified of the message. Uniform neighbour selection can be easily satisfied when peers have a complete view of the network. We consider here a distributed network with $N+1$ peers, where a unique identification number (ID) i, $1 \leq i \leq N+1$, is assigned to each peer. Due to difficulties in the mathematical analysis, cases where nodes have a partial view of the network are out of the scope of this work. However, uniformity can also be achieved in such cases by properly initializing the peerlists where each node logs its partial view of the network and by employing appropriate peerlist exchange schemes. For instance, in [Eugster *et al.*, 2003], a lightweight probabilistic broadcast algorithm is proposed, so that uniformly distributed individual views can be maintained, regardless of the peerlist size of a random peer.

Summarizing, in our model of the gossip-based scheme, communication between peers is assumed to take place in rounds, where in each round a node may contact one or more peers. We assume that uniform selection of multiple neighbours in one gossiping round is performed. The node that initiates the message dissemination process is referred to as the initiator, and is assumed to be the only "informed" node at the beginning of the process. Any node that receives the message will become an informed one, and may remain informed thereafter. Apart from the initiator, the other nodes that assist the dissemination or search can have different behavioural patterns. We distinguish between cooperative and noncooperative nodes. Nodes in the first category always become active upon being informed, *i.e.*, generate themselves dissemination or query messages in subsequent rounds. Noncooperative nodes on the other hand are unwilling to participate in the dissemination or search process themselves, but may return the content if they possess it. In the case of search, we also consider stifler nodes; the term is borrowed from [Nekovee *et al.*, 2007] and signifies nodes that were previously active, but from a certain point on lose interest in the dissemination of the query, and thus cease to participate in the search. Hence, it is a special case of cooperation. To avoid confusion, non-stifler nodes that are non-cooperative are also referred to as plain non-cooperative nodes.

We also distinguish gossip-based dissemination or search algorithms based on the level of knowledge that each node has about the progress of dissemination or search. We consider two extremes: at the one, each node has no knowledge whatsoever about the number or identities of nodes that have been previously informed or queried in the network. At the other extreme, each node has complete knowledge about these facts and avoids sending messages to previously informed or queried nodes at subsequent rounds. We call these cases blind selection scheme and smart selection scheme, respectively.

- *Blind gossiping-target scheme*: An active node forwards a message "blindly" at each round, without avoiding nodes it has informed before. This approach can model devices with small computational capabilities, that cannot keep a log of informed nodes, or cases where the identities of the devices are not known. It is equally appropriate to model situations with random encounters between nodes. For instance, a number of mobility models have exponential meeting times between mobile nodes (such as the Random Walk, Random Waypoint and Random Direction models, as well as more realistic, synthetic models based on these [Spyropoulos *et al.*, 2008]). In our model, the time until a node is informed approaches a geometric distribution, which is the discrete time analog to an exponential.

- *Smart gossiping-target scheme*: An active node forwards a message "smartly" at each round, by avoiding nodes that have been informed before either by itself or other nodes. This demands the knowledge of the identities of all informed nodes, and has a larger overhead compared to the blind gossiping-target scheme. We do not define the exact algorithm by which the identities of all informed nodes are made known to an active node. We only assume that this knowledge can be obtained at a cost that is small compared to the cost of dissemination or search, and use this case mainly as a reference for the efficacy of the blind gossiping-target algorithm. It is evident that informing nodes without any repetitions corresponds to the best – in terms of speed – performance of a gossip-based algorithm. Although it can be hard and costly to implement, there exist schemes that can approximate its performance. For example, a low-cost algorithm that could approximate smart gossiping is that, at each peer-to-peer communication, each node gets the list of nodes previously informed by its peer.

7.2 Modeling the gossip-based dissemination and search algorithms

A graph model of the network is considered. There is an initiator node I, and N other nodes in the graph. In the search case, there is a file f located in m of the other nodes of the graph ($m < N$) that the initiator wants to find. In the first round ($r = 1$) the initiator selects randomly k neighbours or gossiping targets, $1 \leq k \leq N$, to forward the message to. In each round, all the informed nodes select k gossiping targets randomly and independently to forward the message to.

In the case of dissemination the objective is to inform all nodes in the network – or a significant portion of them – in the shortest possible time. A separate algorithm, which we do not examine here, is required to stop the dissemination process. In the case of search if a node has the object then it returns it, and the query can be stopped. Otherwise the queried nodes begin to search themselves by forwarding the query to their neighbours.

A queried or informed node may or may not accept to forward the message. If it accepts, we say that this node is cooperative, otherwise non-cooperative. Cooperative nodes which are queried or informed become "active" and participate in the search or dissemination.

We consider two analytical modeling approaches: an exact modeling based on Markov chain theory, and an approximate one, based on techniques that were applied for the rumour-spreading problem. As shown in Section 7.10, the approximate model is computationally much simpler without losing much in accuracy. All models are validated with simulations. For technical reasons, in each modeling approach we consider slightly different behaviours of non-cooperative nodes with respect to the response to a query: in the first, non-cooperative nodes do not respond at all to a query, whereas in the second non-cooperative nodes only respond to the query if they have the requested object. In both cases, non-cooperative nodes do not forward the query.

7.3 First modeling approach: An $(N+1)-$state Markov model

In this section we provide a summary of an exact modeling approach for both the blind and smart versions of the gossiping algorithm, which is presented in more detail in [Tang *et al.*, 2011]. We consider that at each step, each peer has a complete view of other nodes in the network. In reality, peers may only communicate with a subset of peers in the network, and they have to update their views of the network periodically during information dissemination. However, performing an exact analysis of the gossip-based information dissemination process with dynamic peer partial views is extremely difficult, and requires an extremely large state space.

We consider the process $\{X(r), r > 1\}$ of the number of informed nodes at each round r. This is modeled as a discrete Markov chain (MC) with the state space $S = \{1, 2, ...N + 1\}$, which is the number of informed nodes x_i at each state i ($1 \leq i \leq N+1$)). With the MC, an $(N+1)\times(N+1)$ transition probability matrix P is constructed, in which each of the entries $P_{ij} = \Pr[X_{r+1} = j|X_r = i]$ describes the probability to move from state i to state j at a certain round r. The initial state vector is $s[0] = [1, 0, 0, 0...0]$.

We seek the analytical expressions of P_{ij} for the different schemes described in the previous section. Afterwards, performance metrics that can be used to evaluate the performance of the algorithms are proposed. We derive the transition probabilities P_{ij} combinatorially. This approach is inspired by the occupancy problem in the balls and bins model, where the informed nodes are balls, and nodes to be selected in the network are bins.

7.3.1 The transition probabilities P_{ij}

Scheme 1 – Blind neighbor selection with cooperative nodes

With the blind neighbor selection, a node chooses its targets from the N neighbors in the network (excluding itself). The transition probabilities can be calculated by applying the balls and bins model [Tang *et al.*, 2011].

To move from state i to state j, each of the i informed nodes selects k different neighbours from a set of $j-1$ nodes (excluding itself). The j nodes can be decomposed into two parts: the i informed nodes and the $j-i$ new informed ones. Overall, the $j-i$ new informed nodes should be selected at least once, otherwise the Markov process cannot reach state j.

The operation of the i informed nodes, selecting k different neighbours from the $j-1$ nodes, is equivalent to placing the i groups of k balls to the $n-1$ bins, excluding a bin that has the same numbering as the group of balls. Selecting from the $j-i$ new informed nodes is analogous to the placement to $j-i$ separate bins; and the selection from the i informed ones is analogous to placing the balls to i separate bins. The fact that the $j-i$ new nodes are selected at least once, is equivalent to the requirement that at least the

$j - i$ separate bins are occupied.

In [Tang *et al.*, 2011] we derive the elements in the transition probability matrix P as

$$P_{ij} = \begin{cases} \frac{\binom{N+1-i}{j-i}}{\binom{N}{k}^i} \sum_{t=0}^{j-i} (-1)^t \binom{j-i}{t} \binom{j-1-t}{k}^i & \text{if } i-1 \geqslant k \text{ and } i \leqslant j \leqslant i(k+1) \\ \frac{\binom{N+1-i}{j-i}}{\binom{N}{k}^i} \sum_{t=0}^{j-1-k} (-1)^t \binom{j-i}{t} \binom{j-1-t}{k}^i & \text{if } i-1 < k \text{ and } k+1 \leqslant j \leqslant i(k+1) \\ 0 & \text{otherwise} \end{cases} \tag{7.3.1.1}$$

where $\binom{N+1-i}{j-i}$ is the number of ways to choose $j - i$ new nodes among the $N + 1 - i$ uninformed nodes at state i, and $\binom{N}{k}^i$ is the total number of possible configurations that result when k different neighbours out of N nodes are chosen i times.

Scheme 2 – Smart neighbor selection with cooperative nodes

Given i informed nodes in the network, and that each informed node selects k different neighbours from the remaining $N+1-i$ uninformed nodes, the problem is analogous to the balls and bins model described in [Tang *et al.*, 2011]. By randomly throwing i balls, to $N + 1 - i$ bins, with exactly $N + 1 - j$ bins empty, the transition probabilities P_{ij} are

$$P_{ij} = \begin{cases} p_{N+1-j}(i, N+1-i, k) & \text{if } N+1-i \geqslant k \text{ and } i+k \leq j \leq i(k+1) \\ 1 & \text{if } N+1-i < k \\ 0 & \text{otherwise} \end{cases} \tag{7.3.1.2}$$

where

$$p_{N+1-j}(i, N+1-i, k) = \frac{\binom{N+1-i}{N+1-j}}{\binom{N+1-i}{k}^i} \sum_{t=0}^{j-i-k} (-1)^t \binom{j-i}{t} \binom{j-i-t}{k}^i \tag{7.3.1.3}$$

which is valid only when $N + 1 - i \geqslant k$. The lower boundary in $i + k \leq j \leq i(k + 1)$ is the worst case where all the i informed nodes choose their neighbours from the same k new nodes, leading to minimally, $i + k$ informed nodes at state j. The upper boundary of $j \leq i(k + 1)$ describes the optimal situation where all the k neighbours chosen by the i informed nodes are different, resulting in maximally, $i(k + 1)$ informed nodes at the next state j.

When $N + 1 - i < k$, all the remaining $N + 1 - i$ nodes will be informed when an informed node chooses less than k neighbours. Thus, we confine to $P_{ij} = 1$ in this case.

Scheme 3 – Blind neighbour selection with non-cooperative nodes

The basic idea to solve the blind neighbour selection with non-cooperative nodes obeys the following steps: First, we calculate the probability to have s informed nodes at the next state. This can be done by using the same method in either (7.3.1.1) or (**??**). Thus, there are $z = s - i$ newly informed nodes.

Substituting $j = s$ in (7.3.1.1), we could calculate the probability $P(i, s)$ to move from state i to state s. We define $B(z,\ j - i, \beta)$ as the probability that there are exactly $j - i$ cooperative nodes from the z nodes.

$$B(z,\ j-i,\beta) = \binom{z}{j-i}\beta^{j-i}(1-\beta)^{z-(j-i)}$$

with $z = s - i$ and $0 \leq j - i \leq z$.

Finally, P_{ij} in this case is computed as

$$P_{ij} = \begin{cases} \sum_{s=j}^{\min\{N+1,i(k+1)\}} P(i,s)B(z,\ j-i,\beta) & \text{if } i-1 \geqslant k \text{ and } i \leq j \leq \min\{N+1, i(k+1)\} \\ \sum_{s=j}^{\min\{N+1,i(k+1)\}} P(i,s)B(z,\ j-i,\beta) & \text{if } i-1 < k \text{ and } k+1 \leq j \leq \min\{N+1, i(k+1)\} \end{cases} \tag{7.3.1.4}$$

Scheme 4 – Smart neighbour selection with non-cooperative nodes

The analysis of scheme 4 starts with seeking the probability that there are z new nodes being selected by the i informed ones. By employing (7.3.1.3) with $z + i$ instead of j, we have

$$p_{N+1-(z+i)}(i, N+1-i, k) = \frac{\binom{N+1-i}{N+1-(z+i)}}{\binom{N+1-i}{k}^i} \sum_{t=0}^{z-k} (-1)^t \binom{z}{t} \binom{z-t}{k}^i$$

To move from state i to state j, there should be exactly $j - i$ new informed nodes deciding to be cooperative. The probability that $j - i$ out of the z newly selected nodes are cooperative is computed as

$$B(z, j-i, \beta) = \binom{z}{j-i}\beta^{j-i}(1-\beta)^{z-(j-i)}$$

with $0 \leq j - i \leq z$.

The number z of new selected nodes takes its value between $(j - i,\ N + 1 - i)$ if $N + 1 - i \leq ik$; and $(j - i,\ ik)$ if $N + 1 - i > ik$. Therefore, we come to the transition probabilities

$$P_{ij} = \begin{cases} \sum_{z=k}^{\min\{N+1-i,ik\}} p_{N+1-(z+i)}(i, N+1-i, k) B(z, j-i, \beta) & \text{if} N+1-i > k \text{ and} \\ \qquad i \leq j \leq \min\{N+1-i, ik\} & \\ B(z = N+1-i, j-i, \beta) & \text{if } N+1-i < k \\ 0 & \text{otherwise} \end{cases} \tag{7.3.1.5}$$

7.4 Performance of message dissemination algorithm

Here we present some results based on this analysis.[1] The analytical results for each setting of the parameters are compared against those derived from a simulation of the gossiping process, programmed in C. 10^4 iterations are carried out for each simulation scenario. For both the information dissemination and search process, random selection of k neighbours is performed. The initiator is also randomly chosen in each of the simulation instances (iterations). The information dissemination process stops when there are m preselected informed nodes in the network. For each iteration, we collect the number of gossiping rounds until the program finishes, from which, the distribution function and the mean are computed. For the search process, the number of searched nodes until the end of the program is captured.

The main performance measure we examine is the (minimum) number of rounds to inform m randomly selected nodes, denoted by $A_{N+1}(m)$ for both the blind and smart gossiping cases. Let $e_m(r)$ denote the event of having m randomly selected nodes informed by (and including) round r. It can be shown that

$$Pr[e_m(r)] = \frac{1}{\binom{N}{m}} \sum_{i=1}^{N+1} \binom{i-1}{m} s_i[r],$$

where $s_i[r]$ is the probability of having any i nodes informed by (and including) round r. It follows that $Pr[A_{N+1}(m) = r] = Pr[e_m(r)] - Pr[e_m(r-1)]$. The mean number of rounds to inform m randomly selected nodes is $\bar{A}_{N+1}(m) = \sum_{r=1}^{r_{max}} rPr[A_{N+1}(m) = r]$. For numerical calculations, we take the upper limit of this summation as $r_{max} = max\{r : Pr[e_m(r)] > 1 - \epsilon\}$, where ϵ is a very small positive number.

[1] More results can be found in [Tang *et al.*, 2011].

In Fig. 7.1 we present the results on the probability that all N nodes are informed by round r (*i.e.*, $m = N$). As we can see from the graphs, the simulation results for $Pr[e_m(r)]$ match those from the exact analysis very well. We observe a bi-modal behaviour, *i.e.*, the probability of all nodes being informed increases very abruptly after a certain threshold value of r. The larger the network, the larger this threshold value is. For instance, under the blind selection scheme, it is possible to inform all nodes after four rounds in a small network with $N = 10$, as shown in Fig. 7.1(a). While for a larger network with $N = 100$, it is only possible to inform all nodes after 8 gossiping rounds.

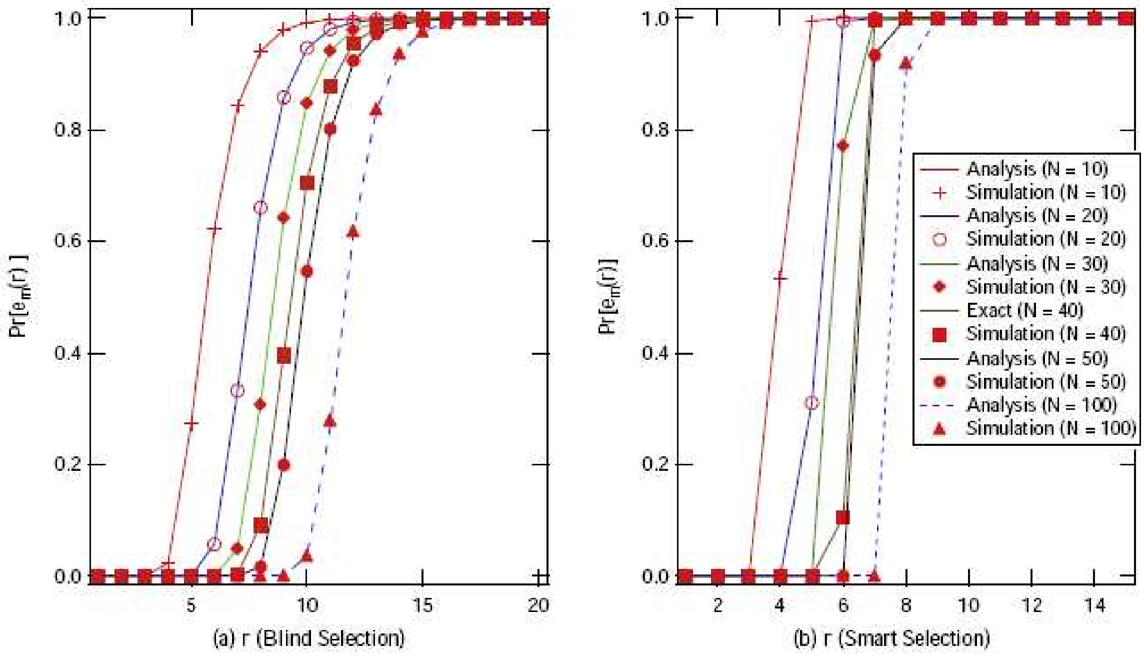


Figure 7.1: The probability that all nodes ($m = N$) are informed by round r under the blind selection scheme (a), and the smart selection scheme (b) ($k = 1$ and $\beta = 1.0$).

In Fig. 7.2 we show the results on the mean number of rounds to inform the entire network. We notice that $\bar{A}_{N+1}(m)$ grows almost proportionally to $\log(N)$. This is consistent with the fact that, asymptotically (for large N and small k), the expected number of rounds of any dissemination process in a general graph scales in the order of $\log(N)$, as shown in [Mieghem, 2005]. In [Pittel, 1987, Karp *et al.*, 2000], the authors also gave the same $\log(N)$ upper bound of gossiping-based algorithms with $k = 1$. Consequently, we can approximate the mean minimum number of rounds to inform the entire network as $\bar{A}_{N+1}(m) \sim \gamma_k \log(N) + \alpha_k$, where γ_k and α_k are variables depending on k. As seen by comparing Fig. 7.2(a) and (b), the speed of disseminating content under the smart selection scheme is less affected by increasing the network size, since the slope γ_k under this scheme is always smaller than that under the blind scheme for the same k. The smart gossiping scheme is always faster than the blind one, and the relative difference increases with k. In the results shown in Fig. 7.2(a), smart gossiping is about 1.7 times faster than the blind one for $k = 1$, and about 1.2 times faster for $k = 2$.

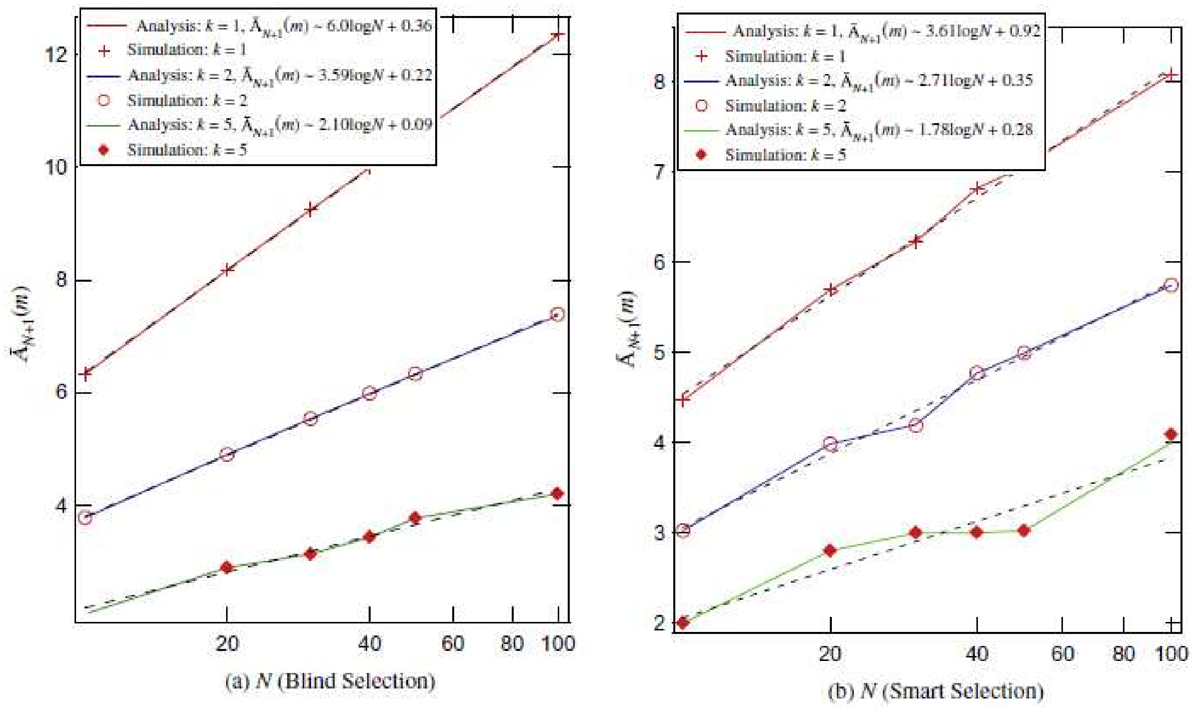


Figure 7.2: Mean number of rounds required to inform the entire network, $\bar{A}_{N+1}(m)$, as a function of N under blind (a), and smart (b) selection schemes ($m = N$, $\beta = 1.0$, and varying k). The x-axis is plotted on log scale and the dotted lines are the fitting curves.

In Fig. 7.3, $\bar{A}_{N+1}(m)$ is plotted as a function of b for different network sizes. As the cooperation probability b increases, $\bar{A}_{N+1}(m)$ decreases logarithmically with the same slope for different network sizes, for both the blind and the smart selection scheme. This phenomenon indicates that the mean number of rounds to inform the entire network decreases at the same speed as a function of β, regardless of the network size. Furthermore, by decreasing the cooperation probability β, the performance of disseminating content degrades for both the blind and the smart algorithm. For example, the mean number of rounds to inform the entire network with $\beta = 0.2$ is approximately 5.3 times of that with

$\beta = 1.0$ for the blind selection and 5.8 times for the smart selection. The performance of the smart selection scheme is, as expected, better than the blind selection scheme. For instance, for $N = 100$, with $\beta = 1.0$, it takes on average, 3.9 more rounds to inform all nodes for the blind selection than for the smart selection; and the blind selection scheme needs 18.1 more rounds to inform the entire network with $\beta = 0.2$, compared with the smart selection.

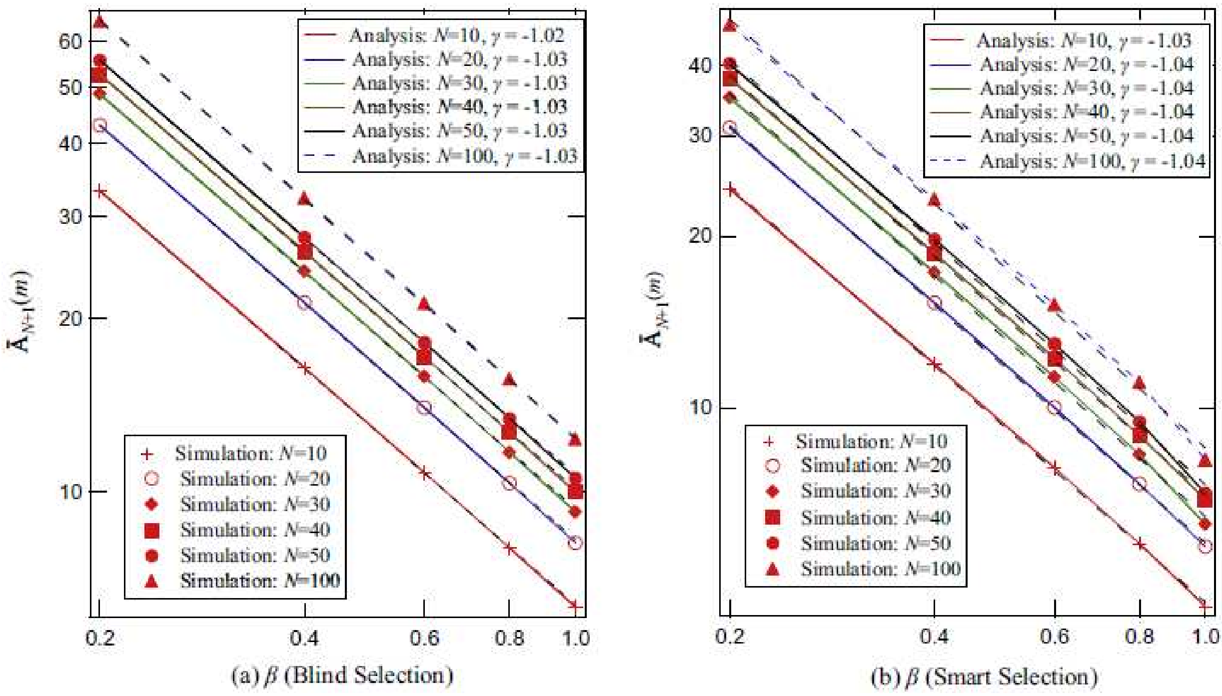


Figure 7.3: Mean minimum number of rounds required to inform the entire network, $\bar{A}_{N+1}(m)$, as a function of β under the blind (a), and smart (b) selection schemes ($k = 1$, $m = N$, and varying β). Both sub-figures are plotted on log-log scale. The dotted lines represent the fitting curves, and γ is the fitting parameter of $\log(\bar{A}_{N+1}(m)) \sim \gamma \log(\beta) + \alpha$.

Results for the search process derived with the exact analytical method are not presented here, but can be found in [Tang *et al.*, 2011]. Instead, in what follows, we will present results for the performance of the search algorithm based on an approximate analysis.

7.5 Second modeling approach: Approximate analysis for blind search with cooperative or non-cooperative nodes

Each node that receives a search query will cooperate to forward the query with probability c ($0 \leq c \leq 1$).[2] If several active nodes query the same node, the latter node decides whether to be cooperative or not by a single Bernoulli trial. We do not consider that the node performs several independent trials, one for each query.

We consider a sequence of steps (or rounds) $r = 1, 2, \ldots$ until the file is found. If at step r there are $\hat{A}(r)$ active nodes then, provided the file is not yet found, the probability of finding it at the rth step, $S(r)$, is:

$$S(r) = 1 - (1 - p_s)^{\hat{A}(r)} \ , \tag{7.5.0.1}$$

where p_s is the probability that a single search (consisting of k different random queries) succeeds.

To find p_s, notice that the problem is equivalent to the one where, in a set of $N - 1$ nodes, there are m marked nodes and we randomly select a group of k nodes. We want to find the probability that at least one marked node is selected. The probability that our selection returns exactly u marked nodes ($u \leq \min(m, k)$) is

$$p_u = \frac{\binom{m}{u}\binom{N-1-m}{k-u}}{\binom{N-1}{k}} \ .$$

Indeed, the marked nodes can be chosen in $\binom{m}{u}$ different ways, the unmarked ones in $\binom{N-1-m}{k-u}$ ways, and the total number of ways to select k nodes is $\binom{N-1}{k}$. Further, $p_s = 1 - p_0$, therefore

$$p_s = 1 - \frac{\binom{N-1-m}{k}}{\binom{N-1}{k}} \ . \tag{7.5.0.2}$$

The probability of finding the file f at the rth step is,

$$p(r) = S(r) \prod_{i=1}^{r-1} (1 - S(i)) \ . \tag{7.5.0.3}$$

This formula is an approximation because it implicitly assumes that each round is independent of the other.

[2] We use a different symbol for the cooperation probability in the second modeling approach, because the behaviour of cooperative nodes is not exactly the same in the two models.

A deterministic approximation $\hat{A}(r)$ for the number of active nodes in each round can be found using the method presented in [Pittel, 1987], which is extended to k neighbours that are cooperative with probability c. Consider the process $\{I(r), r \geq 1\}$ of the number of inactive nodes in each round. Given that at round r there are $i(r)$ inactive nodes and $A(r)$ active ones, the mean number of inactive nodes at round $r+1$ will be

$$\begin{aligned} E[I(r+1)] =& i(r)\left[\left(1-\frac{k}{N-1}\right)^{A(r)} + \left[1-\left(1-\frac{k}{N-1}\right)^{A(r)}\right](1-c)\right] \\ =& i(r)\left[(1-c)+c\left(1-\frac{k}{N-1}\right)^{A(r)}\right] . \end{aligned}$$

For fixed k and large N, we use a second order expansion of $(1-\frac{k}{N-1})^{A(r)}$, so that $(1-\frac{k}{N-1})^{A(r)} \approx e^{-A(r)(\frac{k}{N-1}+\frac{k^2}{2(N-1)^2})}$, and

$$E[I(r+1)] = i(r)\left[(1-c)+ce^{-A(r)(\frac{k}{N-1}+\frac{k^2}{2(N-1)^2})}\right] .$$

From this, by assuming $I(r) = i(r)\ \forall r$, we derive the deterministic approximation

$$I(r+1) = I(r)\left[(1-c)+ce^{-A(r)(\frac{k}{N-1}+\frac{k^2}{2(N-1)^2})}\right] . \tag{7.5.0.4}$$

Using that $A(r) = N - I(r)$, we finally obtain the recursion:

$$A(r+1) = Nc + A(r)(1-c) - (N-A(r))ce^{-A(r)(\frac{k}{N-1}+\frac{k^2}{2(N-1)^2})} , \tag{7.5.0.5}$$

with $A(1) = 1$. Since $A(r)$ is not an integer in general, we round it to the nearest integer, which we denote by $\hat{A}(r) = [A(r)]$.

Following a similar approach as in [Pittel, 1987], it can be shown that the distribution of $I(r+1)$, given $i(r)$ is indeed concentrated sharply around $i(r)[(1-c)+c\exp(-A(r)(\frac{k}{N-1} + \frac{k^2}{2(N-1)^2}))]$, and that the approximation becomes more accurate as $k/N \to 0$.

From (7.5.0.3), we have constructed an approximate distribution for the number of steps until the file is found. We can then derive the mean number of steps until the file is found and the mean number of nodes A involved in the search (activated nodes):

$$E[r] = \sum_{r=1}^{\infty} rp(r), \tag{7.5.0.6}$$

$$E[A] = \sum_{r=1}^{\infty} \hat{A}(r+1)p(r) . \tag{7.5.0.7}$$

(Notice that $\hat{A}(r+1)$ nodes will be activated approximately at round r.)

For numerical calculations, as an upper bound on the support of r we take

$$r_{max} = \min\{r : \prod_{i=1}^{r-1}(1 - S(i)) < \epsilon\} \ , \tag{7.5.0.8}$$

where ϵ is a number close to zero.

Remark 1. In [Tang *et al.*, 2011], we have derived an exact model for a slightly different version of the blind search algorithm. Generally, the exact approach for modeling the search algorithm requires the calculation of the $N \times N$ transition matrix $Q_{[ij]}$, where the (i, j)-th value is the probability of going from i to j active nodes in one round. Then $Q^r(1, i)$ denotes the probability that there are i active nodes in r rounds.

The probability of finding the file in r rounds, denoted here by $B(r)$, can be calculated as

$$B(r) = \sum_{i=1}^{N} \left[1 - \frac{\binom{N-i}{m}}{\binom{N-1}{m}}\right] Q^r(1, i) \ . \tag{7.5.0.9}$$

This is a probability distribution, therefore the probability of finding the file exactly at round r is given by:

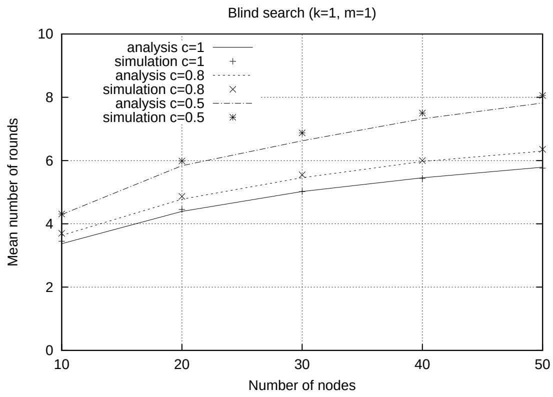
$$B(r) - B(r-1) \ . \tag{7.5.0.10}$$

In Section 7.10, we compare the complexity of the two models and examine the accuracy of the approximate one. It is shown that the reduction in computational cost is of the order of $O(N^2)$, while the relative accuracy of the approximation is higher than 95% in the majority of cases (the comparison holds when $c = 1$).

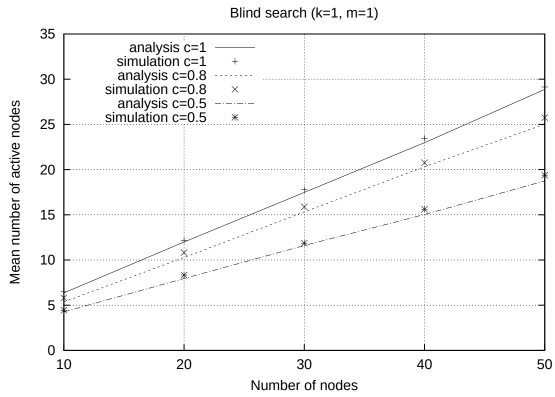
We validate our approximation by means of simulation. The simulations were performed with 100 instances of random file positions, and 100 random executions of the search in each instance, leading to a total of 10^4 repetitions in each experiment. We evaluate the mean number of rounds and the mean number of activated (infected) nodes until at least one copy of the file f is found, varying the number of nodes N in the graph, the cooperation probability c and parameters k and m. The value of ϵ in (7.5.0.8) was set to 10^{-6}. Results for different cases are shown in Fig. 7.4.

These figures illustrate that the simulation results match those from the theoretical analysis very well, except for large values of k.[3] As the size of the network increases,

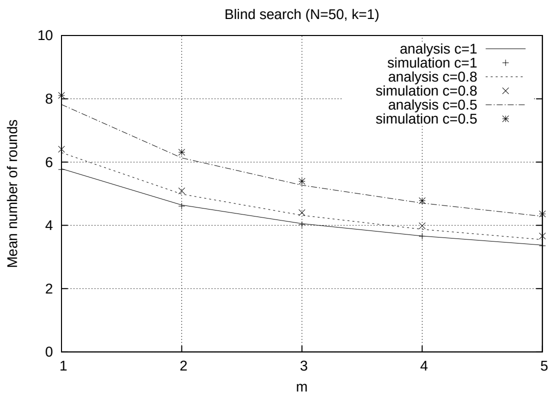
[3] It is noted that the improvement in accuracy by using the exact expression $(1 - \frac{k}{N})^{A(r)}$, or adding more terms in its expansion, is negligible.



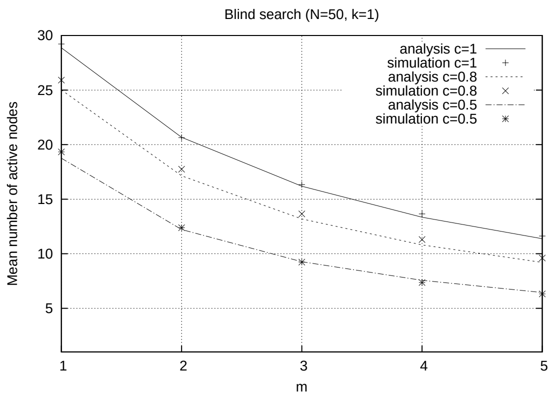
(a)



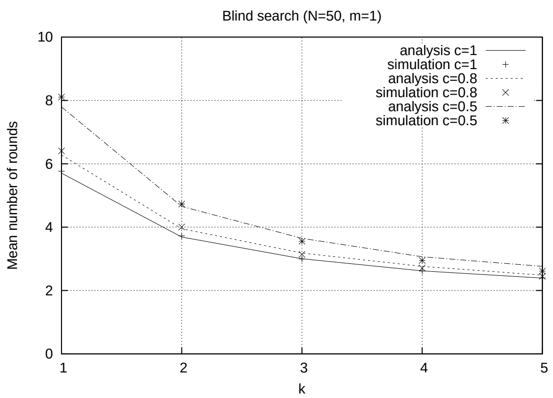
(b)



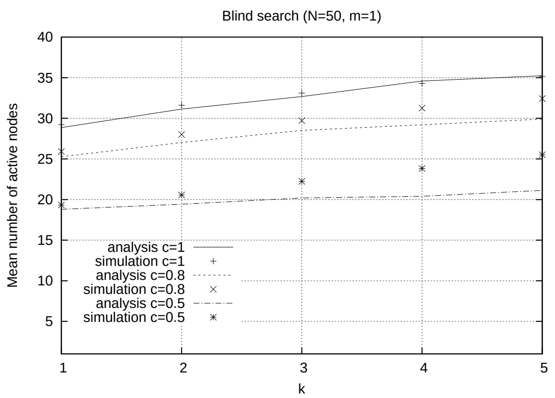
(c)



(d)



(e)



(f)

Figure 7.4: Analytical and simulation results for the mean number of rounds and mean number of active nodes with varying network parameters, for the blind search algorithm.

the number of active nodes increases linearly, while the increase in the mean number of rounds is superlinear, at a decreasing rate, as shown in Fig. 7.4(a) and 7.4(b). This implies that the number of rounds can be well-fitted using a logarithmic function, as will be shown more clearly in Section 7.8.

We examine the impact of search parameters k, m, in Fig. 7.4(c)-7.4(f). Note that the search can become faster by increasing either m or k. By comparing Fig.7.4(c) with 7.4(e), we note that the increase in speed is higher for large values of k. However, Fig.7.4(d) and 7.4(f) show that increasing k has the disadvantage of increasing the number of active nodes, and thus produces higher communication overhead. For example, calculations based on the simulation results show that for $c = 1$ and $N = 50$ nodes, increasing m to 3 yields a relative decrease in the mean number of rounds by 31%, and in the mean number of active nodes by 45%. On the other hand, increasing k to 3 yields a higher relative decrease in the mean number of rounds by 48%, but an *increase* in the mean number of active nodes by 14%. Overall, we remark that increasing the number of queried neighbours results in a great redundancy in the number of nodes that participate in the search with only small gains in speed.

7.6 Analysis for the smart search case with cooperative or non-cooperative nodes

An analysis for the smart search process is presented below. An approximate analysis similar to the one for the blind search model fails here, due to the varying probabilities of successful query. Instead we adopt a direct combinatorial approach, by considering a generalization of the occupancy (balls and bins) problem [Feller, 1968].

The generalized balls and bins problem is defined as follows: In a population of n bins, suppose we randomly distribute r groups of k balls, such that in each group, no two balls go in the same bin and successive distributions of groups of balls are independent. We want to find the probability that exactly v bins remain empty, where $v = 0, 1, \ldots, n-k$ (it is assumed that $n > k$).

We follow the approach in [Feller, 1968, Section IV.2] for the classical occupancy problem (where $k = 1$). The total number of ways to distribute r groups of balls in the way described above is $\binom{n}{k}^r$. Similarly, the total number of ways to assign them to $n-1$ bins is $\binom{n-1}{k}^r$, so that the probability that one *given* bin is empty, is $\binom{n-1}{k}^r / \binom{n}{k}^r$.

Generally, the probability that v *given* bins are empty is $\binom{n-v}{k}^r / \binom{n}{k}^r$.

Therefore, the probability that at least one bin is empty is, by the inclusion-exclusion method,

$$\sum_{i=1}^{n-k} (-1)^{i-1} \binom{n}{i} \frac{\binom{n-i}{k}^r}{\binom{n}{k}^r} .$$

The probability that all bins are occupied, denoted by $p_0(r,k,n)$, is

$$\begin{aligned} p_0(r,k,n) &= 1 - \sum_{i=1}^{n-k} (-1)^{i-1} \binom{n}{i} \frac{\binom{n-i}{k}^r}{\binom{n}{k}^r} \\ &= \sum_{i=0}^{n-k} (-1)^{i} \binom{n}{i} \frac{\binom{n-i}{k}^r}{\binom{n}{k}^r} . \end{aligned} \tag{7.6.0.1}$$

Consider now the case where exactly v non-given bins are empty. These v bins can be chosen in $\binom{n}{v}$ different ways. The k balls of each of the r groups are distributed among the remaining $n-v$ bins such that exactly $n-v$ are occupied. The mean number of such distributions is

$$\binom{n-v}{k}^r p_0(r,k,n-v) .$$

Dividing by the total number of possible configurations $\binom{n}{k}^r$ we obtain the probability $p_v(r,k,n)$ that exactly v bins are empty:

$$p_v(r,k,n) = \binom{n}{v} \frac{\binom{n-v}{k}^r}{\binom{n}{k}^r} \sum_{i=0}^{n-k-v} (-1)^i \binom{n-v}{i} \frac{\binom{n-v-i}{k}^r}{\binom{n-v}{k}^r} . \tag{7.6.0.2}$$

Based on (7.6.0.2), we find transition probabilities of the form $p(x_i, x_j)$, which denotes the probability that if at a certain round of the algorithm there are x_i active nodes, then at the next round there will be x_j active ones ($x_j \geq x_i$). It is emphasized here that each round corresponds to one transition. In our terminology, *"at (or in) a certain round"* will have the meaning *"after the transition that occured in this round and before the next transition"*. The first round marks the transition from 1 active node (the initiator) to a maximum number of $k+1$ active nodes.

Since there are no repetitions for the smart search, the transition probabilities can be found by directly applying (7.6.0.2), substituting $n = N - x_i$, $v = N - x_j$, and $r = x_i$:

$$p(x_i, x_j) = p_{N-x_j}(x_i, k, N - x_i) . \tag{7.6.0.3}$$

From (7.6.0.2),(7.6.0.3) we have

$$p(x_i, x_j) = \binom{N - x_i}{N - x_j} \frac{\binom{x_j - x_i}{k}^{x_i}}{\binom{N-x_i}{k}^{x_i}} \sum_{\ell=0}^{x_j - x_i - k} (-1)^{\ell} \binom{x_j - x_i}{\ell} \frac{\binom{x_j - x_i - \ell}{k}^{x_i}}{\binom{x_j - x_i}{k}^{x_i}} . \qquad (7.6.0.4)$$

Each of the $x_j - x_i$ queried nodes will decide whether or not to be cooperative independently with probability c. The probability that α out of $x_j - x_i$ nodes will actually be activated is

$$B(x_j - x_i, \alpha, c) \stackrel{def}{=} \binom{x_j - x_i}{\alpha} c^{\alpha} (1 - c)^{x_j - x_i - \alpha} .$$

Therefore, the probability that there will be $x_i + \alpha$ active nodes in the next round is

$$p(x_i, x_i + \alpha) = \sum_{x_j - x_i > \alpha} p(x_i, x_j) B(x_j - x_i, \alpha, c) . \qquad (7.6.0.5)$$

Based on the transition probabilities, we construct the $N \times N$ transition matrix Q with entries $p(x_i, x_j)$ for $i, j = 1, \ldots, N$. The value of the ith element of the first row of the matrix Q^r is the probability that there are i active nodes at round r.

Let us denote by $p_s(v)$ the probability that at least one of the v active nodes finds a copy of the file. The probability $S(r)$ of finding the file by (and including) round r is

$$S(r) = \sum_{v} Q^{(r-1)}(1, v) \left(1 - (1 - p_s(v))^{v}\right) , \qquad (7.6.0.6)$$

where $p_s(v)$ is the probability that a search by a single node finds a copy of the file, given that there are already v active nodes.

To find $p_s(v)$, we take a similar approach as in Section 7.5. It is

$$p_s(v) = 1 - \frac{\binom{N-v-m}{k}}{\binom{N-v}{k}} \qquad (7.6.0.7)$$

Finally, the probability of finding the file at the r-th round is given by (7.5.0.3).[4] We emphasize that this formula is again an approximation, since it is assumed that each round is independent of the other.

Based on the above distribution, we easily derive the expected number of rounds until a file is found, as in (7.5.0.6). The mean number of nodes activated during the search process is

$$E[A] = \sum_{r=1}^{\infty} E[\alpha(r)] p(r) , \qquad (7.6.0.8)$$

[4]Notice that (7.6.0.6) is not a cdf, so we can't use $S(r) - S(r-1)$ to find the probability of successful search at round r.

where $E[\alpha(r)]$ is the mean number of active nodes in round r, derived from the distribution $Q^r(1, \cdot)$. (For the smart search, the summation index in (7.5.0.6),(7.6.0.8) is upper bounded by $\lceil N - 1/k \rceil$.)

We take both analytical and simulation results for the same values of the parameters N, c, k and m as for the blind search algorithm. The simulations are conducted for the same number of repetitions as for blind search. Results are shown in Fig. 7.5. Generally, the model is extremely accurate for $c = 1$, but as the cooperation probability decreases it starts to deviate from the simulated behavior. For small values of c, we remark that the model is less accurate than our model for blind search, even though it follows a combinatorial approach that is exact up to (7.6.0.6). We attribute this to the fact that the intermediate search steps are more correlated for the smart search algorithm. Thus, as it is intuitively reasonable, the assumption of independence over rounds leads to worse results for the smart search than for the blind search case.

The same observations hold regarding the effect of the cooperation probability, the number of queried neighbours and the number of copies of the file, as in the blind search case. It is again emphasized that the behavior with respect to increasing k is an outcome of the tradeoff between the increased speed of discovery and the redundancy in the total number of messages sent.

We notice that there is only a small performance improvement of the smart over the blind search algorithm, expressed through the decrease in the mean number of rounds. This improvement becomes less pronounced as k or m increase. For example, based on the simulation results for $N = 50$ and $c = 1$, the relative reduction of smart search in the mean number of rounds is 13% when $k = 1$, $m = 1$, 3% when $k = 1$, $m = 3$, and 5% when $k = 3$, $m = 1$. This impovement is relatively larger when the cooperation probability decreases: for $N = 50$ and $c = 0.5$, the corresponding reductions were 27%, 16% and 6%.

However, the mean number of active nodes may be greater for the smart search, due to the fact that we always query only inactive nodes. This was true in most of the derived results. For $N = 50$ nodes, the relative increased reached up to 10% for $c = 1$ and $k = 1$, $m = 3$, while for $c = 0.5$ it increased up to 31% for $k = 1$, $m = 3$. For the values of $N = 50$ and $c = 0.5$, only a slight decrease of 1% was observed when $k = 3$, $m = 1$.

The overall results illustrate that when comparing the two cases, the smart search

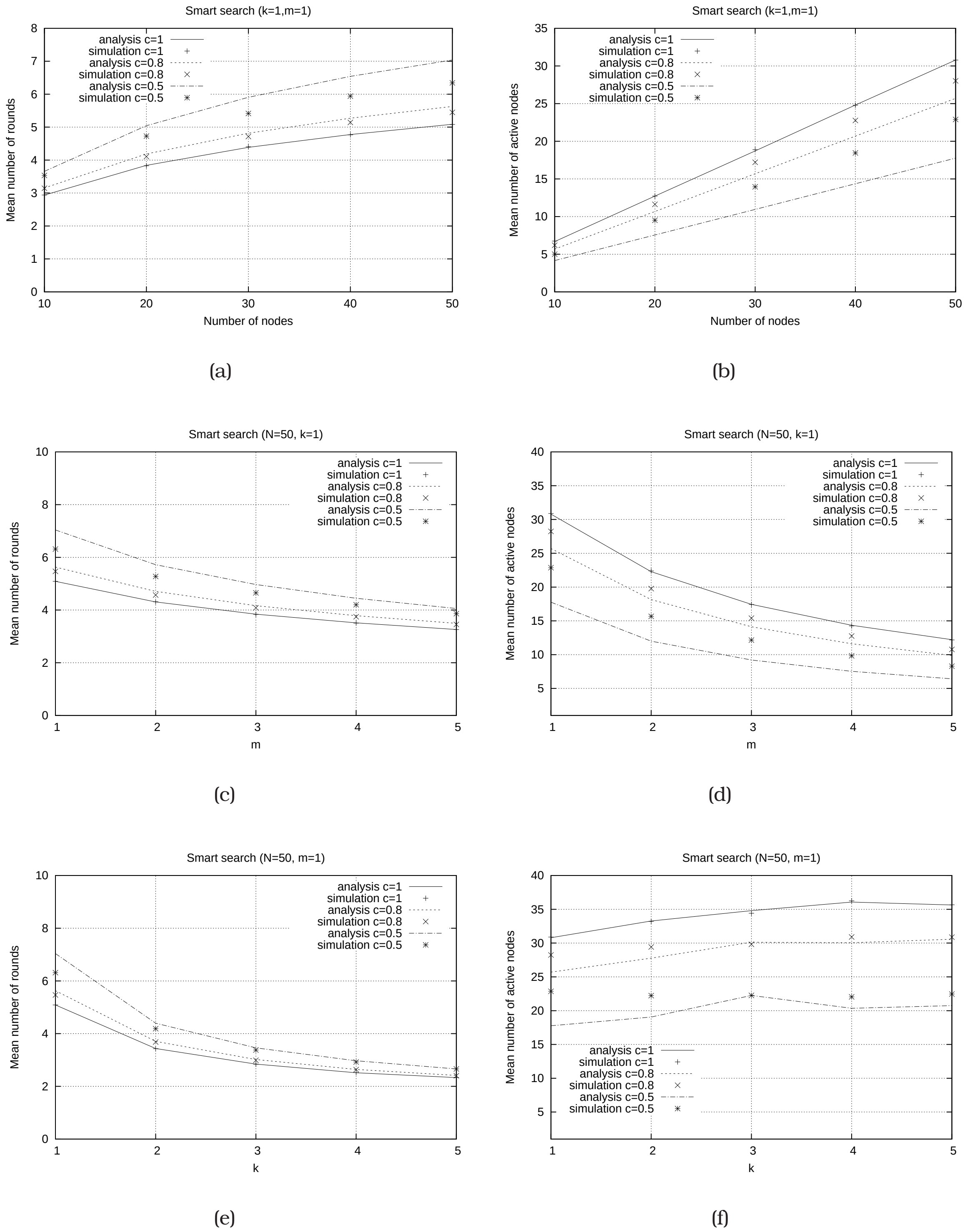


Figure 7.5: Analytical and simulation results for the mean number of rounds and mean number of active nodes with varying network parameters, for the smart search algorithm.

does not offer a significant improvement. This leads us to the conclusion that if the overhead incurred in the smart search algorithm for informing all active nodes of the identities of queried nodes is not negligible compared to that of the search procedure, it is highly likely that there is not much to gain by such a scheme.

7.7 Analysis with stiflers

Another behavioral pattern that we consider is *stifling*. In this pattern, each of the nodes that are (or become) active at a certain round may cease to be active and not participate in the search process any more. This could express a node's loss of interest in spreading the query message further in the network.

We analyse this stifling behaviour based on the assumption that at each round of the search, each active node may become a stifler independently with probability s. A node that becomes a stifler is considered as inactive, and in a blind search it may become active again with probability $1-s$, if queried. We consider that the initiator does not become a stifler, so the number of active nodes will always be greater than zero.

This stifling behaviour will be modeled only for the blind search case, based on our approximative method. In order to model the smart search case, one has to discriminate between active and queried nodes, which leads to a multi-dimensional Markov chain which is not easily amenable to analysis.

Given that there are $i(r)$ inactive nodes at round r, we are interested in finding the mean number of inactive nodes at round $r+1$. This consists of the mean number of active nodes at round r that became inactive (excluding the initiator) and the mean number of inactive nodes at round r that remained inactive. Hence,

$$\begin{aligned} E[I(r+1)] =& (A(r)-1)s + I(r)\left[\left(1-\frac{k}{N-1}\right)^{A(r)} + \left[1-\left(1-\frac{k}{N-1}\right)^{A(r)}\right]s\right] \\ =& (A(r)-1)s + i(r)\left[s + (1-s)\left(1-\frac{k}{N-1}\right)^{A(r)}\right] . \end{aligned}$$

Assuming $I(r) = i(r)\ \forall r$, and using again that $(1-\frac{k}{N-1})^{A(r)} \approx e^{-A(r)(\frac{k}{N-1}+\frac{k^2}{2(N-1)^2})}$, $A(r) = N - I(r)$, we finally obtain the deterministic approximation

$$A(r+1) = 1 + (N-1)(1-s) - (N-A(r))(1-s)e^{-A(r)(\frac{k}{N-1}+\frac{k^2}{2(N-1)^2})} , \qquad (7.7.0.1)$$

with $A(1) = 1$.

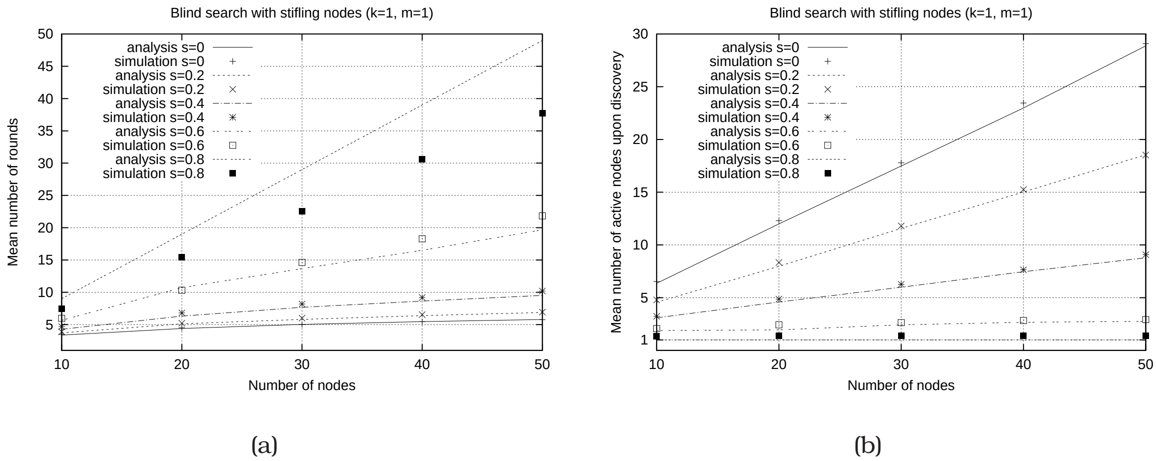


Figure 7.6: Mean number of rounds and mean number of active nodes upon discovery with $k = 1$, $m = 1$ for the blind search algorithm with stiflers.

From there we can follow similar steps as in Section 7.5 to find performance measures of interest.

Results based on simulation and the analytical approximation are shown in Fig. 7.6, for different values of parameters m, k, and the stifling probability s. As s increases, the performance of the search algorithm deteriorates.

We observe that the approximate model follows very well the simulated behaviour, except for larger deviations in the mean number of rounds when the stifling probability is high. This is mainly due to the higher relative error that results from the rounding operation (see the analysis of Section 7.5). As the stifling probability gets higher, the number of active nodes in the network is rounded to one in the model, and therefore the mean number of rounds approaches the inverse of the probability of a successful query of this node (geometric distribution). For example, when $k = 1$, the mean number of rounds approaches the value of $(N-1)/m$.

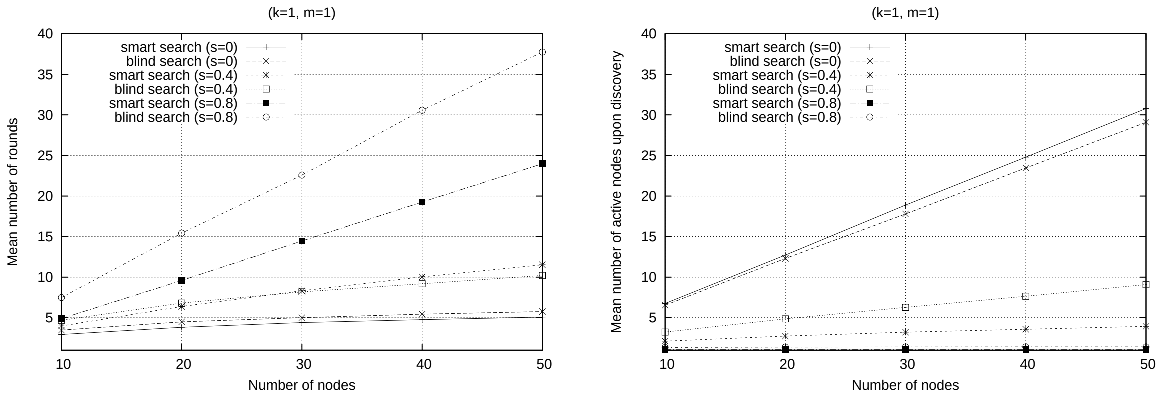
A comparison of the speed of blind search between the stifling and plain non-cooperative case shows that the search algorithm performs worse in the presence of stifler nodes. We may remark from the results that the relative increase in the number of rounds when nodes behave as stiflers – rather than as plain non-cooperative nodes – becomes greater when the number of nodes in the network increase, or when the stifling probability increases. Since stifling is opposite to cooperation, it makes sense to examine dual values of s, c, i.e. such that $s = 1 - c$ holds. For $k = 1\,, m = 1$, and $N = 50$, the mean number of rounds is increased by 9% for nodes that behave as stiflers with probability $s = 0.2$, compared to the case of plain non-cooperative nodes with $c = 0.8$. The relative increase is 5% when $N = 10$. When $s = c = 0.5$, the corresponding relative increase is much greater, and amounts to 78%.

This difference becomes smaller as the search becomes faster, i.e. when increasing either the k or m parameters. Regarding the relative influence of the parameters k, m to the efficiency of the search the same observations hold, as in all previous cases.

The mean number of active nodes calculated here is the mean number of nodes that are active upon discovery of the file. We therefore do not count nodes which were previously active in the search. Hence, it should be noted that the mean number of active nodes displayed here is only indicative of the communication overhead, as it does not count the nodes that were active in intermediate rounds of the algorithm, and hence the corresponding communication costs. Generally, our findings show that this number is much smaller when compared to the plain non-cooperative case, where active nodes remain in that state until the end. For $k = 1\,, m = 1$, and $N = 50$, the number of active nodes upon discovery is decreased by 39% in the stifling case with $s = 0.2$, compared to the plain non-cooperative case with $c = 0.8$.

We have also conducted a series of simulations to see the performance of smart search in the presence of stifling nodes. The smart search algorithm only queries nodes that have not been queried in previous rounds, although all active nodes may become stiflers at any round. In Fig. 7.7, we show a comparison of smart search against blind search for $k = 1\,, m = 1$, with different values of the stifling probability and increasing number of nodes.

We observe that smart search can yield a reduction in the number of rounds to discover the file, which becomes significant for high values of the stifling probability. For $s = 0.8$ and $N = 50$, the relative reduction is 37%. However, the interesting thing is that for smaller values it may also yield a slight increase (see the curves in Fig. 7.7(a) for $s = 0.4$). This seemingly unorthodox result is explained by the fact that since previously queried nodes are not queried again, once they become stiflers, they permanently remain in that state and do not participate again in the search. Thus trying to implement a “smarter” algorithm may also result in reducing the effective number of searchers, thereby



(a) (b)

Figure 7.7: Comparison of smart search against blind search with stiflers for $k = 1\,, m = 1$, with different values of the stifling probability and increasing number of nodes.

slowing down the search.

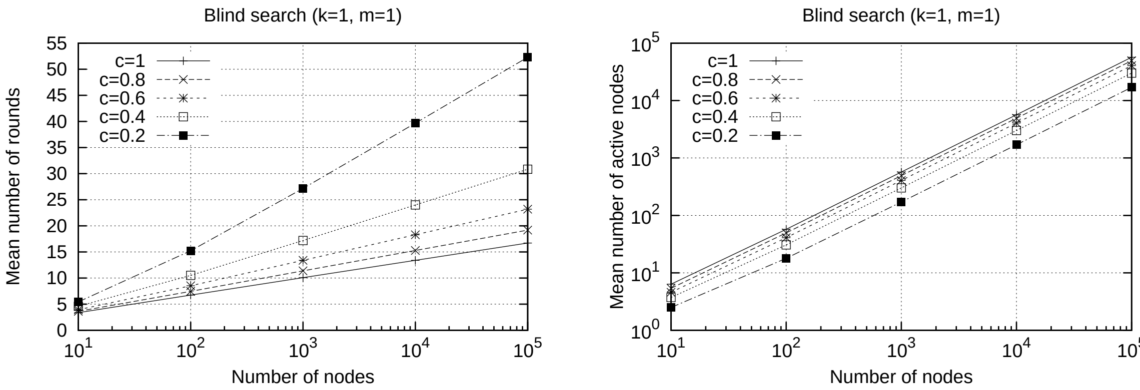
Fig. 7.7(b) shows that the mean number of active nodes upon discovery is smaller for smart search than for blind search, when $s > 0$ (as opposed to the plain non-cooperative case, cf. Fig. 7.4,7.5). The relative decrease becomes greater for medium values of s ($s = 0.4$ in Fig. 7.7(b)). For high values of this probability, the mean number of active nodes approaches one and differences become negligible. On the other hand, from Fig. 7.7(a) the highest gains in speed occur for $s = 0.8$. Therefore the highest gain in speed does not imply the highest reduction in redundant active nodes, and vice-versa.

7.8 Scaling performance of blind search

The low-complexity approximate model we have developed for the blind search algorithm enables us to study its performance for networks with very large numbers of nodes. We have taken results for networks with up to 10^5 nodes, for both behavioural profiles: plain non-cooperative and stifling. In Fig. 7.8, we plot the mean number of rounds and the mean number of active nodes as a function of N for the case of plain non-cooperative nodes, while in Fig. 7.9, similar results are taken for the case of stifling nodes.

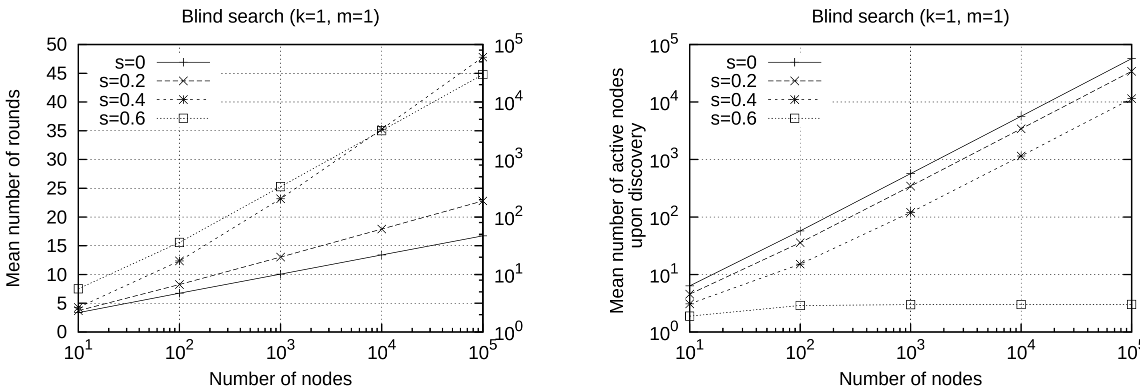
The x-axis in all plots is in $\log$ scale. Fig. 7.8(b) and 7.9(b) are in log-log scale. In Fig. 7.9(a), the curve for $s = 0.6$ is scaled with respect to the right y-axis.

From these results, we observe that the scaling performance of blind search is re-



(a) (b)

Figure 7.8: Scaling performance of blind search in the plain non-cooperative case, for $k = 1$, $m = 1$: (a) mean number of rounds, (b) mean number of active nodes.



(a) (b)

Figure 7.9: Scaling performance of blind search in the stifling case, for $k = 1$, $m = 1$: (a) mean number of rounds, (b) mean number of active nodes upon discovery.

markably simple. In the plain-non cooperative case, the mean number of rounds increases linearly with $\log N$, while the mean number of active nodes increases linearly with N. This is true for almost the whole range of values of c. (A more accurate estimate would be to consider a piece-wise linear function, with a slightly smaller slope for $N < 100$.) In the case of stifling nodes, the behaviour is similar for values of $s < 0.5$ approximately. For large values of s, the mean number of rounds increases proportionately to the increase in the number of nodes, while the mean number of active nodes upon discovery approaches 1.

Based on the observed behaviour, we can easily derive mathematical formulas for these functions, based on least squares fitting. For example, for $k = 1\,, m = 1$ and $c = 1$, $E[r] = 0.629 \log N + 0.057$, and $E[A] = 0.567N + 0.584$.

7.9 Overview of major results

In this section we provide a summary of the major results of the gossiping algorithms for the dissemination and search processes.

For both the blind and smart search cases, we showed that the mean number of active nodes and the mean number of rounds roughly increase linearly with the number of nodes in the network and its logarithm, respectively. For the blind search algorithm, we were able to confirm this behaviour for very large numbers of nodes, using the approximate model which has very low complexity. This is consistent with the logarithmic increase in the expected number of rounds of any dissemination process in a general graph [Mieghem, 2005], pp. 342.

The probability to inform all nodes in the network as a function of the number of rounds of the gossiping process shows a 0-1 bi-modal behaviour, increasing very abruptly from 0 to 1 after a critical round value, which increases with the size of the network.

Decreasing the cooperation probability of nodes increases the mean number of rounds to inform all nodes in a logarithmic fashion, for both the blind and smart search schemes. It also significantly degrades the speed of the search process. On the other hand, as the number of nodes involved in the gossiping process decreases, the communication overhead also decreases. With respect to the search speed, stifling (*i.e.*, ceasing to participate in the search with a specified probability in every round) has an even greater negative impact than the presence of nodes who do not participate in the search from the beginning.

Another important observation concerns the relative impact on the search of the number of queried peers by each node, and of the number of copies of the data object in the network. The increase of both these parameters increases the speed of discovery. The relative increase is greater when the number of queried peers increases, in both the blind and smart search algorithms. However, the corresponding increase in the number of active nodes that (in most cases) ensues is inappropriate, and hence it is preferable to keep the value of this parameter very small. The gossip-based search algorithm performs better when the requested data object is spread to many nodes in the network.

Finally, the overall results illustrate that when comparing the blind and the smart gossiping processes, it appears that there is a non-negligible improvement in the number of rounds to inform the network, or to find a certain object. This improvement becomes, however, less significant when the number of nodes contacted in every round increases, or when multiple copies of a searched file exist in the network. Further, the communication overhead produced by the amount of nodes involved in a smart search process usually increases, since more nodes are activated during the search. This is more pronounced for a larger node cooperation probability. In the presence of stifling nodes, a smart search may also increase the number of rounds to find an object, since nodes that become inactive are not queried again. Therefore, in view of the overhead incurred for informing all active nodes of the identities of queried nodes, a simple blind dissemination or search scheme may be preferable.

7.10 Comparison between the approximate and the exact model for the blind search algorithm

We compare the approximate model for the blind search algorithm with the exact model developed in [Tang *et al.*, 2011], when the cooperation probability $c = 1$. The two models are compared from the points of view of complexity and accuracy.

7.10.1 Comparison of complexity

We will compare the complexity of the two models based on the computational cost for deriving the probability of locating the file at a certain round r, given by (7.5.0.3) in the approximate and by (7.5.0.9) in the exact model. The computational cost is measured based on the number of elementary steps needed to derive the location probability, where each step consists of a small number of elementary operations (addition, subtraction, multiplication or division). We use the O-notation as the asymptotic upper bound of the complexity.

To compute (7.5.0.3), first computations of (7.5.0.2) and (7.5.0.1) need to be done.

Equation (7.5.0.2) involves two binomial coefficients, which can be computed in $O(kN)$ time using the well-known linear recursion formula.[5] The recursive formula (7.5.0.5) involves a small number of multiplications and additions, and an exponential function. The exponential can be calculated easily by splitting the exponent into integer and fractional parts (the latter can be computed within high accuracy with a few terms only in a Taylor expansion, see e.g. [Muller, 1997]). Hence each execution of the recursion has order one, and computing $A(r)$ or $\hat{A}(r)$ takes time $O(r)$. It holds that $\hat{A}(r) \leq N$. Therefore, the computation of (7.5.0.1) and (7.5.0.3), given their input parameters, takes time $O(N)$ and $O(r)$ respectively, since in the first case it involves the computation of $\hat{A}(r)$ polynomials, and in the second of r polynomials, both of degree one. Therefore, the total complexity of deriving the probability of locating the file at round r, using the approximate model is $O(kN + r)$.

To find the probability to find at least one copy of the file with the exact modeling, we need to calculate the r-th power of the transition matrix Q and then solve equations (7.5.0.9),(7.5.0.10) sequentially. The first computation involves the multiplication of an $N \times N$ transition probability matrix, with complexity at worst $O\left(N^3\right)$. For sufficiently large r, we can consider the sequence of matrices Q, Q^2, Q^4, ..., Q^{2^k}, instead of computing the sequence Q, Q^2, ..., Q^r. Since the former one converges considerably faster compared with the latter one. Therefore, we can compute the matrix power in $O\left(\ln(r)N^3\right)$ steps. The computation of (7.5.0.9) involves two binomial coefficients, namely, $\binom{N-i}{m}$ and $\binom{N-1}{m}$, which have complexity $O\left(mN\right)$. Therefore, it takes $O\left(mN^2\right)$ steps to solve (7.5.0.9). The total computational complexity is thus dominated by the complexity to compute the matrix power, which is $O\left(\ln(r)N^3\right)$ in our case.

7.10.2 Comparison of accuracy

We next compare the relative accuracy of the approximate model for calculating the mean number of steps to find at least one copy of the file and the mean number of nodes activated in the search. The relative accuracy is calculated as $(1 - |exact - approx|/exact)100\%$, and is output with two decimal digits. It is reminded that the comparison is done when the cooperation probability is one. Results are shown in Table 7.1

[5] For $j \geq i > 0$, $\binom{j}{i} = \binom{j-1}{i} + \binom{j-1}{i-1}$, with $\binom{j}{0} = \binom{j}{j} = 1$.

below, where N stands for the total number of nodes in the network (including the initiator).

Table 7.1: Relative accuracy (%) of the approximate model for blind search.

(a) Mean number of rounds

	$N = 10$	$N = 20$	$N = 30$	$N = 40$	$N = 50$
$k = 1$, $m = 1$	94.07	97.23	98.94	99.60	100
$k = 1$, $m = 3$	93.97	97.00	98.45	99.18	99.75
$k = 3$, $m = 1$	96.16	98.77	99.77	99.89	99.70

(b) Mean number of nodes activated in the search

	$N = 10$	$N = 20$	$N = 30$	$N = 40$	$N = 50$
$k = 1$, $m = 1$	92.60	95.78	96.12	96.34	97.81
$k = 1$, $m = 3$	95.43	95.97	94.81	95.25	96.58
$k = 3$, $m = 1$	86.69	90.51	93.63	94.98	95.97

These results confirm that the approximation becomes more accurate when the number of nodes in the network increases. The greatest inaccuracy is observed for a relatively large – compared to N – number of queried neighbours k, as was also indicated in Fig. 7.4(f). Generally, the model proves to be credible, with a relative accuracy that is higher than 95% in the majority of the above cases.

Chapter 8

Conclusion

8.1 Summary of main results and possible extensions

Here we present our main results and possible extensions for future work for each chapter:

In Chapter 2 we introduced and studied a distributed algorithm for object replication in a group of nodes in a content network, in which each node selfishly decides which objects to replicate from a distant server based on other nodes' placements. The algorithm accounts for selfish behaviour of autonomous nodes, as well as churn phenomena, *i.e.*, nodes that are not always available or operational. It needs minimal information in order to be applied, consisting of the placements of other nodes in the replication group and a probability estimate of their reliability (availability). The reliability information can be calculated in a distributed manner, or maintained in a central database in the network that nodes query prior to making their decision. In the majority of test cases we examined, it decreases both individual and total access costs, compared to the greedy local or the churn-unaware strategy. Furthermore, by easier satisfying participation constraints and providing fairer access costs to nodes according to their reliability, it incites nodes to cooperate.

This work can be extended to study the behaviour of the algorithm in cases where there exist objects of different sizes or where each node has a subjective estimate for the reliability of other nodes. Another possible extension is to consider communication capabilities of nodes, and the possibility of each one to lie about or distort its placement. In such situations, it is questionable whether a good strategy can be found that would encourage autonomous nodes to cooperate in replicating objects. Finally, simulations on more realistic scenarios are needed in order to verify the applicability of these schemes, taking into consideration all parameters.

In Chapter 3 our results suggested that the level of similarity of nodes' preferences within a social group is key for deciding which content placement strategy (*i.e.*, what kind of behaviour) to follow within a replication group. Altruism emerges as a win-win behaviour only in tight social groups as long as the implementation cost is not an issue: it minimizes the content access cost not only collectively for the whole group but also for each individual node. As *tightness* decreases, the collective group gain under the altruistic and self-aware cooperative strategy (*i.e.*, the cooperative strategy studied also in Chapter 2) fades out, while certain nodes may be mistreated when behaving altruistically. Therefore, and considering also its low complexity, the selfish strategy becomes more attractive. In summary, our evaluation shows that *the benefits of cooperation increase with the group tightness*.

With these in mind and on a more practical note, tightness can be used as a decision criterion when: a) choosing content placement strategies under given group membership; or, more broadly, b) for carrying out performance-driven group management operations such as group formation/merging/splitting.

A possible line of future work is to collect user preference data from a real social network and test our conclusions against more general patterns of dissimilarity. Further, the preliminary results of the effect of new nodes in the access cost of a social group can be helpful in the design of an admission control scheme based on tightness values. Basically, a new node would be accepted into the social group only if its entrance would not overly decrease the tightness of the group. This could potentially decrease the cost of administering and maintaining social groups, since tightness is generally much easier to calculate than the actual cost quantity that is of interest (the total access cost in our paradigm).

Additionally, it is interesting to extend this study to more than one replication groups, with different interest profiles of nodes in each group, and study the effect of node mobility between groups in the optimal placement strategy a node should follow.

In Chapter 4, we proposed a framework called ISCoDe for the clustering of users (nodes) according to common interests. Communities in online social networks do not usually exhibit a high degree of interest similarity; thus the framework can be used as a guide for the formation of more interest-coherent communities in online social networks. We have investigated two similarity metrics, Proportional Similarity (PS) and Inverse Kullback-Leibler distance (InvKL), for weighting edges according to the similarity of preferences of node pairs in a virtual graph. We then used a greedy agglomerative algorithm to iteratively find communities with highest modularity. Our results suggest that both metrics produce reasonable partitions for strong community structure. However, the PS metric is more sensitive in identifying partitions in networks with less apparent interest community structure. On the other hand, InvKL has a higher resolution in networks of nodes with highly similar interests, *i.e.*, it is able to identify smaller-sized communities.

A question that naturally arises concerns the relation between partitioning with respect to modularity maximization and partitioning with the objective to maximize the benefit by object exchange. Our results have shown that the interest distribution mappings influence the discernibility of the framework, and thus there must be a positive correlation between modularity and tightness. A formal proof of this is non-trivial, however it would be of extreme importance in designing effective and less costly community detection or formation algorithms.

We can also consider a distributed scheme for dynamic interest community formation. This scheme should not be computationally expensive so that it can be run by mobile nodes in a short time, should not rely on global information, but only on the local view each node has of the network: its adjacent nodes, it's local community nodes, and nodes adjacent to the boundary of it's community. Each node could examine the nodes in its local view (which can be expanding, *i.e.*, a node initially knows its immediate neighbours, then its neighbours' neighbours, *etc.*) and use a local modularity condition to see if they could be included in its local community (such as the variation of fitness, see [Lancichinetti *et al.*, 2009]). The local communities formed in this way can be compared against the communities formed by a centralized algorithm, such as the one developed in

Chapter 4, to evaluate the performance of the distributed algorithm.

In Chapter 5 we introduced a framework for disseminating content in a networking environment comprised of interest and locality-induced social groups. Node encounters or node visits to locality-induced social groups are considered to be the mechanism for content exchanges and ultimately content dissemination. Two content storage and dissemination strategies have been introduced and investigated: the selfish and the cooperative one. When deciding on the content to store locally, selfish nodes store locally only contents of their interests, whereas cooperative nodes also consider the interests of other nodes that are likely to be encountered in the future. By considering both the usability and discoverability of content, we investigated the performance of the two strategies and explored the conditions under which the cooperative strategy can enhance the content dissemination process compared to a selfish one. A number of results have been derived showing the comparative performance of the two strategies, considering a Zipf distribution over a set of interest classes, and another Zipf distribution over a set of localities. When nodes have different interest distributions and visit different localities, the results showed that cooperation is not beneficial, unless the usability of other nodes' contents is higher than the usability of one's own contents. On the other hand, when nodes exhibit the same preferences, mobility helps in the dissemination of content, and nodes attain an improvement even if they are selfish. The performance of the cooperative strategy is of course much higher and further improves, up to a point when the total number of localities or content classes increase.

In this work we considered all nodes to exhibit the same behaviour, either all selfish or all cooperative. The case where there is a mixture of selfish and cooperative is therefore a reasonable extension. Further, it is interesting to study cases where nodes have different degrees of mobility. For example, it is interesting to study the effect of the presence of a number of static nodes in each locality, either cooperative or selfish. More specifically, one could examine, in a game-theoretic context, whether the presence of a number of cooperative nodes in each locality could suffice to incite all nodes to cooperate, and thus sustain an equilibrium with minimum effort, as in [Lai & Gamal, 2008].

In Chapter 6 we studied a simple model of a nomadic sensor network, and derived conditions under which rational U-nodes will exhibit cooperative behaviour on legs of the network, given that a cooperative U-node will actually transmit acquired sensor data with a probability p when it meets a selfish U-node. The parameter p may either be a fixed parameter of the system, or have a value everyone agrees to. We think it should be greater than zero, to allow for dissemination of information even if U-nodes make irrational or erroneous decisions. Cooperative equilibria can exist in the whole or parts of the network, *i.e.*, one or more legs in the network.

The model we studied admits several simplifying assumptions that sacrifice reality for analytical tractability. However, the assumptions regarding the knowledge each U-node may have are realistic; furthermore, one should be aware of the fact that mobile devices, advanced as they may be, still have limited computational capabilities, as well as limited interactions with an intelligent human user. Hence such a simple model can indeed be used by U-nodes to decide whether or not they will be cooperative.

One drawback however, is that the equilibria produced by this model are not completely self-enforced. The cooperation probability is the same for all U-nodes, and hence it should be set by another authority or by a common agreement between the players. The same analysis can be followed to derive cooperative equilibrium conditions when the cooperation probability is different for each U-node. In a more realistic scenario the cooperation probability would correspond to a reputation value of each U-node to be cooperative. Reputation values can be obtained from interactions between U-nodes (examples of distributed algorithms can be found in [Srinivasany *et al.*, 2011]).

Extensions of this work include studying cooperation equilibria in various graph topologies. Further research issues involve relaxing some of the assumptions used: First, to find cooperation conditions when the number of U-nodes is not known a priori to all players, but is a random variable. Secondly, to describe the dynamics of the game, by studying it either as a repeated game or as an evolutionary game. It is anticipated that an equilibrium can also be sustained in a repeated game, where punishments occur for selfish nodes in subsequent rounds of the game. By the other approach, we can study how the strategies of the players would evolve until a stable situation has been reached.

In Chapter 7, we focused on modeling the process of gossip-based message dissemination and search under the assumption of uniform neighbour selection over the entire set of nodes in the network. We were able to model parallel dissemination and search by multiple nodes, where each node may contact multiple peers in each gossiping round. We used both an exact analytical Markov chain model, as well as an approximate model with much lower computational complexity. From the results several practical performance metrics of interest and important design parameters were obtained. When disseminating or searching for content, the smart selection algorithm where each node has full information about the informed or queried nodes in each round is, in nature, more effective than the blind selection scheme where no information is available. For instance, to inform the entire network, the smart selection scheme may only need half of the gossiping rounds compared with the blind selection algorithm. However, there is a tradeoff between speed, cost and redundancy between these two schemes. A blind search process with smaller cost of managing information could compensate for the smaller speed in disseminating or finding a certain content, by querying more nodes in each round. Further, when nodes are cooperative, they may be more loaded with redundant messages by a smart gossiping scheme, which might, as a consequence, incite them to be non-cooperative. What's more, when stifling nodes exist, a smart search algorithm may be less fast, since it does not prompt nodes that have become inactive to participate again in the search. Finally, our results about content search suggest that the search process is less costly and about the same as fast when several content replicas reside in the network and each node queries only one neighbour at each round, rather than when fewer replicas exist and we try to increase the hit rate by increasing the number of gossiping targets contacted in each round. This is again due to the queries' redundancy created by the second approach, and could be another useful design guideline.

Future research issues we envisage are mostly related to the performance evaluation of the gossip-based search algorithm. One direction of research is to examine the efficiency of the search in different types of networks. From the network in the form of a complete graph that we studied here, we can pass to more general graphs that are often met, such as Erdos-Renyi graphs, or graphs with power-law degree distribution (scale free networks). Performance results in such networks where each node has a different degree will give more realistic evidence of its efficiency and applicability. Another research direction is to compare the performance of the algorithm with different distributed search schemes, in terms of speed and implementation cost. An interesting scheme that is similar to gossip-based spreading but has a different philosophy is that of a *multiple random walk*. In such a scheme, a certain node starts a random walk in the network, and during its course activates other nodes, that also start a random walk. This scheme is expected to have a similar performance in both search and dissemination, and may be applied in many interesting scenarios in communications networks.

8.2 Other research directions

In concluding this thesis, we discuss some research problems that reveal novel and fascinating directions of research in the field of information dissemination and retrieval in modern networks. These are basically related to the problem of information explosion and information overload that we discussed in our introductory chapter.

In dealing with this problem, it seems clear to us that we have to be more systematically concerned with the contents of the objects exchanged in the network. Content characteristics include semantic and freshness characteristics. The former are important for determining the relevance of an information object (owned by some node) with a certain data query, while freshness is important for determining the current value of an object, *i.e.*, its depreciation due to its age [Bouzeghoub, 2004]. By analyzing these characteristics, we may design an information filtering algorithm that makes part of a dissemination or retrieval scheme, and assists in maximizing the value we attain from dissemination or retrieval.

There is an abundance of research in information filtering and retrieval in database systems [O'Riordan & Sorensen, 1999, Turtle & Croft, 1992, Callan, 2000], that may assist us in this effort. The problems we have to look at are:

- how to represent the contents in mini or medium sized-databases of autonomous network nodes.
- how to represent a content query or the interest profile of a user.
- how to compute the probability with which a certain query at a certain node is satisfied, or that incoming data matches the interest profile of a user.

- how to incorporate the mobility of nodes in a realistic model, so as to derive the overall probability for each node to find or receive valuable content in the network.

More detailed guidelines could then be derived on how to propagate content or queries in the network, that do not only consider the communication cost of the algorithm, but also the value of the disseminated or retrieved content.

However, even if we design an efficient algorithm for content dissemination and retrieval that takes into account all costs as well as the value of the content, its application is likely to be difficult in view of problems such as processing complexity, partial and short-time connectivity, information explosion, or partial knowledge that a node will have of the network or of the content itself. In simpler words, a node may not have the time, processing power, or knowledge required to properly evaluate the abundance of content received, or the best way to disseminate and search for content. Therefore, it seems unavoidable to have to rely to a smaller or larger extent on heuristics. The addition of more intelligence in user devices has recently sparkled the interest for using cognitive heuristics in networks with autonomous nodes[1]; these are algorithms which try to adapt cognitive processes in order to solve complex communication problems. Humans, but also some animals, have a remarkable ability to take good decisions under limited knowledge and in a short time. By imitating the processes of making such decisions, we may arrive at efficient algorithms to deal with such complex problems. A well-known inference related to the problem of information overload is the less-is-more effect [Goldstein & Gigerenzer, 2008], according to which less information is better than more information when making decisions, when we are not able to recognize or process the extra information.

Cognitive heuristics rely on the proper identification of some basic attributes or cues, which are the most fundamental for the situation at hand, have less uncertainty and can lead to a fast and good solution. Decisions are then made solely based on these cues, without considering other parameters that may exist, but are less important. A recent application of cognitive heuristics for data dissemination can be found in [Conti *et al.*, 2011]. In content dissemination and retrieval, a relevant problem would be to find efficient cues to describe the appropriateness of each node as a source, recipient or forwarder of messages, or to describe the similarity between information objects, their freshness, and

[1] Recognition project, http://www.recognition-project.eu/

usability.

The above-described research directions are indicative of the adaptation of telecommunications networks and services to the human environment, which will undoubtedly continue as technology evolves.

Bibliography

[Abolhasan, 2004] Abolhasan, M. (2004). A review of routing protocols for mobile ad hoc networks. Ad Hoc Networks *2*(1), 1–22.

[Allen & Shoard, 2005] Allen, D. & Shoard, M. (2005). Spreading the load: mobile information and communications technologies and their effect on information overload. Information Research *10*(2).

[Allen *et al.*, 2010] Allen, S. M., Colombo, G., & Whitaker, R. M. (2010). Cooperation through self-similar social networks. ACM Trans. Auton. Adapt. Syst. *5*, 4:1–4:29.

[Avrachenkov *et al.*, 2007] Avrachenkov, K., Nemirovsky, D., & Pham, K. S. (2007). A survey on distributed approaches to graph based reputation measures. in *Proc. of International Workshop on Tools for solving Structured Markov Chains (SMCtools)*.

[Basar & Olsder, 1999] Basar, T. & Olsder, G. (1999). (Society for Industrial and Applied Mathematics (SIAM)). 2nd edit.

[Belkin & Croft, 1992] Belkin, N. J. & Croft, W. B. (1992). Information filtering and information retrieval: Two sides of the same coin?. Commun. ACM *35*(12), 29–38.

[Bertsekas, 1982] Bertsekas, D. (1982). Dynamic behavior of shortest path routing algorithms for communication networks. IEEE Transactions on Automatic Control *27*(1), 60–74.

[Biswas & Morris, 2004] Biswas, S. & Morris, R. (2004). Opportunistic routing in multi-hop wireless networks. SIGCOMM Comput. Commun. Rev. *34*(1), 69–74.

[Boldrini *et al.*, 2007] Boldrini, C., Conti, M., Jacopini, J., & Passarella, A. (2007). HiBOp: a history based routing protocol for opportunistic networks. in *IEEE International Symposium on World of Wireless, Mobile and Multimedia Networks (WoWMoM 2007)* pp. 1–12.

[Boldrini *et al.*, 2008] Boldrini, C., Conti, M., & Passarella, A. (2008). Contentplace: Social-aware data dissemination in opportunistic networks. in *The 11-th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM'08).*

[Borst *et al.*, 2010] Borst, S. C., Gupta, V., & Walid, A. (2010). Distributed caching algorithms for content distribution networks. in *INFOCOM* pp. 1478–1486. IEEE.

[Bouzeghoub, 2004] Bouzeghoub, M. (2004). A framework for analysis of data freshness. in *Proceedings of the 2004 international workshop on Information quality in information systems* IQIS '04 pp. 59–67 New York, NY, USA. ACM.

[Brandes *et al.*, 2008] Brandes, U., Delling, D., Gaertler, M., Gorke, R., Hoefer, M., Nikoloski, Z., & Wagner, D. (2008). On modularity clustering. IEEE Transactions on Knowledge and Data Engineering *20*(2), 172–188.

[Brandes & Erlebach, 2005] Brandes, U. & Erlebach, T. (2005). Network analysis: Methodological foundations (lncs). volume 3418. Springer, Heidelberg.

[Breslau *et al.*, 1999] Breslau, L., Cao, P., Fan, L., Phillips, G., & Shenker, S. (1999). Web caching and Zipf-like distributions: Evidence and implications. in *Proc. IEEE INFOCOM* pp. 126–134.

[Buttyán *et al.*, 2007] Buttyán, L., Dóra, L., Félegyházi, M., & Vajda, I. (2007). Barter-based cooperation in delay-tolerant personal wireless networks. in *WoWMoM* pp. 1–6.

[Buttyán *et al.*, 2010] Buttyán, L., Dóra, L., Félegyházi, M., & Vajda, I. (2010). Barter trade improves message delivery in opportunistic networks. Ad Hoc Netw. *8*, 1–14.

[Caesar & Rexford, 2005] Caesar, M. & Rexford, J. (2005). BGP routing policies in isp networks. Network, IEEE *19*(6), 5–11.

[Callan, 2000] Callan, J. (2000). Distributed information retrieval. in *Advances in Information Retrieval* pp. 127–150. Kluwer Academic Publishers.

[Carreras *et al.*, 2005] Carreras, I., Chlamtac, I., Woesner, H., & Zhang, H. (2005). Nomadic sensor networks. in *Proc. EWSN* pp. 166–175.

[Chuah *et al.*, 2010] Chuah, M., Yang, P., & Hui, P. (2010). Cooperative user centric information dissemination in human content-based networks. in *Proceedings of the 2010 IEEE 16th International Conference on Parallel and Distributed Systems* ICPADS '10 pp. 794–799 Washington, DC, USA. IEEE Computer Society.

[Clauset *et al.*, 2004] Clauset, A., Newman, M. E. J., & Moore, C. (2004). Finding community structure in very large networks. Physical Review E , 1–6.

[Cohen & Shenker, 2002] Cohen, E. & Shenker, S. (2002). Replication strategies in unstructured peer-to-peer networks. SIGCOMM Comput. Commun. Rev. *32*, 177–190.

[Cohen & Havlin, 2003] Cohen, R. & Havlin, S. (2003). Scale-free networks are ultrasmall. Phys. Rev. Lett. *90*, 058701.

[Cohen *et al.*, 2002] Cohen, R., Havlin, S., & Ben-Avraham, D. (2002). Structural properties of scale free networks in *Handbook of graphs and networks* (Bornholdt, S. & Schuster, H. G., eds.) (Wiley-VCH).

[Conti *et al.*, 2011] Conti, M., Mordacchini, M., & Passarella, A. (2011). Data dissemination in opportunistic networks using cognitive heuristics. in *WOWMOM* pp. 1–6.

[Cormen *et al.*, 2001] Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2001). (The MIT Press). 2nd revised edit.

[Crowcroft *et al.*, 2003] Crowcroft, J., Gibbens, R., Kelly, F., & Ostring, S. (2003). Modelling incentives for collaboration in mobile ad hoc networks. in *WiOpt*.

[Denuit & Bellegem, 2001] Denuit, M. & Bellegem, S. V. (2001). On the stop-loss and total variation distances between random sums. Statistics & Probability Letters *53*(2), 153–165.

[Derhab & Badache, 2009] Derhab, A. & Badache, N. (2009). Data replication protocols for mobile ad-hoc networks: A survey and taxonomy. Communications Surveys and Tutorials Journal *11*.

[Dousse *et al.*, 2002] Dousse, O., Thiran, P., & Hasler, M. (2002). Connectivity in ad-hoc and hybrid networks. in *INFOCOM* volume 2 pp. 1079–1088.

[Duch & Arenas, 2005] Duch, J. & Arenas, A. (2005). Community detection in complex networks using extremal optimization. Physical Review E *72*(2), 027104+.

[Eugster *et al.*, 2004a] Eugster, P., Guerraoui, R., Kermarrec, A.-M., & Massoulié, L. (2004a). From epidemics to distributed computing. IEEE Computer *37*(5), 60–67.

[Eugster *et al.*, 2003] Eugster, P. T., Guerraoui, R., Handurukande, S. B., Kouznetsov, P., & Kermarrec, A.-M. (2003). Lightweight probabilistic broadcast. ACM Trans. Comput. Syst. *21*, 341–374.

[Eugster *et al.*, 2004b] Eugster, P. T., Guerraoui, R., Kermarrec, A. M., & Massoulie, L. (2004b). Epidemic information dissemination in distributed systems. Computer *37*(5), 60–67.

[Fan *et al.*, 2007] Fan, Y., Li, M., Zhang, P., Wu, J., & Di, Z. (2007). The effect of weight on community structure of networks. Physica A: Statistical Mechanics and its Applications *378*(2), 583–590.

[Félegyházi & Hubaux, 2006] Félegyházi, M. & Hubaux, J.-P. (2006). Game theory in wireless networks: A tutorial Technical Report LCA-REPORT-2006-002 EPFL.

[Félegyházi *et al.*, 2006] Félegyházi, M., Hubaux, J.-P., & Buttyán, L. (2006). Nash equilibria of packet forwarding strategies in wireless ad hoc networks.. IEEE Trans. Mob. Comput. *5*(5), 463–476.

[Feller, 1968] Feller, W. (1968). *An Introduction to Probability Theory and Its Applications* (Wiley).

[Fernández & Gómez, 2008] Fernández, A. & Gómez, S. (2008). Solving non-uniqueness in agglomerative hierarchical clustering using multidendrograms. J. Classif. *25*, 43–65.

[Fortunato & Barthélemy, 2007] Fortunato, S. & Barthélemy, M. (2007). Resolution limit in community detection. Proceedings of the National Academy of Sciences *104*(1), 36–41.

[Gadkari, 2008] Gadkari, A. (2008). Caching in the distributed environment. The Architecture Journal.

[Gil & Ratnakar, 2002] Gil, Y. & Ratnakar, V. (2002). Trusting information sources one citizen at a time. in *ISWC '02: Proceedings of the First International Semantic Web Conference on The Semantic Web* pp. 162–176 London, UK. Springer-Verlag.

[Girvan & Newman, 2002] Girvan, M. & Newman, M. E. (2002). Community structure in social and biological networks. Proc Natl Acad Sci U S A *99*(12), 7821–7826.

[Goldstein & Gigerenzer, 2008] Goldstein, D. G. & Gigerenzer, G. (2008). The recognition heuristic and the less-is-more effect volume 1 of *Handbook of Experimental Economics Results* (Elsevier).

[Good *et al.*, 2010] Good, H. B., de Montjoye, Y.-A., & Clauset, A. (2010). The performance of modularity maximization in practical contexts. PHYS.REV.E *81*, 046106.

[Gupta & Kumar, 2000] Gupta, P. & Kumar, P. R. (2000). The capacity of wireless networks. IEEE Transactions on Information Theory *46*(2), 388–404.

[Haas *et al.*, 2006] Haas, Z. J., Halpern, J. Y., & Li, L. (2006). Gossip-based ad hoc routing. IEEE/ACM Trans. Netw. *14*, 479–491.

[Jamin *et al.*, 2000] Jamin, S., Jin, C., Jin, Y., Raz, D., Shavitt, Y., & Zhang, L. (2000). On the placement of Internet instrumentation. in *INFOCOM* pp. 295–304. IEEE.

[Jin & Bestavros, 2002] Jin, S. & Bestavros, A. (2002). Small-world internet topologies: Possible causes and implications on scalability of end-system multicast Technical report.

[Kamvar *et al.*, 2003] Kamvar, S. D., Schlosser, M. T., & Garcia-Molina, H. (2003). The eigentrust algorithm for reputation management in p2p networks. in *Proceedings of the 12th international conference on World Wide Web* WWW '03 pp. 640–651 New York, NY, USA. ACM.

[Kangasharju *et al.*, 2001] Kangasharju, J., Roberts, J., & Ross, K. W. (2001). Object replication strategies in content distribution networks. in *Computer Communications* pp. 367–383.

[Karp *et al.*, 2000] Karp, R., Schindelhauer, C., Shenker, S., & Vocking, B. (2000). Randomized rumor spreading. in *Proceedings of the 41st Annual Symposium on Foundations of Computer Science* pp. 565– Washington, DC, USA. IEEE Computer Society.

[Kempe *et al.*, 2003] Kempe, D., Dobra, A., & Gehrke, J. (2003). Gossip-based computation of aggregate information. in *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science* FOCS '03 pp. 482– Washington, DC, USA. IEEE Computer Society.

[Kernighan & Lin, 1970] Kernighan, B. W. & Lin, S. (1970). An efficient heuristic procedure for partitioning graphs. The Bell system technical journal *49*(1), 291–307.

[Kullback, 1959] Kullback, S. (1959). *Information Theory and Statistics* (Wiley: New York).

[Kurose & Ross, 2004] Kurose, J. F. & Ross, K. W. (2004). (Addison Wesley).

[La *et al.*, 2010] La, C. A., Michiardi, P., Casetti, C., Chiasserini, C. F., & Fiore, M. (2010). A lightweight distributed solution to content replication in mobile networks. in *2010 IEEE Wireless Communication and Networking Conference* pp. 1–6. IEEE.

[La *et al.*, 2011] La, C.-A., Michiardi, P., Casetti, C., Chiasserini, C.-F., & Fiore, M. (2011). Content replication and placement in mobile networks. CoRR *abs/1102.3013*.

[Lai & Gamal, 2008] Lai, L. & Gamal, H. E. (2008). On cooperation in energy efficient wireless networks: the role of altruistic nodes. IEEE Transactions on Wireless Communications *7*(5-2), 1868–1878.

[Lancichinetti *et al.*, 2009] Lancichinetti, A., Fortunato, S., & Kertesz, J. (2009). Detecting the overlapping and hierarchical community structure of complex networks. New Journal of Physics *11*(3), 033015+.

[Laoutaris *et al.*, 2006] Laoutaris, N., Telelis, O., Zissimopoulos, V., & Stavrakakis, I. (2006). Distributed selfish replication. IEEE Trans. Par. Distr. Systems *17*(12), 1401–1413.

[Le Boudec & Vojnovic, 2005] Le Boudec, J.-Y. & Vojnovic, M. (2005). Perfect simulation and stationarity of a class of mobility models.. in *INFOCOM* pp. 2743–2754.

[Leff *et al.*, 1993] Leff, A., Wolff, J., & Yu, P. (1993). Replication algorithms in a remote caching architecture. IEEE Trans. Par. and Distr. Systems *4*(11), 1185–1204.

[Liben-Nowell *et al.*, 2002] Liben-Nowell, D., Balakrishnan, H., & Karger, D. (2002). Analysis of the evolution of peer-to-peer systems. in *21st ACM Symposium on Principles of Distributed Computing (PODC)* Monterey, CA.

[Lim *et al.*, 2006] Lim, S., Yu, C., & Das, C. (2006). Clustered mobility model for scale-free wireless networks. pp. 231–238.

[Loukopoulos *et al.*, 2002] Loukopoulos, T., Ahmad, I., & Papadias, D. (2002). An overview of data replication on the internet. in *ISPAN '02: Proceedings of the 2002 International Symposium on Parallel Architectures, Algorithms and Networks* pp. 31–36 Washington, DC, USA. IEEE Computer Society.

[Luenberger, 2008] Luenberger, D. G. (2008). (Springer). third edit.

[Malkin, 2000] Malkin, G. (2000). (Addison-Wesley).

[Marbach & Qiu, 2005] Marbach, P. & Qiu, Y. (2005). Cooperation in wireless ad hoc networks: a market-based approach. IEEE/ACM Trans. Netw. *13*, 1325–1338.

[Mieghem, 2005] Mieghem, P. V. (2005). *Performance Analysis of Communications Networks and Systems* (Cambridge University Press: New York, USA).

[Mitchell & Teague, 2003] Mitchell, J. C. & Teague, V. (2003). Autonomous nodes and distributed mechanisms. in *Proceedings of the 2002 Mext-NSF-JSPS international conference on Software security: theories and systems* ISSS'02 pp. 58–83 Berlin, Heidelberg. Springer-Verlag.

[Mori, 1997] Mori, R. D. (1997). *Spoken Dialogues with Computers* (Academic Press, Inc.: Orlando, FL, USA).

[Moy, 1998] Moy, J. T. (1998). *OSPF: Anatomy of an Internet Routing Protocol* (Addison-Wesley Longman Publishing Co., Inc.: Boston, MA, USA).

[Mulerikkal & Khalil, 2007] Mulerikkal, J. P. & Khalil, I. (2007). An architecture for distributed content delivery network. in *ICON* pp. 359–364. IEEE.

[Muller, 1997] Muller, J.-M. (1997). *Elementary functions: algorithms and implementation* (Birkhauser Boston, Inc.: Secaucus, NJ, USA).

[Musolesi *et al.*, 2005] Musolesi, M., Hailes, S., & Mascolo, C. (2005). Social networks based ad hoc mobility models. in *Proceedings of the 3rd UK-UbiNet Workshop*.

[Myers & Well, 2003] Myers, J. L. & Well, A. D. (2003). *Research Design and Statistical Analysis (2nd edition)* (Lawrence Erlbaum).

[Myerson, 1997] Myerson, R. B. (1997). (Harvard University Press).

[Nekovee *et al.*, 2007] Nekovee, M., Moreno, Y., Bianconi, G., & Marsili, M. (2007). Theory of rumour spreading in complex social networks. Physica A: Statistical Mechanics and its Applications *374*(1), 457–470.

[Newman, 2004a] Newman, M. (2004a). Fast algorithm for detecting community structure in networks. Physical Review E *69*, 066133.

[Newman, 2004b] Newman, M. E. J. (2004b). Analysis of weighted networks. Phys. Rev. E *70*(5), 056131.

[Newman, 2004c] Newman, M. E. J. (2004c). Detecting community structure in networks. The European Physical Journal B - Condensed Matter and Complex Systems *38*(2), 321–330.

[Newman & Girvan, 2004] Newman, M. E. J. & Girvan, M. (2004). Finding and evaluating community structure in networks. Physical Review E *69*(2), 026113+.

[Nishihara, 1997] Nishihara, K. (1997). A resolution of N-person prisoners' dilemma. Economic Theory *10*(3), 531–540.

[O'Riordan & Sorensen, 1999] O'Riordan, C. & Sorensen, H. (1999). Information retrieval: An overview Technical report Technical Report of the Dept. of IT, NUI, Galway.

[Padmanabhan *et al.*, 2008] Padmanabhan, P., Le, Gruenwald, V. A., & Atiquzzaman, M. (2008). A survey of data replication techniques for mobile ad hoc network databases. The VLDB Journal *17*, 1143–1164.

[Pathan & Buyya, 2006] Pathan, A. M. K. & Buyya, R. (2006). A taxonomy and survey of content delivery networks. Grid Computing and Distributed Systems GRIDS Laboratory University of Melbourne Parkville Australia *148*, 1–44.

[Pittel, 1987] Pittel, B. (1987). On spreading a rumor. SIAM J. Appl. Math. *47*, 213–223.

[Pothen *et al.*, 1990] Pothen, A., Simon, H. D., & Liou, K. P. (1990). Partitioning sparse matrices with eigenvectors of graphs. SIAM Journal on Matrix Analysis and Applications *11*(3), 430–452.

[Radicchi *et al.*, 2004] Radicchi, F., Castellano, C., Cecconi, F., Loreto, V., & Parisi, D. (2004). Defining and identifying communities in networks. Proceedings of the National Academy of Sciences of the United States of America *101*, 2658–2663.

[Rahman & Hailes, 1997] Rahman, A. A. & Hailes, S. (1997). A distributed trust model. in *NSPW '97: Proceedings of the 1997 workshop on New security paradigms* pp. 48–60 New York, USA. ACM.

[Reed & Long, 1996] Reed, B. & Long, D. D. E. (1996). Analysis of caching algorithms for distributed file systems. SIGOPS Oper. Syst. Rev. *30*, 12–21.

[Reichardt & Bornholdt, 2006] Reichardt, J. & Bornholdt, S. (2006). Statistical mechanics of community detection. Physical Review E *74*(1), 016110+.

[Royer & Toh, 1999] Royer, E. M. & Toh, C.-K. (1999). A review of current routing protocols for ad hoc mobile wireless networks. IEEE Personal Communications *6*(2), 46–55.

[Salem *et al.*, 2006] Salem, N. B., Buttyan, L., Hubaux, J.-P., & Jakobsson, M. (2006). Node cooperation in hybrid ad hoc networks. IEEE Transactions on Mobile Computing *5*, 365–376.

[Sarkar *et al.*, 2007] Sarkar, S. K., Basavaraju, T. G., & Puttamadappa, C. (2007). *Ad Hoc Mobile Wireless Networks: Principles, Protocols and Applications* (Auerbach Publications: Boston, MA, USA). 1st edit.

[Schrijver, 1998] Schrijver, A. (1998). (John Wiley & Sons).

[Scott, 2000] Scott, J. P. (2000). (SAGE Publications).

[Seredynski *et al.*, 2006] Seredynski, M., Bouvry, P., & Klopotek, M. A. (2006). Evolution of cooperation in ad hoc networks under game theoretic model. in *MobiWac* pp. 126–130.

[Shikfa *et al.*, 2009] Shikfa, A., Onen, M., & Molva, R. (2009). Privacy in content-based opportunistic networks. in *WAINA '09: Proceedings of the 2009 International Conference on Advanced Information Networking and Applications Workshops* pp. 832–837 Washington, DC, USA. IEEE Computer Society.

[Spyropoulos *et al.*, 2008] Spyropoulos, T., Psounis, K., & Raghavendra, C. S. (2008). Efficient routing in intermittently connected mobile networks: the multiple-copy case. IEEE/ACM Trans. Netw. *16*, 77–90.

[Srinivasan *et al.*, 2003] Srinivasan, V., Nuggehalli, P., Chiasserini, C., & Rao, R. (2003). Cooperation in wireless ad hoc networks. in *INFOCOM*.

[Srinivasany *et al.*, 2011] Srinivasany, A., Teitelbaum, J., Liang, H., Wu, J., & Cardei, M. (2011). Reputation and trust-based systems for ad hoc and sensor networks. to be published.

[Tang *et al.*, 2011] Tang, S., Jaho, E., Stavrakakis, I., Koukoutsidis, I., & Mieghem, P. V. (2011). Modeling gossip-based content dissemination and search in distributed networking. Comput. Commun. *34*, 765–779.

[Turtle & Croft, 1992] Turtle, H. R. & Croft, B. W. (1992). A comparison of text retrieval models. Comput. J. *35*(3), 279–290.

[Vahdat & Becker, 2000] Vahdat, A. & Becker, D. (2000). Epidemic routing for partially connected ad hoc networks. Technical Report CS200006.

[Vegelius *et al.*, 1986] Vegelius, J., Janson, S., & Johansson, F. (1986). Measures of similarity between distributions. Quality and Quantity *20*(4), 437–441.

[Wang *et al.*, 2003] Wang, J., Tsang, W. W., & Marsaglia, G. (2003). Evaluating Kolmogorov's distribution. Journal of Statistical Software *8*(18), 1–4.

[Wang & Crowcroft, 1992] Wang, Z. & Crowcroft, J. (1992). Analysis of shortest-path routing algorithms in a dynamic network environment. ACM Computer Communication Review *22*, 63–71.

[Watts & Strogatz, 1998] Watts, D. J. & Strogatz, S. H. (1998). Collective dynamics of "small-world" networks. Nature *393*, 440–442.

[Wellman, 1996] Wellman, B. (1996). For a social network analysis of computer networks: a sociological perspective on collaborative work and virtual community. in *Proceedings of the 1996 ACM SIGCPR/SIGMIS conference on Computer personnel research* SIGCPR '96 pp. 1–11 New York, NY, USA. ACM.

[Zhao & Tong, 2005] Zhao, Q. & Tong, L. (2005). Energy efficiency of large-scale wireless networks: proactive versus reactive networking. IEEE Journal on Selected Areas in Communications *23*(5), 1100–1112.