CONSTRUCTING LOW DIAMETER TOPIC-BASED PUB/SUB OVERLAY NETWORK WITH MINIMUM MAXIMUM NODE DEGREE

SINA LAYAZALI

JANUARY 2016

CONSTRUCTING LOW DIAMETER TOPIC-BASED PUB/SUB OVERLAY NETWORK WITH MINIMUM MAXIMUM NODE DEGREE

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE
OF
ÇANKAYA UNIVERSITY

BY
SINA LAYAZALI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF
COMPUTER ENGINEERING

JANUARY 2016

Title of Thesis: **Constructing Low Diameter Topic-Based Pub/Sub Overlay Network With Minimum Maximum Node Degree**

Submitted by **Sina LAYAZALI**

Approval of the Graduate School of Natural and Applied Sciences, Çankaya University

Prof. Dr. Halil Tanyer EYYUBOĞLU
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Müslim BOZYİĞİT
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Prof. Dr. Müslim BOZYİĞİT
Supervisor

Examination Date: 29.01.2016

Examining Committee Members

Prof. Dr. Müslim BOZYİĞİT	(Çankaya Univ.)	
Assist. Prof. Dr. Emre SERMUTLU	(Çankaya Univ.)	
Assist. Prof. Dr. Kasım ÖZTOPRAK	(Karatay Univ.)	

STATEMENT OF NON-PLAGIARISM PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name	:	Sina LAYAZALI
Signature	:	
Date	:	29.01.2016

ABSTRACT

Constructing Low Diameter Topic-Based Pub/Sub Overlay Network With Minimum Maximum Node Degree

LAYAZALI, Sina
M.Sc., Department of Computer Engineering
Supervisor: Prof. Dr. Müslim BOZYİĞİT

January 2016, 55 pages

Publish/subscribe communication systems, where a large number of nodes (publishers and subscribers) are interested in variety of topics, have received more attention in recent years. Constructing an efficient overlay network that connects the nodes interested in several topics is an important issue in these systems. In constructing effective and scalable overlay networks, pub/sub network designers prefer to keep the diameter and maximum degree of the network low. This is in addition to lower running time cost and higher scalability. However, the existing algorithms seem to fail in decreasing maximum node degree and reducing diameter.

To address these issues for a given set of nodes with variety of topics, we present a heuristic algorithm known as Constant Diameter Minimum Maximum Degree (CD-MAX) which decreases the maximum node degree and maintains the diameter of the overlay at most at two. The new algorithm improves the maximum node degree, by 64 percent.

The CD-MAX algorithm selects the node with the lowest node degree and connects it to its neighbors. For more than two nodes with an equal highest node degree, this algorithm selects the

node with the highest density. The CD-MAX algorithm has a refinement version that decreases the maximum node degree even further.
The algorithm is validated and analyzed through simulations.

Keywords: Peer-To-Peer Network, Publish/Subscribe Systems, Overlay Network Diameter, Node Degree, Decentralized / Centralized Topology, Minimum Maximum Node Degree.

ÖZ

Minimum Maksimum Düğüm Dereceli Düşük Çaplı Konu Tabanlı Yayınla / Abone Ol Bindirmeli Ağ Kurulumu

Sina LAYAZALI
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Müslim BOZYİĞİT
Ocak 2016, 55 sayfa

Birçok abonelik içeren, Yayınla / Abone Ol iletişim sistemleri giderek daha yaygınlaşmaktadır. Konuların her birine ayrı ayrı abone olmuş düğüm iletişimlerini birbirine bağlayan etkin bir bindirmeli ağ tasarımı yapmak bu sistemlerin temel problemidir. Bir bindirmeli ağ tasarlamak için ideal olan, yalnızca maksimum düğüm derecesinin düşük tutulması değil, aynı zamanda ağın düşük çaplı ve ölçeklenebilir olması da önemlidir. Mevcut algoritmalar maksimum düğüm derecesi ve çapını düşürmekte başarısız bulunmaktadır.

Bu çalışmada, maksimum düğüm derecesini düşüren ve ağ çapını en fazla ikide sabitleyen, Sabit Çap Minimum Maksimum Dereceli (CD-MAX) adı verilen yeni bir algoritma sunulmaktadır. CD-MAX algoritması maksimum düğüm derecesini yaklaşık %64 iyileştiren bir algoritmadır. Ölçeklenebilir bir ağ kurabilmek için, CD-MAX algoritması en düşük düğüm dercesine sahip düğümü seçer ve bu düğümü bitişik düğülmelere bağlar. Eşit düğüm derecesine sahip birden fazla düğüm varsa, algoritma en yüksek yoğunluğa sahip düğümü seçer. CD-MAX algoritmasının düğüm derecesini daha da aşağı çeken bir versiyonu da bu çalışmaya dahil edilmiştir.

Algoritmanın doğrulanması ve analizi simülasyon ortamında gerçekleştirilmiştir.

Anahtar Kelimeler: Eş Düzeyli ağ, Yayınla / Abone Ol Sistemleri, Bindirmeli Ağ Çapı, Düğüm Derecesi, Dağıtık / Merkezi Topoloji, Minimum Maksimum Düğüm Derecesi.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to Ass. Prof. Melih Onuş for having led me to the topic and Prof. Dr. Müslim BOZYİĞİT for the thesis supervision.

It is a pleasure to express my special thanks to my family for their valuable support, especially my mother who has accompanied me throughout my stay in Ankara.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

LIST OF ABBREVIATIONS

P2P Networks	Peer-To-Peer Networks
Pub/Sub Systems	Publish/Subscribe Systems
GM	Greedy Merge Algorithm
CD-ODA	Constant Diameter Overlay Design Algorithm
CD-MAX	Constant Diameter Minimum Maximum Algorithm
DHT	Distributed Hash Table
TCO	Topic Connected Overlay

CHAPTER I

INTRODUCTION

1.1 Background

In pub/sub systems, publishers forward different types of messages to specific subscribers in a decoupled mode. Publishers broadcast the information through logical channels and subscribers receive them based on their topic interests. Pub/sub systems are divided into two different types namely *topic-based* and *content-based*. In the first group, publishers broadcast their messages to the "topics", where each topic is exclusively pertained to a specific logical channel. Then subscribers receive all messages associated with the topics to which they subscribed [1]. Publishers also take the responsibility of classifying the messages which subscribers can receive.

In a *content-based* system, subscribers only receive those messages whose attributes match with the interest of subscribers. These attributes characterize the logical channels [2]. Because of scalability and wide applicability, many applications such as stock-markets, Twitter, GUIs, cloud computing, online multiplayer games and RSS brokering services have been introduced during the last decade [3] [4] [5].

In this work, a fully decentralized topic-based pub/sub system based on the P2P connected overlay for each topic m is studied. For each topic $m \in M$, the subgraph derived from the nodes interested in m is connected. Hence, the nodes interested in topic m do not need to rely on other nodes in order to send or receive their messages. Such a network is referred to as topic-connected. Since nodes with a high number of connections need to maintain all the connections (e.g., checking the accessibility of neighbors) and data streaming through the connections, the overlay networks with low maximum node degree and low diameter are desirable in empirical studies. If there is a proper correlation between node subscriptions, then by adding only one edge between two nodes, the connectivity of many topics subscribed by those two nodes will be satisfied.

Hence, the maximum node degree and number of connections provided by the overlay will be considerably lower. The importance of the contribution of nodes has been highlighted in recent papers such as “Constructing Scalable Overlay for pub-sub with Many Topics” shown by “Gregory Chockler” [6].

In this thesis, we focus on constructing an efficient TCO with the least possible maximum degree and a maximum diameter of 2. Reducing the maximum degree of overlay can play a vital role in a variety of network fields such as survivable and wireless network design [7] [8] [9] [10]. Chockler et al. presented the concept of topic-connectivity in which an individual overlay network connects nodes with similar topics [6]. They introduced the Greedy Merge algorithm to construct an overlay with the least possible number of connections. We use their idea as a reference and a principle to construct an effective and scalable overlay network. Furthermore, a number of other solutions for overlay design have been introduced recently [2]. However, all of these existing methods suffer from high diameter or maximum node degree [11] [12] [13].

1.2 Objectives

The aim of this dissertation is to develop an algorithm to construct a scalable TCO which has a low maximum node degree and a constant diameter 2 simultaneously. In fact, our algorithm (CD-MAX) has improved the results of the algorithms in the literature including CD-ODA I and CD-ODA II. These algorithms are implemented on the network and the results were compared with the results of CD-MAX.

1.3 Organization of the Thesis

This dissertation contains six chapters. In Chapter I, we introduce the problems related to pub/sub systems design and the main goal of this dissertation is explained. In Chapter II, we conduct a survey on pub/sub networks. In Chapter III, the concept of the Greedy Merge algorithm has been interpreted which builds an overlay with minimum number of edges. The algorithm plays a principle role for constructing an efficient pub/sub overlay network.

In chapter IV, the algorithms which have been used for Topic-Based pub/sub Overlay Network design are demonstrated.

Chapter V includes the developments and discussion of CD-MAX. Chapter VI includes the comparison of experiment results of the CD-MAX algorithm with existing methods, such as GM, CD-ODA, CD-ODA I and CD-ODA II. Finally in Chapter VII, the conclusion is presented.

Chapter II

Publish/Subscribe systems

2.1 Introduction

The pub/sub systems attract large number of users due to loosely coupled communication [14] [15] [3] [12] [16] [17]. Generally, subscribers irrespective of publishers express their passion for an event occurring in the environment. Then they receive the events which are coincident with their interests asynchronously. Pub/sub systems are well-known communication paradigm to establish wide range of distributed applications [18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28]. Today, the Internet has significantly reshaped the distributed systems which comprise large numbers of entities whose behavior should change under different conditions. These constraints increased the demand for loosely coupling approaches which improved the flexibility and scalability of applications [29]. In pub/sub systems, publishers who are conscious of the attendance of subscribers, broadcast information pertaining to a variety of events throughout the system by assigning a specific value from a set of well-defined attributes. After that, the subscribers who announced their topics of interest by proper subscription, wait until they are informed about a matched phenomenon. The pub/sub structure delivers the events to subscribers whose subscriptions are matched to those events [30]. Creating a centralized pub/sub system provides a proper view of the system which leads to matching algorithm implementation to becoming much easier. However, there is one drawback. It lacks scalability when the number of publishers and subscribers grows. Therefore, a decentralized pub/sub topology is more suitable in order to provide an effective techniques for matching. The P2P model is a suitable architecture to establish large scale distributed systems [31].

Generally, pub/sub systems are classified in two different groups namely *topic-based* and *content-based*. The first group appears as a news room where users participate in an ideal group of topics by declaring their interests. Consequently, all messages about those topics will be sent to every user who has joined that particular group [32].

On the other hand, a *content-based* method allocates an ability to their users to express their own interests by assigning predicates over a specific value from a set of well-defined attributes. In fact, the matching algorithm between publishers and subscriptions is implemented based on the value of attributes referred to as content.

In order to achieve scalability and fault tolerance, pub/sub systems are implemented in a distributed manner. Distributed solutions mostly attempt topic-based systems [33] [34]. However, in content-based systems, some intend to multicast messages according to routing trees methods and a number of others use the concept of rendezvous guaranteeing all the events and subscriptions take part in the system [6] [29] [35] [36] [37] [38]. Furthermore, a number of application designers endeavor to combine both content-based and topic-based pub/sub systems [27]. For instance, in some implementation, a topic-based system is accomplished in a decentralized manner by utilizing a DHT, such as by Peter “Triantafillou and Ioannis Aekaterinidis” [39].

2.2 Publish/Subscribe Challenges

In order to design a pub/sub system, there are three main challenges; these include:

- How subscribers can express their interests in the events observed by the system.
- How notification service, which can be an individual, centralized server or a collection of distributed processes, are performing; and
- How publishers forward messages to subscribers. In other words, in which manner notification services manage the underlying network levels to send content to subscribers properly.

These states are coupled strongly and their contribution can affect system performance. For instance, a rudimentary subscription algorithm may improve the functionality of multicasting; however, it facilitates poor expression ability for subscribers to announce their interested topics [30].

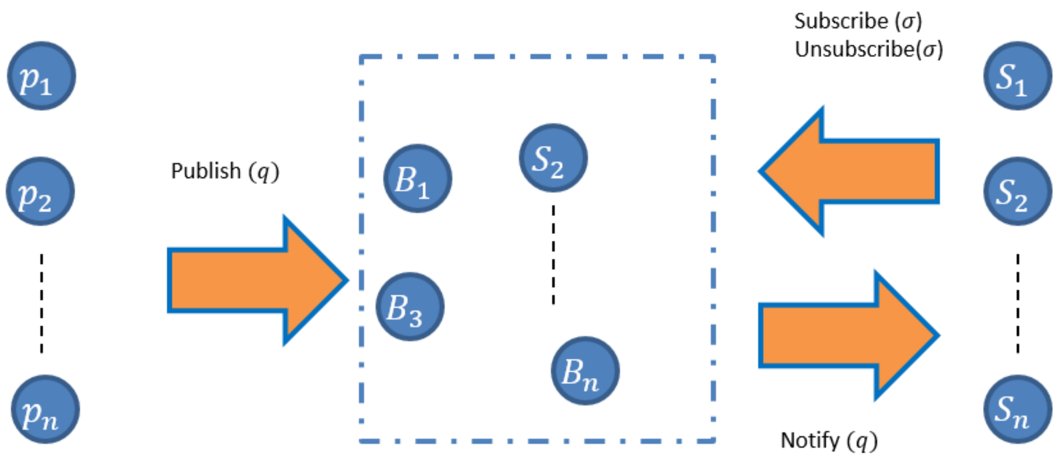


Figure 2.1: General Pub/Sub System Model

2.3 Principle Publish/Subscribe Characteristics

In this section, a general view of the pub/sub system with its functions and components participating in the system is presented. Figure 2.1, shows a generic model of a pub/sub system.

2.3.1 Publish/Subscribe Components

Generally, a pub/sub system can be modelled as a trimerous <∏, β, ∑ > collections of functions. The sets involved are determined based on their functionality: ∏ = $p_0, \ldots, p_{i-1}$ is a set of i processes in the system that act as publishers providing information to those need it.
∑ = $C_0, \ldots, C_{j-1}$ is set of j processes referred to as subscribers that are known as consumers of information provided by publishers. The set of publishers and the set of subscribers can have non-zero intersection, which means that the process can also operate as both publisher and subscriber.
β = $B_0, \ldots, B_{k-1}$ is set of k processes known as brokers.
It is assumed that publishers and subscribers are not coupled, meaning that a process in ∏ is not able to make a connection directly with a process that belongs to ∑ nor contrariwise (unless it acts as both publisher and subscriber).

Decoupling is an ideal feature of communication systems. Systems with decoupling mechanism do not need to consider any issues such as addressing and synchronization.

A set of brokers named β, presents a logical centralized service that enables publishers and subscribers to connect. In other words, any publisher or subscriber can exclusively connect to the brokers in order to send or receive specific information. Not only do brokers provide communication between publishers and subscribers, they also keep publishers and subscribers in a decoupled mode during communication process.

A broker is also referred to as a notification service and every publisher and subscriber participating in the network acts as a client for the broker. [40].

2.3.2 Publisher and Subscriber Communication Mechanism

Each publisher or subscriber can interact with a notification service via a set of provided functions. A publisher broadcasts some specific data pertaining to an event q to other components by implementing the publish (q) function over the broker. After that, the broker submits the information to specific subscribers by executing notification (q) over them.

In pub/sub systems, subscriber can install or remove a subscription on the broker by executing the subscribe (σ) and unsubscribe (σ) functions respectively [30].

2.3.3 Notification and Subscription

In a pub/sub system, publishers create an event, while a broker dispatches the notification to consumers (subscribers). In pub/ sub systems, notifications are referred to as a collection of attribute-value pairs. Each attribute comprises a title, a plain character string, and a type, which is considered to be principle data types presented in programming languages or query languages (e.g. integer, real, string, etc.). Subscribers express their interests via subscription. A subscription is defined as a set $\partial = (p, r)$ where, $r \in \sum$ is the sets of subscribers waiting for an event and p is responsible for announcing events occurring through the system. It can be said that a notification n matches a subscription if and only if it convinces p associated with that subscription.

The algorithm which submits that notification n coinciding with p is referred to as matching, where $(n \subset p)$ [30].

2.4 Network Communication Topology

Before we start considering the pub/sub architecture, it would be very helpful to consider some basic information about two important network communications protocols namely client-server and P2P, as explained as follow:

2.4.1 Client-Server Network Architecture

Today client-server networks are ubiquitous in array of applications. Websites are example of client-server networks where users send request to servers in order to acquire specific information.

Moreover, desktop applications which comprise local and wide area network connectivity are connected to a server. Figure 2.1 shows a simple client-server architecture.

There are many applications in which client-server architecture is implemented; however, most suffer from scalability issues meaning an increasing number of clients over the network, would increment communication load on the server, thereby leading it to collapse or inability to respond.

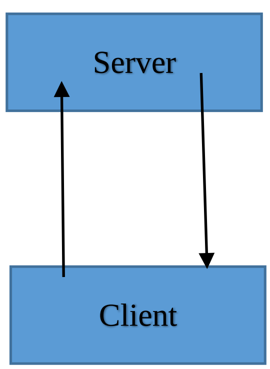


Figure 2.2: Client-Server Architecture

Figure 2.3 shows the clients-server architecture with many clients. Recently, to solve this drawback, a number of solutions have been presented. Examples include the “scale up” which dedicates more power and resources to the server and “scale out” which adds extra severs [41]. However, there are obstacles to set up solutions on the networks. For example, scaling up is costly due to requirements for more advanced hardware. Although the second solution is potentially more flexible, additional infrastructure layers are required.

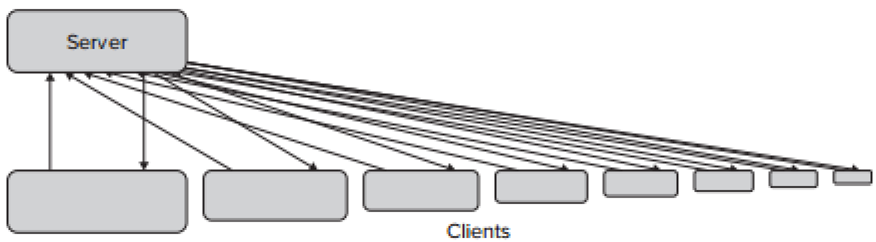


Figure 2.3: Client-Server Architecture with Large Number of Clients

2.4.2 Peer-To-Peer (P2P) Network Architecture

P2P has been one of the most effective technologies presented in the field of networking in recent years. Although file sharing applications use P2P, it does not mean that P2P is merely used in only file sharing applications. P2P can be seen as an ad-hoc network or two connected PCs in a room transferring specific files [42].

Recently P2P file sharing applications such as Napster, Scribe, Bayeux and Siena have astonished many users [41]. An employer’s idea is that “P2P provides users to share their content and beneficial services in a decentralized manner” [43].

This definition highlights two principle characteristics of P2P networks.

- Scalability: compatibility of the system in terms of changes in the size. For example, Stability and simplicity of the network should be guaranteed even though network components are growing.
- Fault Tolerance / Reliability: failure of each component cannot cause to malfunction of an entire system.

Files sharing networks, such as Gnutella are desirable example of reliability and scalability of P2P networks. P2P networks are classified into two distinct classes: Pure P2P and Hybrid P2P. In the first group, there is no central server. Gnutella and Freenet are well known examples of this type of network. Unlike the Pure P2P networks, in Hybrid networks, such as Magi, Groove and Napster a central server is provided to obtain specific information like peer identity. In this type of network, every peer before connecting to others should primarily contact a central server [41].

2.4.2.1 Peer-To-Peer Architecture Topology

The P2P approach differs from its rival, the client-server attitude. Instead of concentrating on making communication between the server and its own clients effective, P2P tends to consider the manners in which clients communicate. For example, a website called www.book.com declares that a new version of book published on the website can be downloaded for free. However, the download link will be removed following day. Therefore, users will engage with refreshing the page and they will be eager to download the book once it appears. If we imagine that when a book becomes available, massive requests through clients cause server collapses under the strain.

Now it is time for P2P to take center stage in order to prevent a web browser collapse. Unlike the client-server mechanism, which tends to send files from a server to receivers (all clients) directly, P2P technology sends files to a small number of clients and a number of the remaining clients can download files from the clients that have already downloaded the file. Subsequently any other remaining clients can download the file from the second level and so on. This process will be even faster when a file is split into chunks and divided among clients who download it directly from a server or other clients who downloaded earlier.

2.4.2.2 Peer-To-Peer Challenges

In a P2P network app, each client should be capable of discovering, connecting and even communicating with other clients participating in a network application. To overcome such issues, the server can keep a list of clients which is accessible to clients that keep them in touch with themselves. Alternatively, they can use an infrastructure that enables the client to discover other clients. File sharing systems are mostly equipped with particular lists on the server in which services are defined as trackers.

In a file sharing topology, a client can act as a server by proclaiming that it has a file available to share and so registers it with a tracker. It is obviously clear that a real (pure) P2P network comprises only clients. (that is, no servers at all)

In P2P networks, dividing peers into different groups, where peers can communicate with their group members, boosts network performance and impedes connection topology to become extremely complex. Moreover, classifying peers in different local based groups can improve network performance, since peers can communicate with each other with a small number of hops among network computers.

Although communication protocols, such as TCP/IP, have been advanced significantly, improvement in high-level (e.g. services) and low level technologies (e.g. multicast protocols in which multiple endpoints received data concurrently) are still continuing.

Figure 2.4 shows a typical P2P architecture. P2P systems are convenient for pub/sub architecture design.

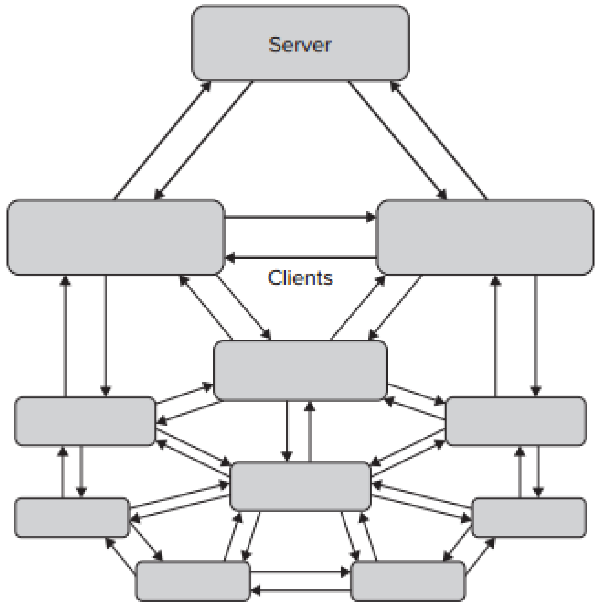


Figure 2.4: P2P Architecture

A group of peers can be deem well-connected if and only if at least one of the following conditions can be implemented on them:

- Any pair of peers can connect. In other words, any peer communicates with any ideal requested peer.
- There are a small number of connections intersecting any pair of peers.
- Removing any peer cannot impede other peers from communicating with each other.

It has to be noted that these statements do not mean that each peer must connect to other peers directly. In fact, a group of peers which tends to be well-connected merely needs to connect with a very number of peers.

2.5 Publish/Subscribe System Architecture

The architecture of pub/sub generally can be divided into P2P and client-server groups [29]. Clients also perform as either subscribers, or publishers or both subscribers and publishers. From now on, publishes and subscribers are referred to as event clients.

2.5.1 Client-Server Model

In this model, every components performs as an event client or broker. When an event occurs in the system, the broker broadcasts it to its clients. Moreover, each server communicates with other servers to achieve a number of benefits including scalability.

The client-Server model can be designed in form of a Centralized Topology, Hierarchical Topology, Ring Topology and Irregular Polygon topologies.

2.5.1.1 Star Topology

In this topology, an individual central server intermediates between subscribers and publishers. As shown in Figure 2.5, four publishers broadcast events through the system to be delivered to subscribers interested in specific events. In this topology, it is feasible that a subscriber can receive events from any publishers randomly via a single server intermediating between providers and subscribers.

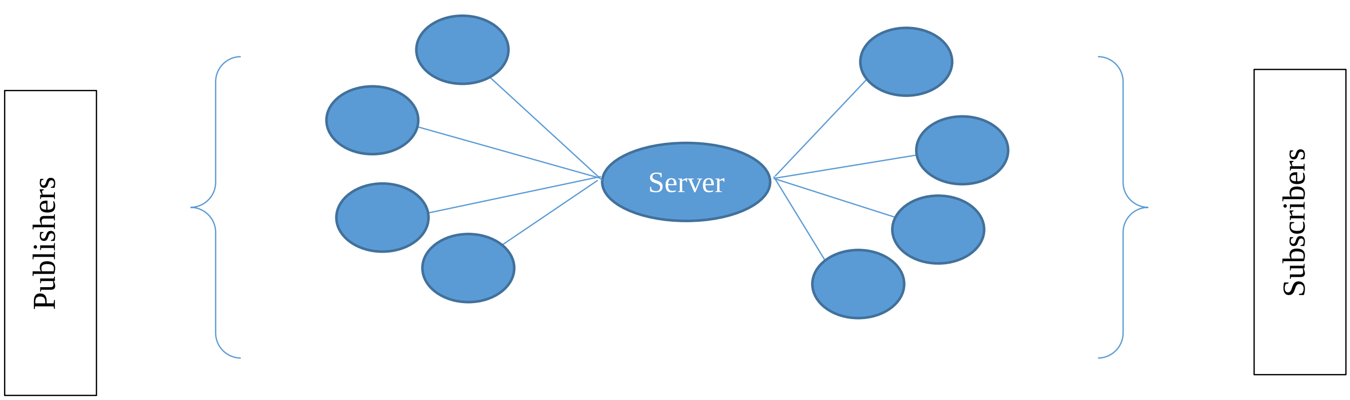


Figure 2.5: Star Topology

2.5.1.2 Hierarchical Server Topology

Due to the existence of hierarchical communication among event servers, this topology is referred to as hierarchical topology which is known as masterpiece of pub/sub system topology. As demonstrated in Figure 2.6, each server maintains a number of publishers and subscribers. In this topology, every event server is connected to a root server and server-to-server communication and client-to-server obey same protocol.

The main purpose of this topology is to achieve scalability. A root server receives all broadcasted events and subscriptions from all of its clients and broadcasts them to its subtree. Moreover, the root server operates as a gatekeeper maintaining the traffic of the subtree. Note that, the server-to- server and server-to-client connection protocols use similar protocol.

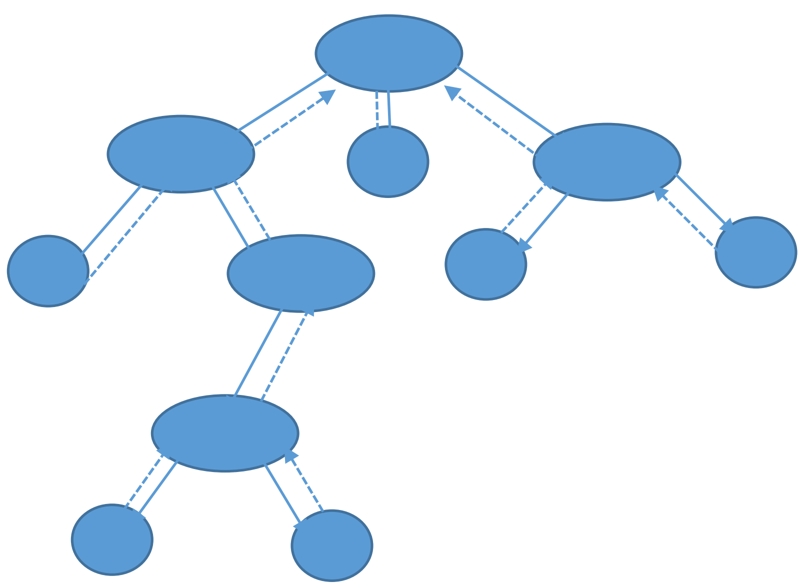


Figure 2.6: Hierarchical Server Topology

2.5.1.3 Ring Topology

As illustrated in Figure 2.7, servers communicate with other servers in a manner consistent with P2P, and the connection of servers is provided in the form of a ring. A server can communicate with other servers by bidirectional communication protocol to transmit subscriptions.

The server-to-server communication type differs from the communication protocol used between clients and servers in terms of the amount and type of information that is being transmitted.

In server-to-server communication, two end nodes manage the information for each other. In client-server communication, a client can create a subscription. Or it becomes a recipient for broadcasted messages. On the other hand, servers merely act as access points or as routers to send messages.

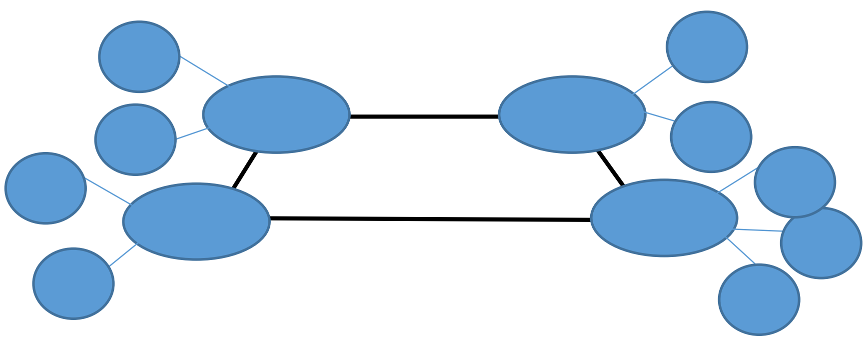


Figure 2.7: Ring Topology

2.5.1.4 Irregular Polygon Topology

In fact, Irregular Polygon Topology is a generic version of Ring Topology in which all servers communicate in a ring form. Similarly to Ring Topology, servers in Irregular polygon topology communicate bidirectionally (Figure 2.8).

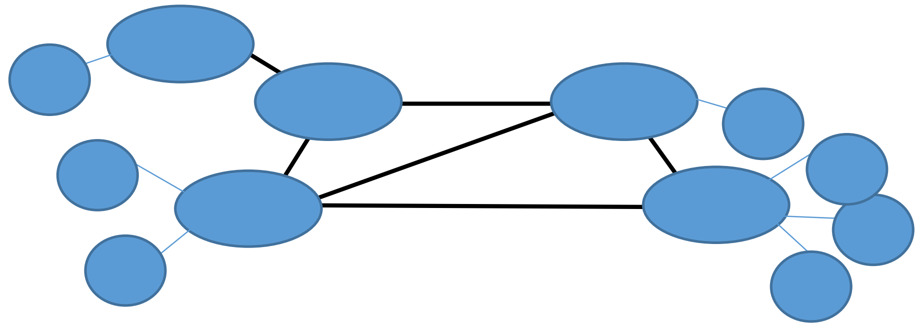


Figure 2.8: Irregular Polygon Topology

2.5.2 P2P Model

In the P2P paradigm shown in Figure 2.9, all nodes can operate in a variety of roles such as a publisher, subscriber, root or internal node of a multicast tree and each feasible combination related to them.

In this topology, neither node servers nor node clients are determined and some server capability, such as persistence, transaction or security issues, is provided as a local part of the server nodes. The overlay network which is going to be design by CD-MAX is derived from the P2P model and from the Star Topology. In our topology every node is supposed to be both publisher and subscriber both.

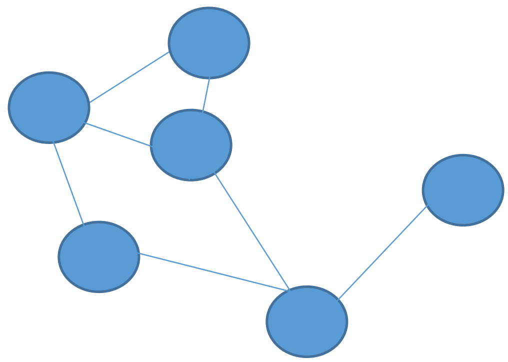


Figure 2.9 General Model of a Pub/Sub System based on the P2P model

CHAPTER III

Greedy Merge Algorithm

3.1 Introduction

In this chapter, we focus on a basic pub/sub system design in which no intermediation (broker) is implemented. In order to design an efficient pub/sub system, an effective publication routing protocol can play a vital role on a system's performance. Therefore, quality of a constructed overlay can be assessed based on the complexity of the routing scheme applied [21].
If all nodes subscribe the same topic $m \in M,$ the complexity would be intuitively minimized. In this case, the topic can be arranged into a dissemination tree that includes the following features [6] [16]:

1. The tree comprises merely the nodes which are interested in topic m
2. The diameter of the tree would be lower

In this chapter, we only consider the first issue, the second issue will be dealt in the following chapters.

Suppose that G is an overlay network, then the essential prerequisite to include issue (1) is a topic-connectivity where a sub-graph connects all nodes interested in topic m. Note that, by a separated overlay terminology such as ring, tree or star, it would be much easier to create a topic-connectivity when a new topic joins the system [6].

To investigate this issue a *minimum topic-connected overlay problem* is introduced. The main goal of this problem is to design an overlay network with a minimum number of links. In this chapter, we present the GM algorithm that solves the problem of efficiency. The algorithm constructs an overlay including node set N, topic set M and a topic assignment function $(Interest)$ with a minimum number of edges.

Note that this algorithm acts as a reference for our algorithm which builds the overlay network with the lowest maximum node degree. For all overlays considered in this thesis, each node can act as both publisher and subscriber. The $Interest$ assignment function for each node $n \in N$ subscribing topic $m \in M$ is based on a Boolean type over an $N \times M$ matrix. Thus, the presentation of an interest function for node n interested in topic m, is $Interest\ (n, m) = true$. An overlay network over the collection of nodes is a graph defined as follows:

$$(N, E) : E \subseteq N \times N$$

We present a topic connected components for the topic as follows:

$$n = \{n \in N : Interest(n, m)\}$$

3.2 The Minimum Topic Connected Overlay (MIN-TCO) Problem

The principle aim of the MIN-TCO problem is to establish a topic connected overlay network with the least possible number of connections [6]. For a given set of nodes N, topics M and a topic assignment function $Interest$ over $N \times M$, a definition of MIN-TCO $(N, M, Interest)$ is presented as follows:

Definition: (MIN-TCO $(N, M, Interest)$) establishes a topic connected overlay $G = (N, E)$ such that:

$$|E| = \min E' \subseteq 2^E\{|E'|: (N, E')\}$$

In addition, $TCO(N, M, Interest)$ has the responsibility of registering nodes into the overlay by utilizing the given number of links $Q > 0$. In other words, the possibility of joining node $n \in N$ into a topic connected overlay can be determined by $TCO(N, M, Interest)$.

Definition: ($TCO(N, M, Interest, Q)$). Given $inp = [N, M, Interest, Q]$, decide whether $inp \in L_{TCO}$.

Officially, $L_{TCO} = \{[N, M, Interest, Q]$ is an overlay network $G = (N, M)$ which provides a topic connected for each $m \in M, Interest$, and $|E|$ =Q}.

3.3 Greedy Merge Algorithm Implementation

This algorithm begins with the overlay network $G = (N, \emptyset)$. For each topic $m \in M$ there are $\sum_{m \in M} | \{n: Interest(n, m) = true\}|$ individual topic connected components of G.
The algorithm continues by connecting two nodes at each repetition, untill the resulting overlay comprises maximally one topic connected component for each $m \in M$. The two pair of nodes connected during at each repetition, are those which have the greatest number of topics in common.

For each new pair of nodes such as (n, w) that is connected to the overlay network, $M_{(n,w)} \subseteq M$ is considered as a collection of topics $m \in Interest(n, m) \cap Interest(w, m)$. Moreover, these nodes are associated with two distinct number of topic connected components of those topics. Therefore, when these nodes are connected, two topic connected components for each $m \in M_{(n,w)}$are incorporated as an individual connected component $C_m = Nodes[n][m] \cup nodes[w][m]$. Hence, the coordination of (n, w) reduces the quantity of topic connected components for each topic $m \in M$ is $|T_{n,w}|$, where $|T_{n,w}|$ is the number of topics subscribed to by nodes n and w.
At each repetition, the algorithm discovers the edge (n, w) which its $|T_{(n,w)}|$ is the maximum among all node.

Clearly, each added edge leads the overlay to merge two topic connected components. The total number of topic connected components of the overlay is reduced at most. In addition, by adding an edge, the total number of topic $m \in M$ subscribed by two nodes n and w is reduced by one.

When an edge with $|T_{(n,w)}| > 0$ cannot be discovered, the algorithm pauses. Because this condition implies that for each $m \in M$, the subgraph $G_m \subseteq G$ evolving with subscribers associated with that topic, is connected.

In order to discover the ideal edge (n, w), Algorithms 3.1 and 3.2 utilize an ancillary array LinkContrib. In fact, LinkContrib $[i]$ is the subset of all possible edges over $N \times N$ with contribution i; i.e., $(n, w) \in linkContribution\ [i]$ if $i = |T_{(n,w)}|$. In other words, the LinkContrib for edge $e \in N \times N$ connecting nodes n and w, is the number of topics which are common between these two nodes. Algorithm 3.3 shows the implementation of the discovery of the best edge $e = (n, w)$ to be added to the overlay.

Note that, while edge e is added to the overlay and omitted from LinkContrib, the contribution of the other edges should be updated.

Data Structure
✓ Output Overlay Edges: a set of overlay edges, initially 0;
✓ Nodes: a 2-dimensional array over $N \times M$ whose elements are subsets of N such that for each $n \in N, m \in M$: (1)$interest(n, m) = true$, and (2) for each $w \in Nodes[n][m]$: $Interest(w, m)$ and both w and v belong to the same topic-connected component for m.
✓ LINKCONTRIB: an array of size $\|T\|$ with elements being sets of edges chosen from $N \times N$. If edge $e \in LINKCONTRIB[i]$, then $e \notin Out\ Put\ Overlay\ Edge$, and adding E to the overlay at the current iteration will reduce the number of topic-connect components by $i (where\ 1 \leq i \leq \|M\|)$.
✓ $HIGHESTCONTRIB$: holds the biggest i for which $LINKCONTRIB[i] \neq 0$.

Algorithm 3.1: Data Structure of Greedy Merge Algorithm ("Chockler") [6]

Data Structure Initialization

For all $nodes\ n$ **do**
For all $topic\ m\ such\ that\ Interest\ (n, m)$ **do**
$Nodes[n][m] \leftarrow \{n\}$
For all $edge\ e = (n, w)$ **do**
Contrib $\leftarrow |\{m \in M: Interest(n, m) \wedge Interest(w, m)\}|$
If $Contrib > 0$ **then**
Add e **to** $LinkContrib[contrib]$
$HighestContrib \leftarrow \max(i|LinkContrib[i])\ not\ empty|)$

Algorithm 3.2: Data Structure Initialization of Greedy Merge Algorithm [6]

Overlay Construction

$OutPutOverLayerEdges \leftarrow \emptyset$
While $HighestContrib > 0$ **do**
$e \leftarrow some\ edge(n, w)\ from\ LinkContrib[HighestContrib]$
$OutPutOverLayerEdges \leftarrow OutPutLayerEdges\ \cup \{e\}$
Delete e from $LinkContrib[HighestContrib]$
For all topic m such that $Interest(n, m) \wedge Interest(w, m)$ **do**
For all $n' \in Nodes[n][m], w' \in Nodes[w][m], (n', w') \neq (n, w)$ **do**
Locate i such that $(n', w') \in LinkContrib[i]$
Delete i such that $(n', w') \in LinkContrib[i]$
If $i > 1$ **then**
Add $(n', w;)$ to $LinkContrib[i - 1]$
$new - connected - component - list \leftarrow Nodes[n][m] \cup Nodes[w][m]$
For all $u \in new - connected - component - list$ **do**
$Nodes[u][m] \leftarrow new - connected - component - list$
While $HighestContrib > 0$ and $LinkContrib[HighestContrib]\ is\ empty$ **do**
$HighestContrib \leftarrow HighestContrib\ - 1$
$OutPut(N, OutPutOverLayEdges)$ and halt

Algorithm 3.3: Overlay Construction of the Greedy Merge Algorithm [6]

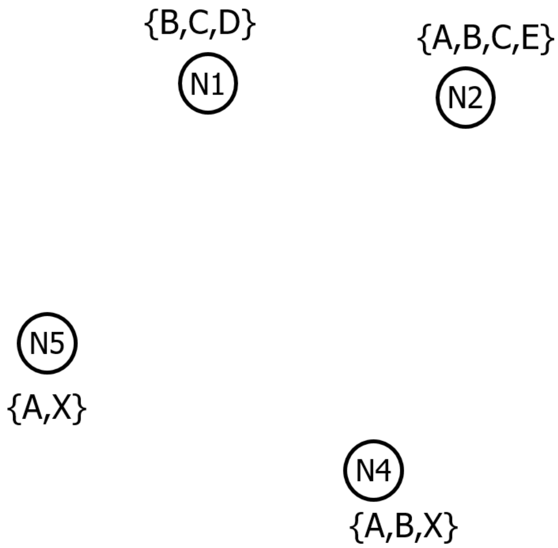
3.4 Example for Greedy Merge Algorithm

As it is obvious in Figure 3.1, there are 5 nodes which subscribe to different topics. In addition, there is a table of topic connected components which shows the total number of topics subscribed by each node. For this example, the GM algorithm finds two nodes which have the maximum number of common topics and connects them with an edge. Put differently, any edges added to the network is the one which reduces the total number of topic connected components maximally.

When these two nodes are connected, the number of each common topic will decrease by 1 and in the following, the topic connect component table will be updated. Figure 3.2 demonstrates the implementation of the GM algorithm over the overlay network.

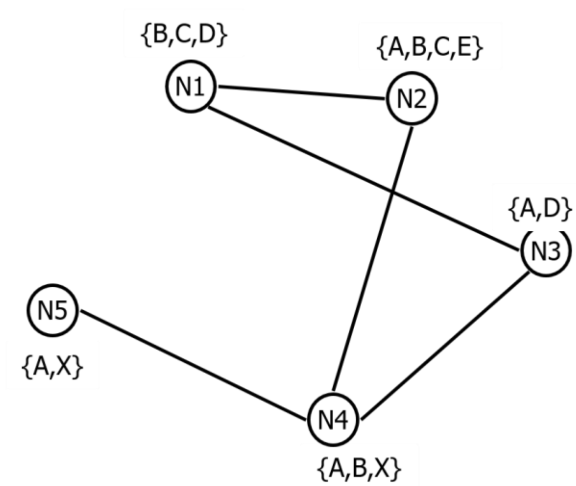
Nodes	Topics
$\boldsymbol{N_1}$	$\{B, C, D\}$
$\boldsymbol{N_2}$	$\{A, B, C, E\}$
$\boldsymbol{N_3}$	{A,D}
$\boldsymbol{N_4}$	{A,B,X}
$\boldsymbol{N_5}$	{A,X}

Table 3.1: Topic Assignment for Example 3.1



Topic	# of conn. Comps
A	4
B	3
C	2
D	2
X	2
E	1

Figure 3.1: Example of GM Algorithm Implementation (Part 1)



Topic	# of conn. Comps
A	1
B	1
C	1
D	1
X	1
E	1

Figure 3.2: Example of GM Algorithm Implementation (Part 2)

CHAPTER IV

Low Diameter Publish/Subscribe Overlay Algorithms

4.1. Introduction

In this chapter, the existing low-diameter algorithms for constructing a topic-based pub/sub overlay network including the number of nodes $n \in N$ interested in topic m, will be presented. The overlays are constructed in a *decentralized* manner by these algorithms. Hence, the nodes which are subscribed to similar topics do not need to depend on other nodes to forward their messages through the network. As mentioned in Chapter III, most of the approaches simultaneously failed to decrease the maximum degree and the diameter of the overlay. For example, “Chockler” posed the GM algorithm to solve the problem. Although, the algorithm prepares a low maximum node degree and requires the lowest number of links to construct an overlay. However, this algorithm suffers from a high rate of diameter and a lack of scalability [6]. This algorithm is presented in Chapter III. The CD-ODA, CD-ODA I and CD-ODA II algorithms establish a topic-based pub/sub network in which each node interested in a variety of topics is assured to have low a diameter to forward its messages. However, the maximum degree resulting from these algorithms is considerably high.

The diameter and maximum degree provided by a network can play a vital role in efficient routing [7] [9] [44]. Hence, designing a network with a minimum diameter and low maximum degree can noticeably improve the simplicity and performance of the network. In the following chapters, the efficiency of our presented algorithms is validated by experimental results.

4.2 Constant Diameter Overlay Design Algorithm (CD-ODA)

This algorithm starts with an overlay network defined in Chapter III as $G(N, \emptyset)$.

At each repetition, a node with the highest number of neighbors is selected. the number of neighbors for node n can be computed as follows:

$$n_u = |\{n \in N | \exists \in M, Interest(n,m) = Interest(u,m) = 1\}|$$

Constant Diameter Overlay Design Algorithm (CD-ODA)
1. $M \leftarrow Set\ of\ all\ topics$
2. **While** M is not empty **do**
3. For each node u, calculate number of nodes n such that there exists a topic m in M and $Interest(u,m) = Interest(n,m) = 1$ Denote this number by n_u
4. Find node u with maximum n_u
5. Put an edge between u and all nodes v such that there exists a topic $m \in M$ with $Interest(u,m) = Interest(n,m) = 1$
6. Remove all topics m from M such that $Interest(u,m) = 1$
7. **End while**

Algorithm 4.1: CD-ODA Algorithm ("Onus") [2]

When node u with all its neighbors is connected, all topics associated with the interests of node u will be emitted from the topic sets. The CD-ODA algorithm is shown by Algorithm 4.1.

4.3 Constant Diameter Overlay Design Algorithm I (CD-ODA I)

In order to design a network, Algorithm (CD-ODA) considers the number of neighbors, but Algorithm (CD-ODA I) presented below, checks the number of weighted neighbors. Similarity to the previous algorithm, Algorithm 4.2 starts with an overlay network$G(N, \emptyset)$. At each repetition, node u which has the maximum number of weighted neighbors is selected.
Node u and its neighbors are connected, and then the topics in which node u is interested, will be removed from the topic list.

Constant Diameter Overlay Design Algorithm I (CD-ODA I)
1. $M \leftarrow Set\ of\ all\ topics$
2. **While** M is not empty **do**
3. For each node u, calculate total number of weighted neighbors. Denote this number by w_u
4. Find node u with maximum w_u
5. Put an edge between u and all nodes v such that there exists a topic $m \in M$ with $Interest(u, m) = Interest(n, m) = 1$
6. Remove all topics m from M such that $Interest(u, m) = 1$
7. **End while**

Algorithm 4.2: CD-ODA I Algorithm ("Onus") [2]

4.4 Constant Diameter Overlay Design Algorithm II (CD-ODA II)

Similarity to the other presented algorithms in this section, this algorithm also starts up with an overlay network $G(N, \emptyset)$.

Constant Diameter Overlay Design Algorithm II (CD-ODA II)
1. $M \leftarrow Set\ of\ all\ topics$
2. **While** M is not empty **do**
3. For each node u, calculate its connection density. Denote this number by d_u
4. Find node u with maximum w_u
5. Put an edge between u and all nodes v such that there exists a topic $m \in M$ with $Interest(u, m) = Interest(n, m) = 1$
6. Remove all topics m from M such that $Interest(u, m) = 1$
7. **End while**

Algorithm 4.3: CD-ODA II Algorithm ("Onus") [2]

During each repetition, each node which has maximum connection density $d_u$is selected. Then, the edges between a selected node and its neighbors are added. Afterwards, all topics in the node's subscription are removed from set of topics. Since all these three algorithms construct a network based on the star topology, the diameter of the network will be at most 2. In other words, all nodes deployed over the network, require maximally 2 connections to forward or receive any subscribed messages. Compared with the GM algorithm, these algorithms have considerably lower diameter. However, the number of edges needed to connect the nodes is higher than for the GM algorithm.

In addition, these three algorithms suffer from high rate of maximum node degree.

CHAPTER V

Low Diameter Topic-Based Pub/Sub Overlay Network With Minimum Maximum Node Degree Algorithm

5.1 Introduction

Most of the approaches to designing a scalable overlay network have failed to achieve a good tradeoff between maximum degrees and diameter of the overlay. To decrease the number of connections, Chockler et al. presented the problem of constructing overlay with the least possible links. He considered this issue as an NP-Complete problem and posed the greedy merge algorithm to solve that problem [6]. Moreover, a number of other solutions presented in the previous chapter provide overlays with a low rate of diameter. Nevertheless, the maximum node degree of these approaches is considerably high [2] [21] [45].

In this chapter, a novel algorithm (CD-MAX) is presented in order to design an overlay with minimum maximum node degree. As CD-MAX builds the overlay in a star topology, the algorithm keeps the diameter of the overlay at most at 2. The algorithm improves the maximum degree by 64 percent. To construct a network, CD-MAX initially selects the nodes which have the lowest node degree. Each selected node takes the responsibility of topics to which they are subscribed. Moreover, the CD-MAX algorithm is attached with $refinement$ that improves the maximum degree.

After CD-MAX implementation, the $refinement$ checks all the nodes with a maximum node degree. Then, it discovers other nodes with a lower degree for topics subscribed to by the node with a maximum node degree. If the $refinement$ can find alternative nodes, the edges which are connected to the node with a maximum node degree will be removed. After that, any discovered nodes become the center of topics which are subscribed by the previous node. In this chapter, five different examples are illustrated that give an explicit view of the CD-MAX algorithm. We validate and analyze the performance of the algorithm in the following chapter.

5.2 Implementation

In this section, we present the CD-MAX algorithm in order to decrease the maximum node degree problem. This algorithm shown in Algorithm 5.1, starts with the overlay network $G(N, \emptyset)$ similarity to other presented algorithms. During each loop, CD-MAX selects a node u with maximum connection density d_u among those which increase the current maximum degree minimally. Afterward, node u is connected to its neighbors and the topics subscribed by node u are eliminated from the original topics set. In other words, upon each repetition, the CD-MAX algorithm selects the node u with the lowest node degree. If CD-MAX finds more than two nodes with an equal lowest node degree, the algorithm selects the node with the highest node density (d_u). Then, CD-MAX adds edges between node u and all nodes defined as its neighbors. Topics subscribed by the node u will be removed from the topic sets.

Note that the CD-MAX algorithm includes a refinement part. When CD-MAX is terminated, CD-MAX $refinement$ takes center stage. It checks all nodes and finds the nodes with the maximum degree, which are the center nodes for specific topics. After that, it searches the overlay to discover alternative nodes with a lower node degree for those topics to which they subscribed by the node with a maximum node degree. If $refinement$ manages to locate those nodes, the edges related to the node with a maximum node degree will be removed and new discovered nodes will be connected to their neighbors for those topics.

We call this trend CD-MAX $refinement$ for this part of the algorithm. This feature can decrease the maximum node degree more significantly than the normal CD-MAX algorithm. This algorithm is shown in Algorithm 5.2.

Constant Diameter Maximum (CD-MAX)
1. $M \leftarrow Set\ of\ all\ topics$
2. **While** M is not empty **do**
3. For each node u, calculate its connection density. Denote this number by d_u
4. Find node u with maximum d_u among the ones which increases the current maximum degree minimally
5. Put an edge between u and all nodes v such that there exists a topic $m \in M$ with $Interest(u, m) = Interest(n, m) = 1$
6. Remove all topics m from M such that $Interest(u, m) = 1$
7. **End while**

Algorithm 5.1: CD-MAX Algorithm

CD-MAX refinement
1. $M \leftarrow Set\ of\ all\ topics$
2. **While** maximum node degree improving
3. For each node $u \in N$, calculate the node degree. Denote this number by d_u
4. Find node d_u which has maximum node degree
5. Find node $q \in N$ which has lower node degree and $Interest(\mathrm{q}, m) = Interest(u, m) = 1.$
6. Remove all edges connected to node u
7. Put an edge between q and all nodes n such that $\mathrm{Interest}(\mathrm{q}, m) = Interest(n, m) = 1.$
8. End while

Algorithm 5.2: CD-MAX refinement Algorithm

5.3 CD-MAX Algorithm Complexity

In this section, it is proved that all algorithms with the lowest diameter are implemented in $\Theta(|N|^2 * |M|^2)$ time. These algorithms build an overlay with a constant diameter of 2 for each topic. During running time, we can calculate overall neighbors' weight or the density for each node.

In the CD-MAX algorithm, node q with a minimum degree is selected to be the center of the topics to which it subscribes. The topics to which node q subscribes will be removed from the topics list. This iteration will be repeated until all topics are checked. At each iteration, one topic minimally will be checked. Hence, CD-MAX algorithm need $\Theta(|N|^2 * |M| * |M|) = \Theta(|N|^2 * |M|^2)$time to check all topics. Our algorithm builds an overlay with diameter at most at 2, because it constructs a star topology for each topic.

5.4 Examples

In order to present the concept of our algorithm, five different examples are demonstrated. None of the existing algorithms presented can build an overlay with lower than the minimum maximum degree achieved by CD-MAX algorithm. These examples play a key role in clarifying the concept of the CD-MAX algorithm.

Example 5.1: Maximum node degree of all the algorithms is the same

As illustrated in Table 5.1, each node is interested in topics 10 and 20. According to Figure 5.1, all algorithms CD-ODA, CD-ODA I, CD-ODA II and CD-MAX require $n - 1$ to construct the overlay with a diameter of 2. In addition, the maximum degree of all these algorithms is $n - 1$.

Nodes	Topics
$\boldsymbol{N_n}$	$\{10,20\}$

Table 5.1: Topic Assignment for Example 5.1

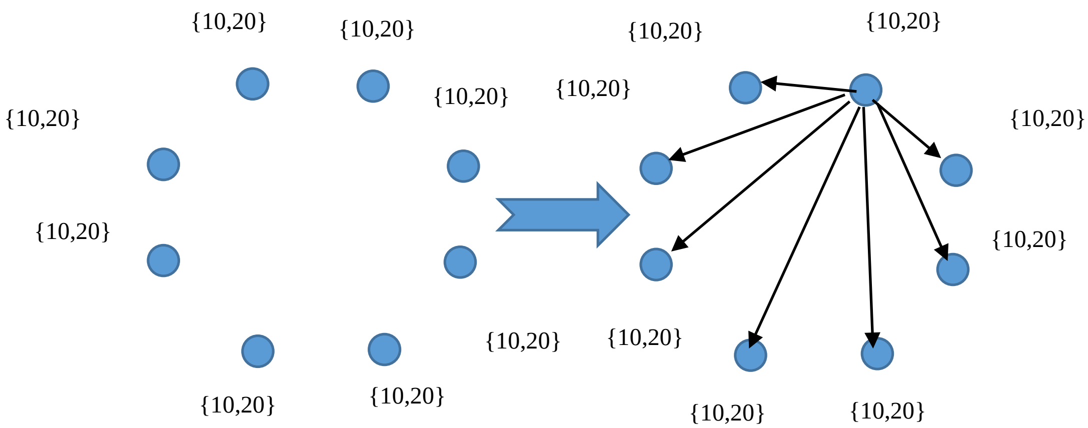


Figure 5.1: Implementation of CD-MAX Algorithm over Example 5.1

Example 5.2: CD-MAX Improved the Maximum Node Degree of the Overlay

Based on Table 5.2, node collection A is interested in topics10 and 20. Node collection B subscribed to topics 10 and 30. Node collection C subscribed to topics 20 and 30. Node collection D subscribed to topic 10 and 40. Note that each collection comprises an unknown number of nodes, in this case n.

To construct the overlay based on CD-ODA, CD-ODA I, CD-ODA II and CD-MAX, we need $6n - 2$ edges. The first three existing algorithms have a maximum node degree at a rate of $4n - 1$. However, CD-MAX builds the overlay network at a maximum of $3n - 1$. The red and black arrows demonstrate the implementation of CD-MAX and other existing algorithms respectively (Figure 5.2). Note that, CD-MAX refinement could not decrease the maximum node degree for this example.

Nodes	Topics
Collection *A*	{10,20}
Collection *B*	{10,30}
Collection *C*	{20,30}
Collection *D*	{10,40}

Table 5.2: Topic Assignment for example 5.2

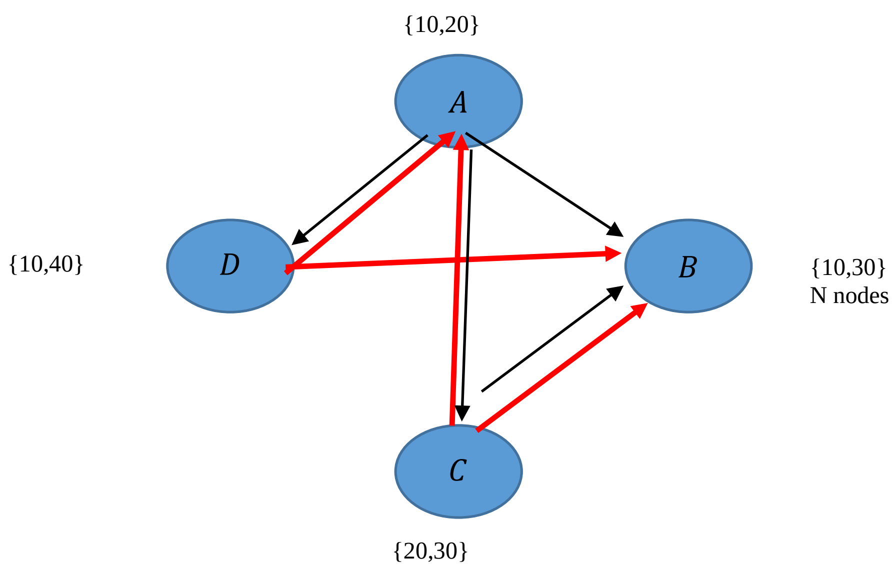


Figure 5.2: Implementation of CD-MAX Algorithm over Example 5.2

Example 5.3: Maximum node degree of CD-MAX algorithm is lower

As shown in Table 5.3, $(n-1)/4$ nodes are interested in topics $\{10,20\}$, $(n-1)/4$ nodes are interested in topics $\{20,30\}$, $(n-1)/4$ nodes are interested in topics $\{30,40\}$ and $(n-1)/4$ nodes are interested in topics $\{40,50\}$. In addition, node u subscribes to topics $\{10,20,30,40,50,60\}$. According to the three existing algorithms, node u is the center of all topics and it would be connected to all nodes. As it is obvious in Figure 5.3, the CD-MAX algorithm provides the overlay at a $\left(2 * \frac{n-1}{4}\right) + 1$ maximum degree. For this example, the CD-MAX algorithm improved the maximum degree of the overlay at 50% lower than the others. Same as previous example, the CD-MAX $refinement$ cannot improve node degree of the overlay.

Nodes	Topics
Set A	$\{10,20\}$
Set B	$\{20,30\}$
Set C	$\{30,40\}$
Set D	$\{40,50\}$
$\boldsymbol{u}$	$\{10,20,30,40,50\}$

Table 5.3: Topic Assignment for Example 5.3

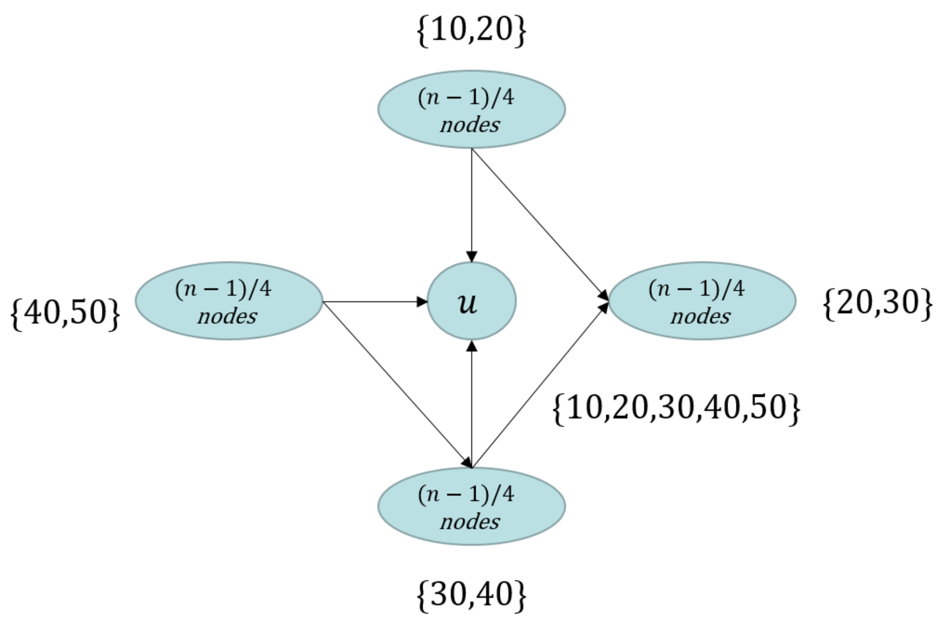


Figure 5.3: Implementation of CD-MAX Algorithm over Example 5.3

Example 5.4: CD-MAX Refinement Improves the Maximum Node Degree of the Overlay

For this example (Figure 5.4), there are $\left(\frac{3n}{2}\right) + 1$ nodes placed over the network. It can be seen that each node subscribes to specific topics. To construct the overlay network, for all three algorithms (CD-ODA, CD-ODAI and CD-ODA II), the node interested in topics $\{x_1, x_2, x_3, \ldots, x_n\}$ acts as a center of the overlay which will be connected to all other nodes participating in the network.

However, the CD-MAX functions in a different manner. For example, the node which subscribes to x_1 is connected to the nodes which are interested in $x_{1,2}$ and $x_1, x_2, x_3, \ldots, x_n$. This algorithm provides the overlay with a maximum node degree n related to the node interested in topics $\{x_1, x_2, x_3, \ldots, x_n\}$. However, it is not the end of implementation. It is the time for CD-MAX $refinement$ to re-construct the overlay with a lower maximum node degree provided by the normal CD-MAX. For this example, node u that is interested in $x_1, x_2, x_3, \ldots, x_n$ has maximum node degree. All edges which are connected to this node is removed (Grey Arrows) and CD-MAX $refinement$ finds other nodes with lower node degree.

Nodes	Topics
$\boldsymbol{N_i}$	$\{x_i\}$
$\boldsymbol{N_{i,j}}$	$\{x_{i,j}\}$
$\boldsymbol{N_{1,2,3,\ldots,n}}$	$\{x_{1,2,3,\ldots,n}\}$

Table 5.4: Topic Assignment for Example 5.4

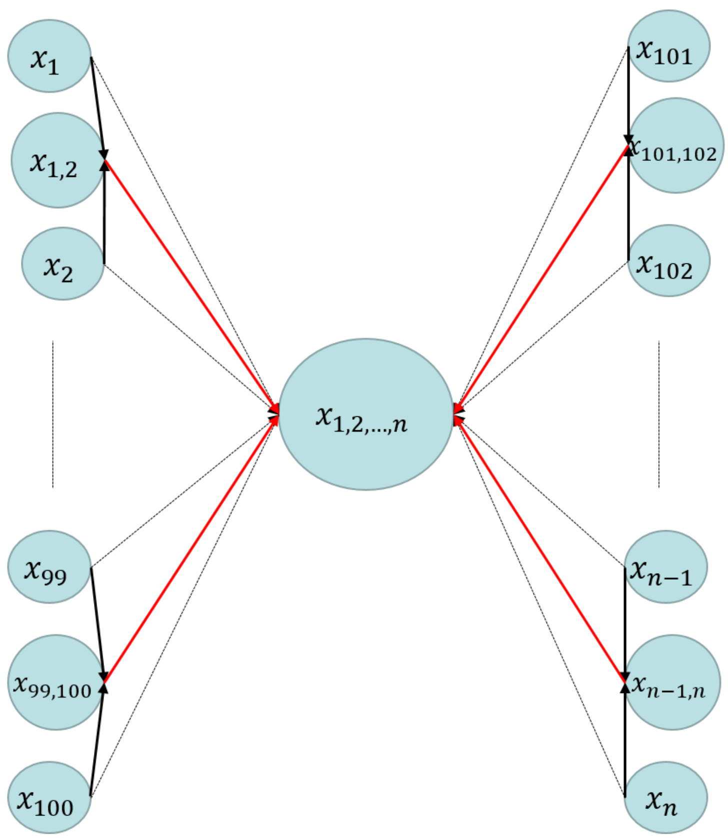


Figure 5.4: Implementation of CD-MAX Algorithm over Example 5.4

These new nodes ($N_{i,j}$) take the responsibility of the topics which are subscribed by node u. Note that nodes $N_{i,j}$ have priority over nodes $N_i$and $N_j$through the CD-MAX $refinement$ implementation.

The red arrows imply the new edges that are added by the CD-MAX $refinement$ algorithm. For this example, CD-MAX $refinement$ plays a key role in decreasing the maximum node degree. This algorithm provides the overlay with a $(n)/2$ maximum node degree.

Example 5.5: Maximum Node Degree of the Overlay is improved by CD-MAX refinement

As Figure 5.5 illustrates, there are 8 different number of nodes located throughout the network. Table 5.5 shows the nodes with their respective topics deployed over the network. The nodes degree are shown in Table 5.6. To implement CD-MAX, the following procedure is used.

Nodes	Topics
0	{1,2,5,7,8}
1	{0,1,3,5,7,8,9}
2	{1,4,5}
3	{2,4,6}
4	{0,2,3,4,9}
5	{2,3,6}
6	{2,5}
7	{1,6}

Table 5.5: Topic Assignment for Example 5.5

As it is obvious in Table 5.6, node number 7 has the lowest node degree of all the other nodes. Hence, it is selected to be the first node to be connected to its neighbors. The node becomes center of topics $\{1, 6\}$. Therefore, topics 1 and 6 are removed from the original topic list and the topics set for step 1 will be {0,2,3,4,5,7,8,9}.

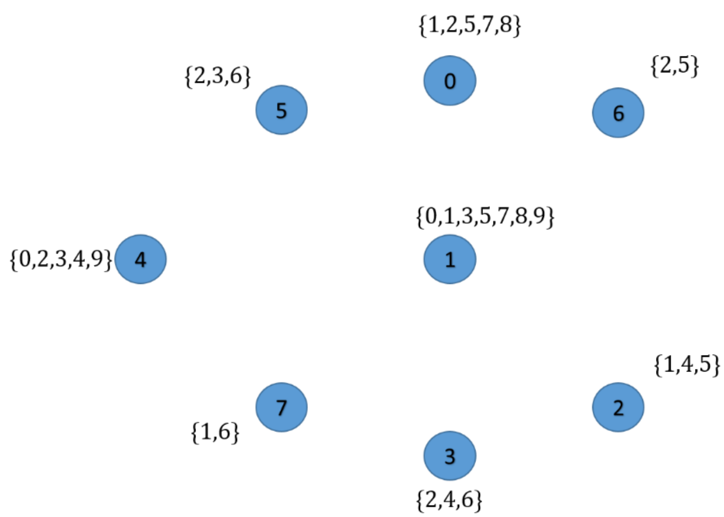


Figure 5.5: Implementation of CD-MAX Algorithm over Example 5.5 (Part 1)

Node	Node Degree	Node Density
0	7	------
1	6	------
2	6	------
3	6	------
4	6	------
5	6	------
6	6	------
7	5	------

Table 5.6: Node Degree and Node Density for Example 5.5 (Part 1)

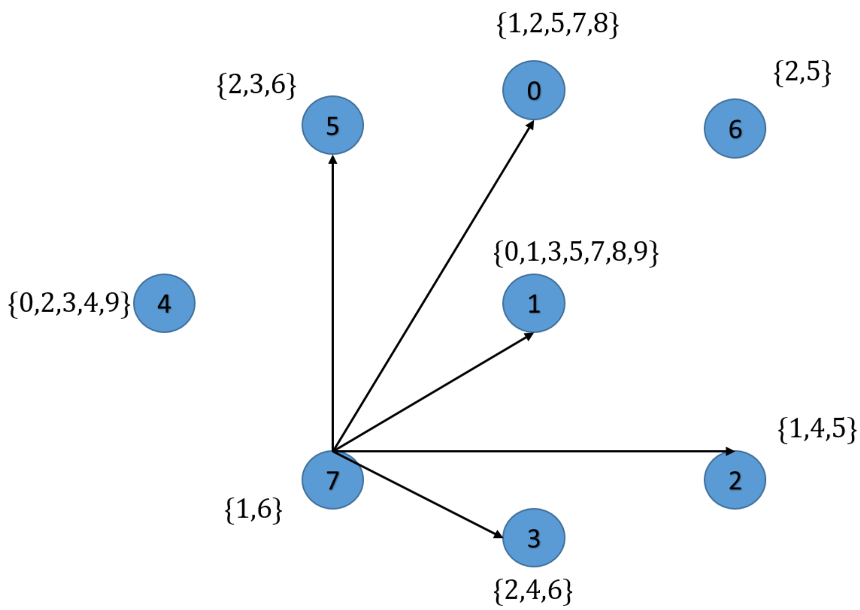


Figure 5.6: Implementation of CD-MAX Algorithm over Example 5.5 (Part 2)

As explained previously, when more than two nodes have an equal lowest node degree, the node with the higher node density will be selected as a node center.

In the second step, nodes 1 through 6 have the lowest node degrees. However, node 1 is the node which has the highest node density (Table 5.7). Hence, this node is selected to be connected to its neighbors (Figure 5.7). As a result, node 1 becomes the center of topics {0,1,3,5,7,8,9}. The remaining topic list will be {2,4}.

In the following steps, shown in Figures 5.8 and 5.9, nodes number 2 and 6 are selected to become the center of topics 4 and 2 respectively. Consequently topics 4 and 2 are removed from the topic list.

Node	Node Degree	Node Density
0	7	9/6
1	6	9/5
2	6	5/5
3	6	6/5
4	6	10/6
5	6	6/5
6	6	7/5

Table 5.7: Node Degree and Node Density for Example 5.5 (Part 2)

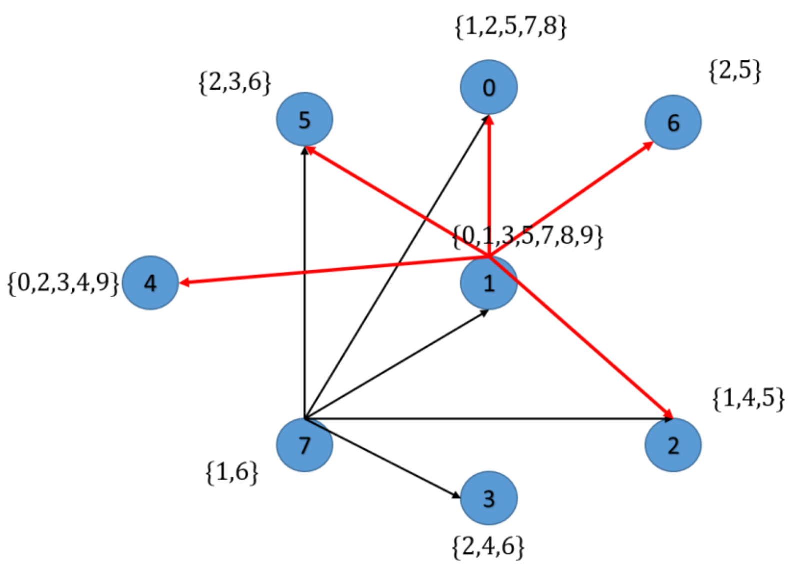


Figure 5.7: Implementation of CD-MAX Algorithm over Example 5.5 (Part 3)

Node	Node Degree	Node Density
0	6	-----
2	4	-----
3	6	-----
4	6	-----
5	6	-----
6	5	-----

Table 5.8: Node Degree and Node Density for Example 5.5 (Part 3)

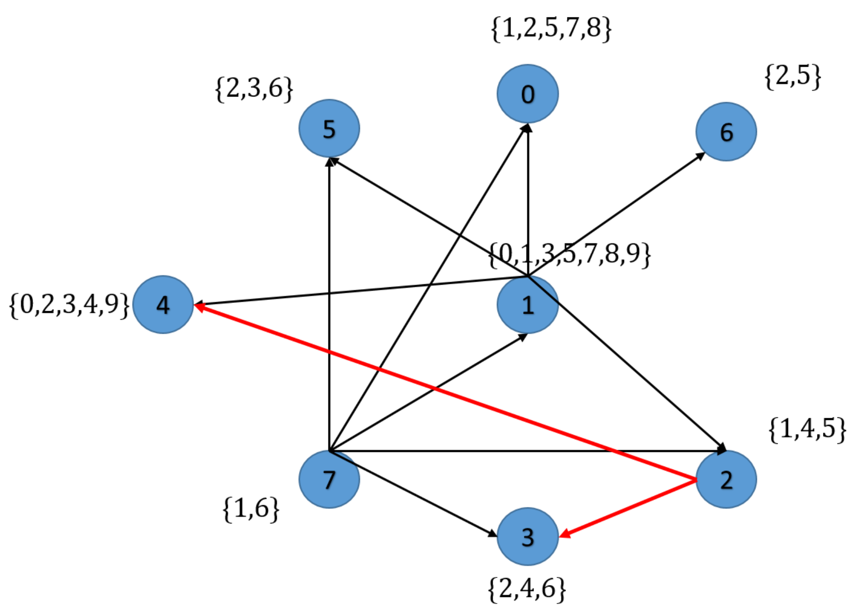


Figure 5.8: Implementation of CD-MAX Algorithm over Example 5.5 (Part 4)

Node	Node Degree	Node Density
0	6	-----
3	6	-----
4	6	-----
5	6	-----
6	5	-----

Table 5.9: Node Degree and Node Density for Example 5.5 (Part 4)

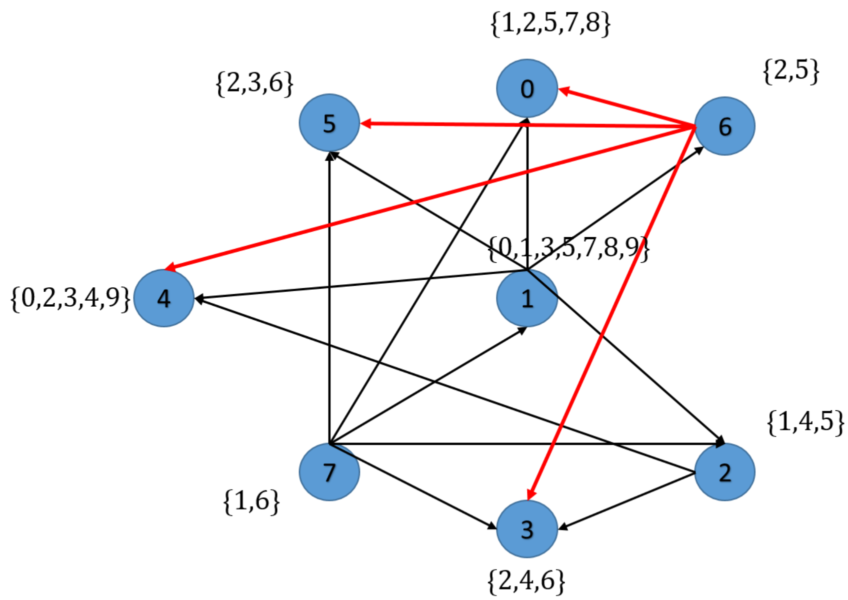


Figure 5.9: Implementation of CD-MAX Algorithm over Example 5.5 (Part 5)

Figure 5.9 demonstrates the result of NORMAL CD-MAX implementation over Example 5.5. For this overlay, CD-MAX provides the overlay with maximum node degree of 6.

However, CD-MAX $refinement$ can decrease the maximum node degree. This algorithm checks the nodes with a maximum node degree. The first node which is checked is node number 1.

Therefore, all edges related to node 1 will be removed and CD-MAX $refinement$ finds another nodes which are interested in topics {0,3,5,7,8,9} and have a lower node degree. As presented in Figure 5.10, node number 0 and 4 are selected to be the center of topics sets {5,7,8} and {0,3,9} respectively.

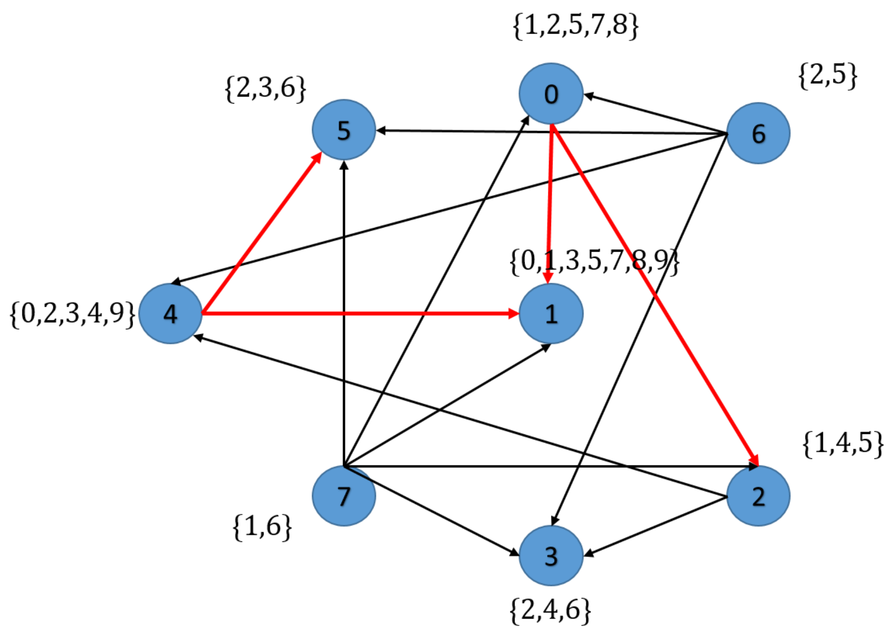


Figure 5.10: Implementation of CD-MAX Algorithm over Example 5.5 (Part 6)

Although CD-MAX $refinement$ decreased the node degree of node number 1. Node number 7 still maintains the maximum node degree of the overlay at 5. Therefore, CD-MAX $refinement$ should find alternative nodes for topics 1 and 6. All edges joined to node number 7 are removed as nodes number 1 and 5 become the center of topics 1 and 6 respectively (Figure 5.11).

As a result, the maximum node degree of the overlay decreases by 2. If each node is to be considered as a collection of nodes comprising an unknown number of nodes say n, the maximum node degree of the overlay provided by CD-MAX would be $5n - 1$, because of the original node degree of 4. Meanwhile this number for CD-ODA as well as for CD-ODA I is $8n - 1$. In the case of CD-ODA II, it is $7n - 1$.

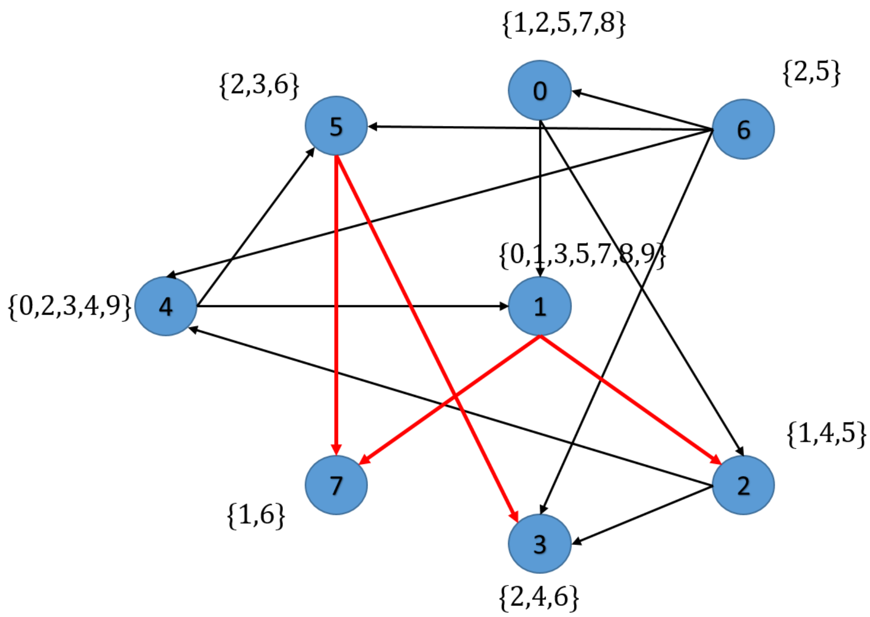


Figure 5.11: Implementation of CD-MAX Algorithm over Example 5.5 (Part 7)

Table 5.1, shows the results of all existing algorithms and CD-MAX algorithms that are implemented over five different examples, together with optimum solution. As shown, CD-MAX achieved the optimum solution.

	Example II	Example III	Example IV	Example V	Example IV
CD-ODA	$n-1$	$6n-2$	$n-1$	$3n/2$	$8n-1$
CD-ODA I	$n-1$	$6n-2$	$n-1$	$3n/2$	$8n-1$
CD-ODA II	$n-1$	$6n-2$	$n-1$	$3n/2$	$7n-1$
CD-MAX	$n-1$	$4n-1$	$\left(2*\frac{n-1}{4}\right)+1$	$n/2$	$5n-1$
Optimum	$n-1$	$4n-1$	$\left(2*\frac{n-1}{4}\right)+1$	$n/2$	$5n-1$

Table 5.10: Maximum Node Degree of Overlays Networks Designed by four Algorithms

Chapter VI

Results

6.1 Introduction

The algorithms presented in this thesis are implemented in the C++ programming language. These algorithms are compared on the basis of the average and maximum node degree over the resulting overlays. In comparison with the GM algorithm, the CD-MAX algorithm produces at least 2.02 times more edges. However, the diameter provided by CD-MAX is significantly lower than GM. In fact, CD-MAX maintains the diameter of the overlay at most at 2. Meanwhile, this value for the GM algorithm would be $\Theta(n)$, where the value of n is equal to the number of nodes. In addition, the maximum node degree provided by these two algorithms are approximately equal. Compared with all constant diameter algorithms, CD-MAX provides the same diameter. However, the algorithm noticeably improved the maximum node degree.

In fact, the CD-MAX algorithm provides a desirable rate of diameter and maximum degree for each overlay. In this chapter, each algorithm presented in this thesis is implemented and simulated through different overlay networks. Both the number of topics and the number of nodes will vary through the simulation. As noted earlier, each node has specific subscription size and due to memory restriction in this experiment each node subscribes to 10 topics. This number is changed for only one simulation between 15 and 35 topics. Moreover, each node $n \in N$ can be interested in each topic $m \in M$ with a probability of p_i in which $\sum_i p_i = 1$.
The topic distribution are selected according to the studies in [2] [46]. In this chapter, we consider how varying the number of nodes, topics and topic popularity distribution affect the average and maximum node degree.

6.2 The Average and Maximum Node Degree as the Number of Nodes Changes

In this step of the experiment, the number of nodes is being changed between 200 to 400 nodes. But, the topic quantity would be stayed at constant number 100. As cited previously, subscription size is fixed at 10 and each node subscribes to different topics randomly. The average node degree of the overlays computed via following formula is illustrated.

$$(2 \times total\ Number\ of\ Edges)/(Total\ Number\ of\ Nodes)$$

Figure 6.1, demonstrates the average node degree of all the algorithms. The average node degree provided by all the existing algorithms decreases slightly as the number of nodes grows. Since more nodes are added throughout the network, the possibility of joining nodes with a higher correlation will increase. Therefore, with a lower number of edges, more nodes are connected and the average node degree of the overlay will decrease and the maximum node degree of the constant diameter algorithms will increase (Figure 6.2).
However, the average node degrees of CD-MAX and CD-MAX $refinement$ increase, as the number of nodes grows. Unlike the other constant diameter algorithms, in which a small number of nodes covers most topics, in order to decrease the node degree of the overlay, the CD-MAX and CD-MAX $refinement$ algorithm requires nodes with a lower correlation to become the center of the topics. Hence, more edges would be needed to connect the nodes thereby raising the average node degree.

Compared with the GM algorithm, CD-MAX and CD-MAX refinement require 2.02 and 2.74 times more edges respectively. Table 6.1 shows the average node degree resulting from the algorithms for different node quantities. To consider the maximum node degree with a growing number of nodes, more nodes should be connected to the star nodes. Hence, the maximum node degree provided by every constant diameter algorithm would increase sharply (Figure 6.2). However, the maximum node degree of the GM algorithm will decrease because the node degree of the overlay is distributed by more nodes with a higher correlation. The most important point about the GM algorithm is that, occasionally a node which has the highest correlation with many other nodes will appeare in the network. This node increases the maximum node degree considerably as it is connected to many nodes in a star topology (Table 6.2, the GM algorithm, overlay with 350 topics).

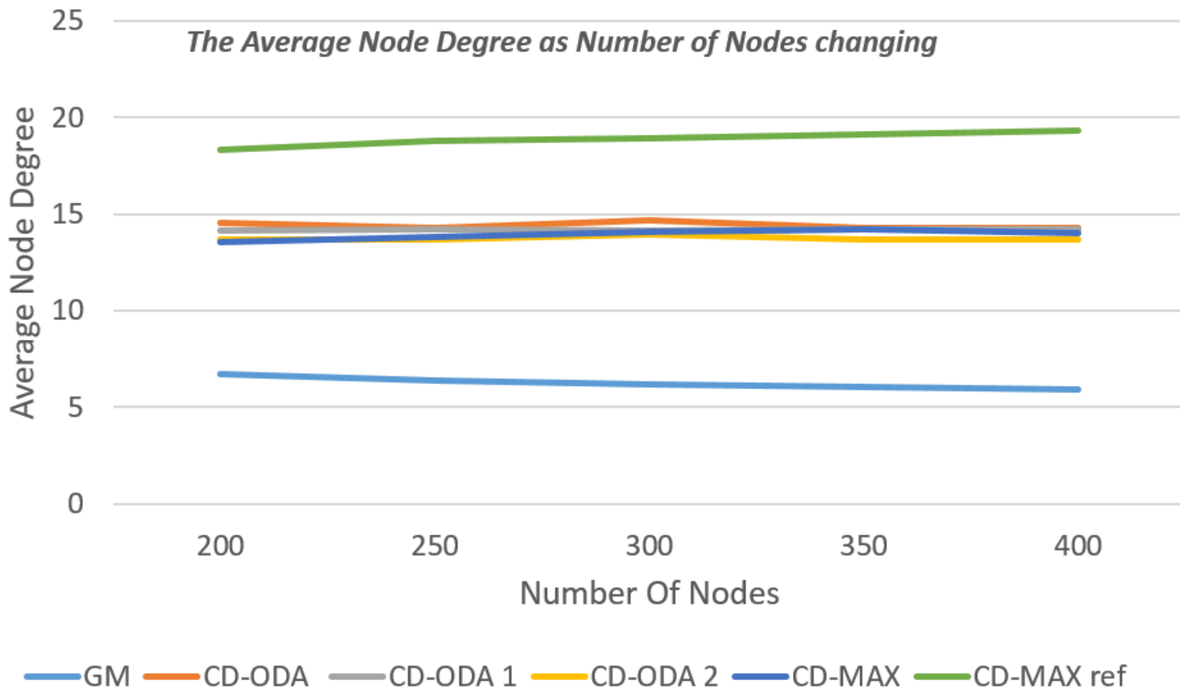


Figure 6.1: Average Node Degree for Different Number of Nodes

	GM	$CD-ODA$	$CD-ODA$ I	$CD-ODA$ II	$CD-MAX$	$CD-MAX_{ref}$
200	6.69	14.55	14.2	13.7	13.52	18.34
250	6.4	14.3	14.2	13.7	13.8	18.8
300	6.17	14.7	14.2	13.94	14.1	18.9
350	6.07	14.3	14.2	13.7	14.2	19.1
400	5.9	14.3	14.1	13.6	14.2	19.3

Table 6.1: Average Node Degree for Different Number of Nodes

Although GM algorithm has low maximum and average node degree, but it has higher diameter [6] [21] [2].

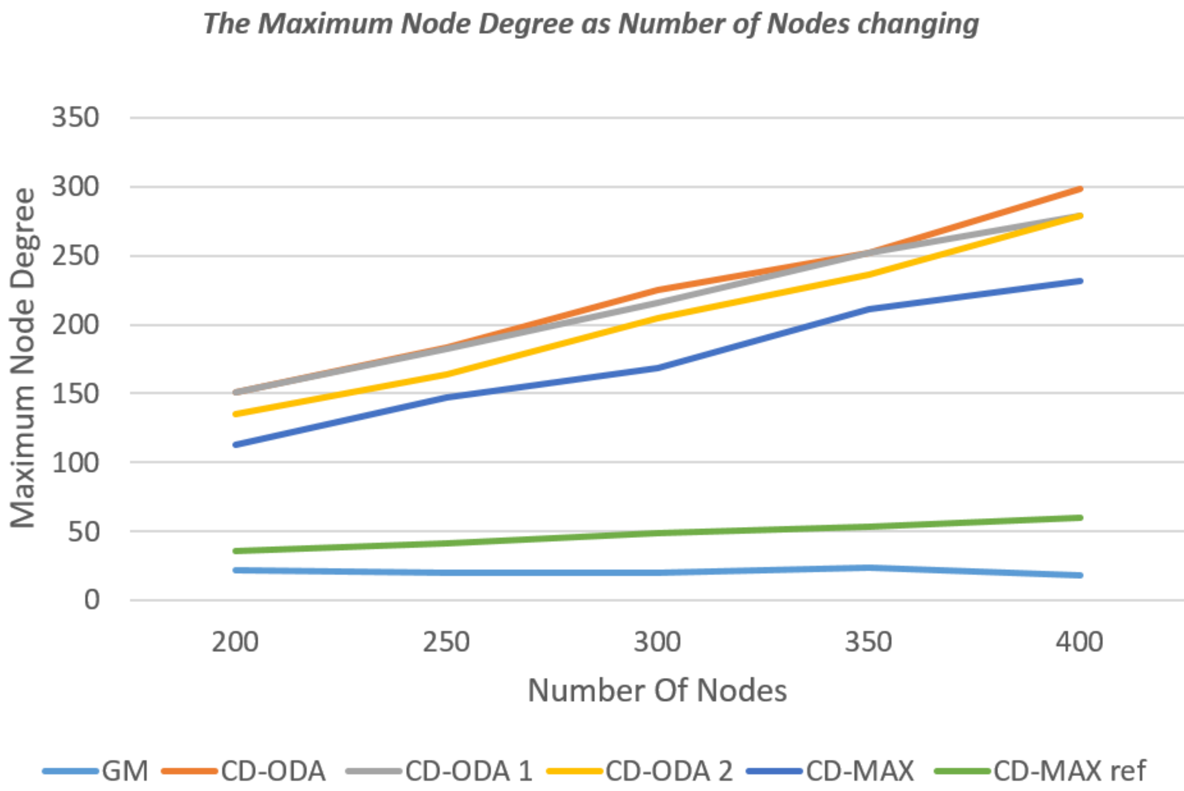


Figure 6.2: Maximum Node Degree for Different Number of Nodes

	GM	$CD-ODA$	$CD-ODA$ I	$CD-ODA$ II	$CD-MAX$	$CD-MAX_{ref}$
200	22	151	151	135	113	36
250	20	183	182	164	147	41
300	20	225	216	205	168	49
350	24	252	252	236	211	53
400	18	298	279	279	232	60

Table 6.2: Maximum Node Degree for Different Number of Nodes

6.3 The Average and Maximum Node Degree as the Number of Topics Changes

Unlike experiment 6.2, the number of nodes is fixed at 100 nodes and the number of topics varies from 200 to 400 topics. Similarity to the previous part, subscription size is assigned at 10 topics. When the number of topics is increased, the overlay will face two different conditions. Firstly, the correlation between nodes will become lower, so more edges will be used to connect the nodes. Secondly, the number of nodes which lacks neighbors will also increase. If the first condition overcomes the second, the average node degree will increase. CD-MAX refinement for 250 as the number of topics in Figure 6.3 is a sign of the effect of the first condition. However, if the second condition dominates the first, the average node degree will decrease. Overall, these states have the most effect on the average degree of the overlay. As Figure 6.3 shows, the second condition has a greater effect than the first condition. Hence, the overall average node degree of the overlay for every algorithm reduces when the quantity of topics increases.

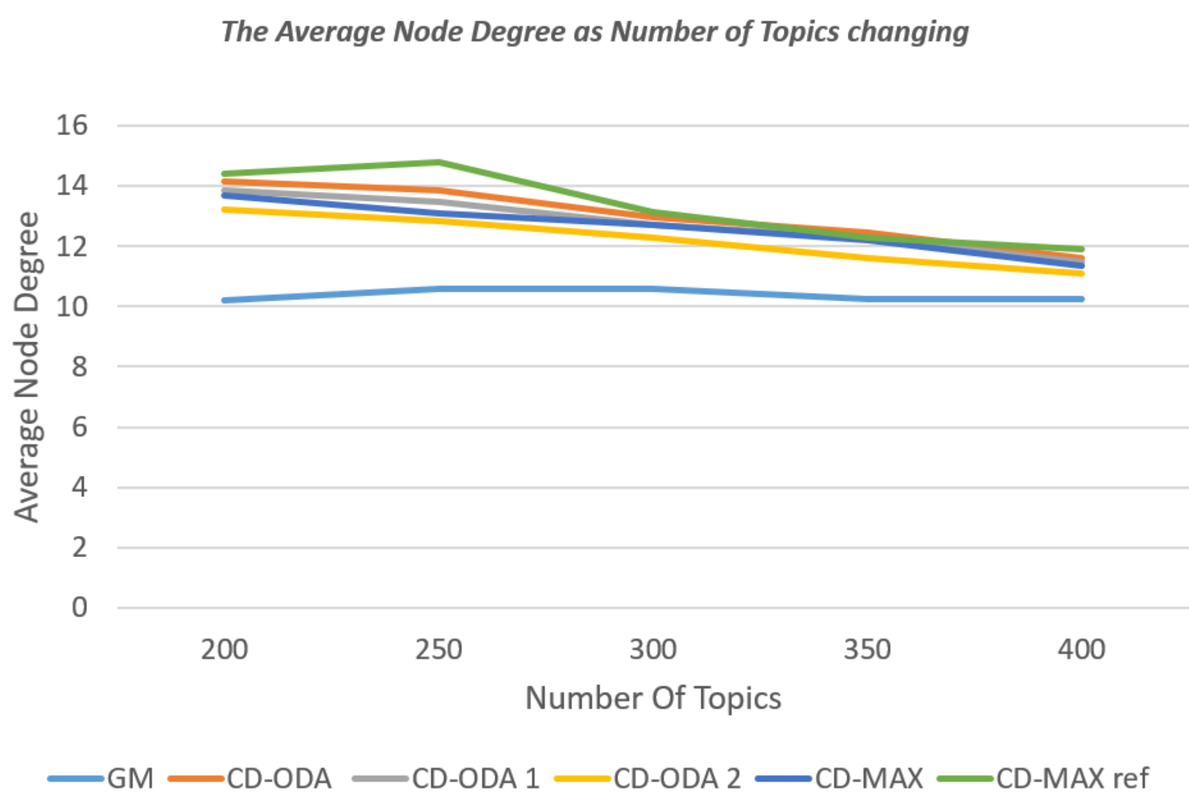


Figure 6.3: Average Node Degree for Different Number of Topics

	GM	$CD-ODA$	$CD-ODA$ I	$CD-ODA$ II	$CD-MAX$	$CD-MAX_{ref}$
200	*10.22*	*14.14*	*13.84*	*13.22*	*13.68*	*14.4*
250	*10.68*	*13.84*	*13.48*	*12.82*	*12.08*	*14.8*
300	*10.58*	*12.96*	*12.72*	*12.3*	*12.72*	*13.12*
350	*10.24*	*12.44*	*12.28*	*11.62*	*12.22*	*12.28*
400	*10.10*	*11.62*	*11.48*	*11.08*	*11.36*	*11.92*

Table 6.3: Average Node Degree for Different Number of Topics

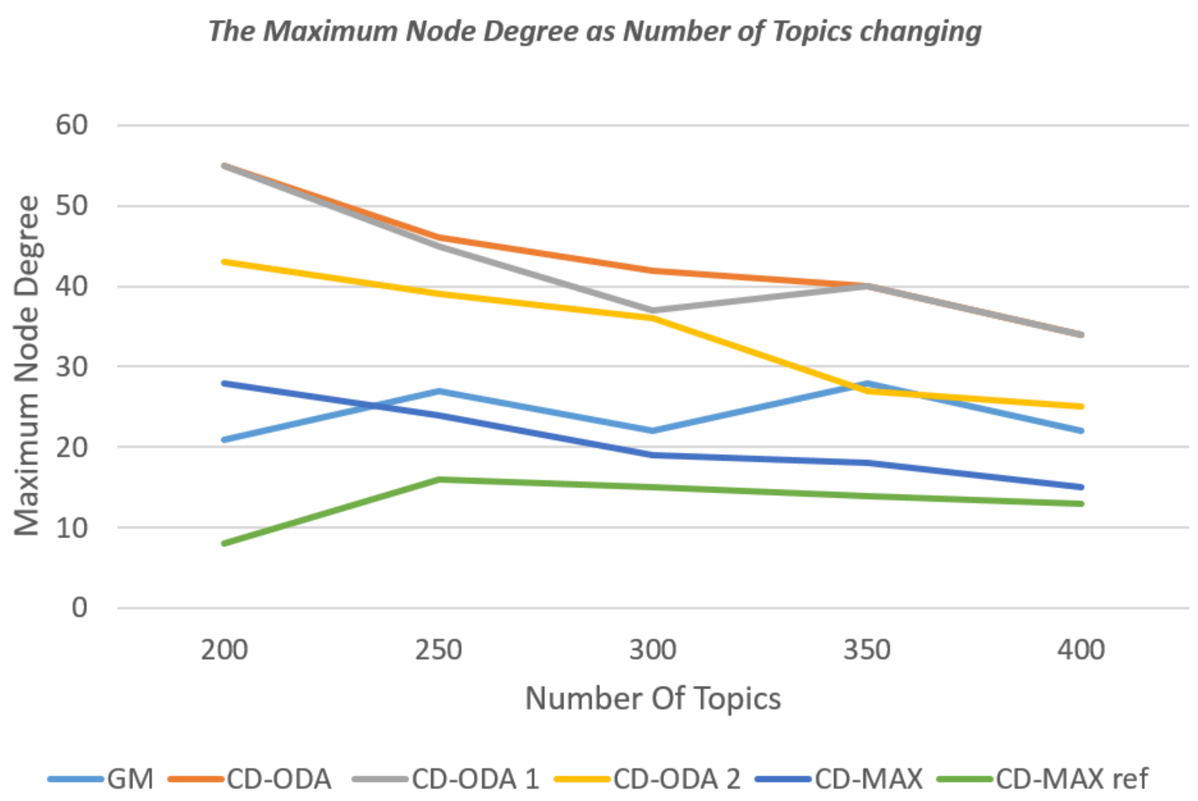


Figure 6.4: Maximum Node Degree for Different Number of Topics

	GM	$CD\text{-}ODA$	$CD\text{-}ODA\ I$	$CD\text{-}ODA\ II$	$CD\text{-}MAX$	$CD-MAX_{ref}$
200	21	55	55	43	28	18
250	27	46	45	39	24	16
300	22	42	37	36	19	15
350	28	40	40	27	18	14
400	22	34	34	25	15	13

Table 6.4: Maximum Node Degree for Different Number of Topics

In addition, this state (condition 2) has affected the maximum degree of all algorithms. As can be seen in Figure 6.4, the maximum node degree of all algorithms are decreasing, as the topics set includes more different topics.

6.4 The Average and Maximum Node Degree as the Subscription Size Changes

As final experiment, quantity of nodes and topics are kept at 200 and 100 respectively. However, the subscription size starts changing between 15 to 35 numbers. As noted before, each node subscribes to different topics randomly by $Interest$ function. Figure 6.5 and 6.6 illustrate the effects of changing subscription size on the presented Algorithms in this dissertation. When subscription size grows, nodes can get connected with each other with higher rate of correlation. Subsequently, rate of average node degree will be decreased.

On the other hand, when subscription size is growing, the contribution between each node will rise. Therefore, the GM algorithm can find many pairs of nodes, which reduces the total topic connected component dramatically. Hence, the maximum node degree will decrease as the subscription size increases.

For all algorithms with star topology, maximum node degree will increase as only nodes are chosen to be the center of many topics in order to be connected with their neighbors.

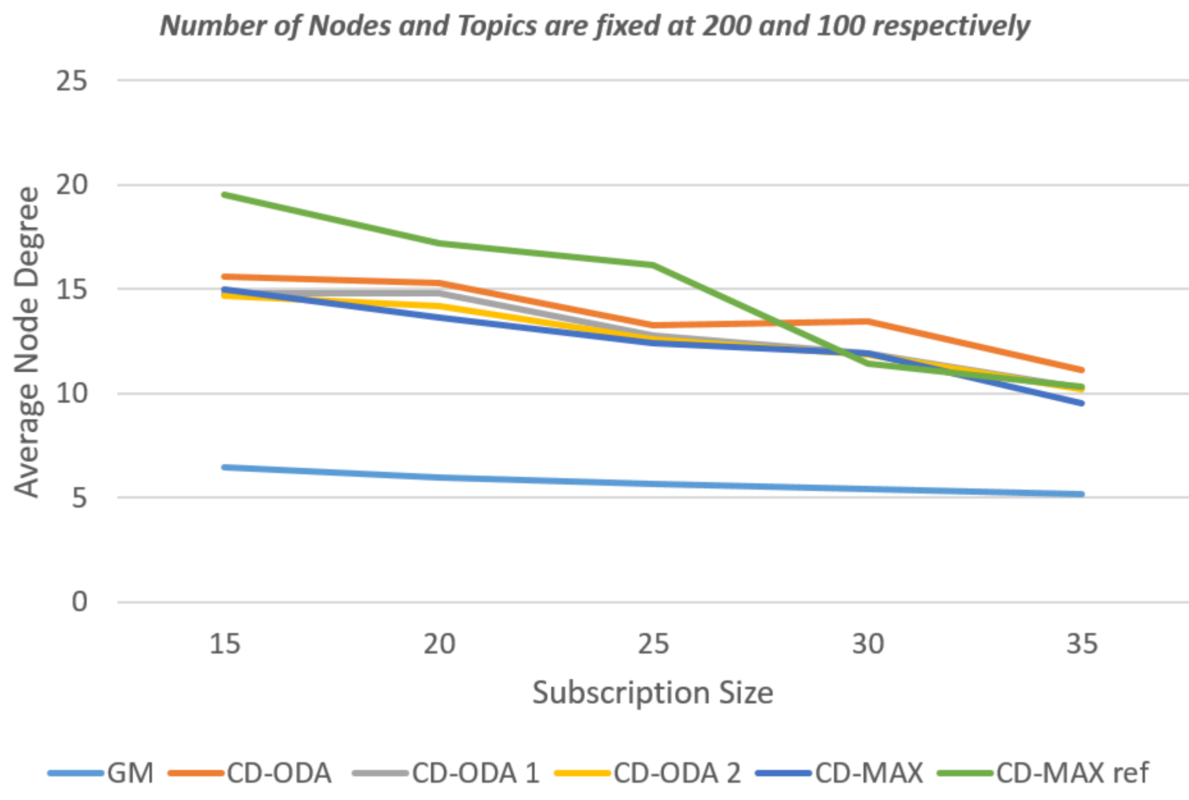


Figure 6.5: Average Node Degree for Different Subscription Size

	GM	*CD-ODA*	*CD-ODA I*	*CD-ODA II*	*CD-MAX*	$CD-MAX_{ref}$
15	*6.49*	*15.63*	*14.83*	*14.69*	*15.01*	*19.56*
20	*6*	*15.33*	*14.8*	*14.2*	*13.65*	*17.21*
25	*5.7*	*13.27*	*12.79*	*12.62*	*12.42*	*16.19*
30	*5.42*	*13.46*	*11.9*	*11.87*	*11.94*	*11.42*
35	*5.17*	*11.16*	*10.29*	*10.24*	*9.51*	*10.33*

Table 6.5: Average Node Degree for Different Subscription Size

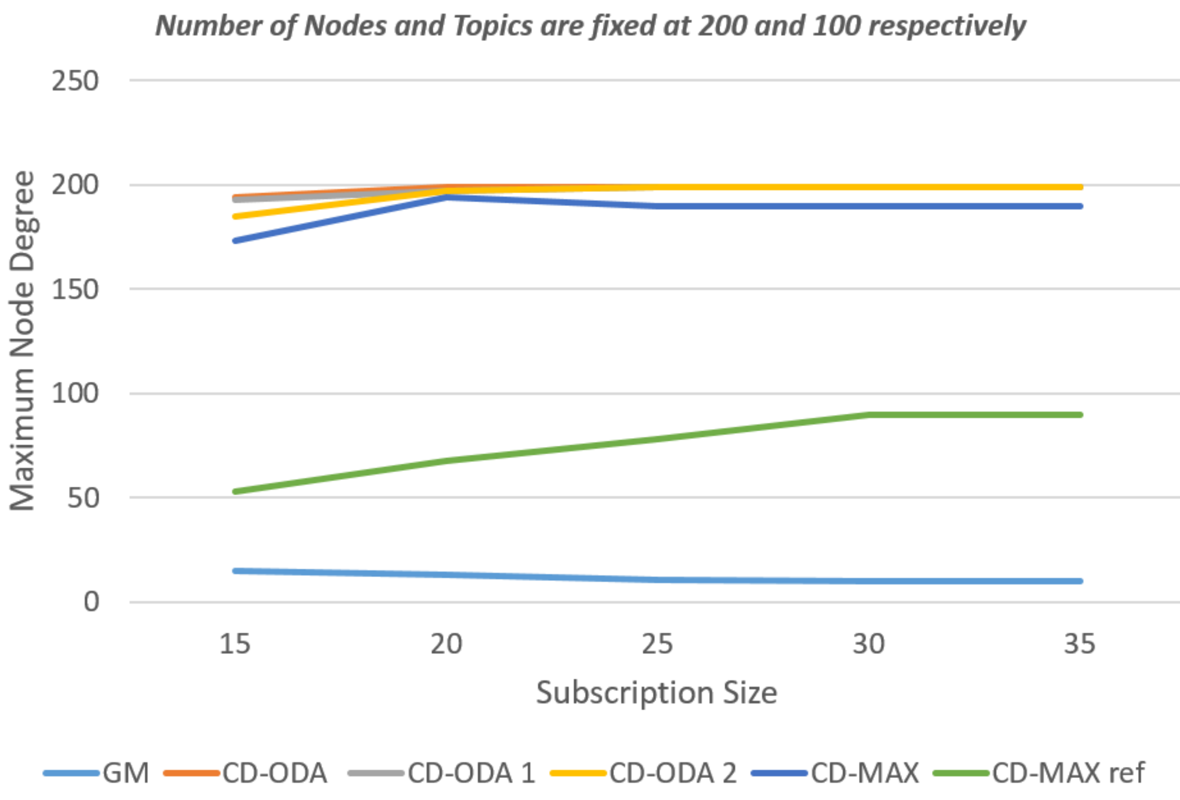


Figure 6.6: Maximum Node Degree for Different Subscription Size

	GM	*CD-ODA*	*CD-ODA I*	*CD-ODA II*	*CD-MAX*	*CD-MAX ref*
15	15	194	193	185	173	53
20	13	199	197	197	194	68
25	11	199	199	199	190	78
30	10	199	199	199	190	90
35	10	199	199	199	190	90

Table 6.6: Maximum Node Degree for Different Subscription Size

CHAPTER VII

Conclusion

In this dissertation, we presented an algorithm (CD-MAX) that provides overlay networks with a minimum maximum node degree and low diameter. The algorithm is implemented in a decentralized manner, thus it is faster. It considerably decreases the maximum degree, thus the resulting overlay network becomes more scalable compared to the other algorithms studied. According to existing studies in the field, minimizing the maximum degree plays a key role in a number of network domains, such as survivable and wireless networks [3] [4] [8] [9] [10] [47].

Chockler and Onus [31] studied a number of algorithms to decrease the diameter and average node degree of overlay networks. However, their solutions do not have much success in reducing the maximum node degree and diameter of an overlay network simultaneously. This problem is considered to be an important challenge in the design of effective pub/sub systems. Our solution is deduced from the Constant Diameter Overlay Design Algorithm II (CD-ODA II) presented by Onus [31]. In this case, design complexity is at logarithmic level.

Compared with the GM algorithm, CD-MAX needs 2.02 times more connections. In addition, this value reaches 2.07 when $refinement$ is applied which can reduce the maximum degree even further.

Amongst all the presented algorithms, CD-MAX with $refinement$, achieves the best results in terms of node degree. According to Experiment 6.3, for an overlay network with 100 nodes and 200 topics and a subscription size 10, CD-MAX $refinement$ produces a maximum degree of 18. Meanwhile this value for GM is 21. Although there is no significant difference between the two algorithms, the overlay resulting from CD-MAX $refinement$ has a much lower diameter. Moreover, the maximum node degree stemming from the existing low diameter algorithms for Experiment 6.4 is 2.4, which is greater than that of the CD-MAX $refinement$ algorithm.

The CD-MAX algorithm cannot satisfy pub/sub network designers in terms of minimization of number of links. According to experiments, CD-MAX $refinement$ needs many more connections to build a network. For future work, more effort needs to be put into constructing an overlay network with still better maximum degree and minimum diameter clustering [11] [12] [13] [48].

.

REFERENCES

1. **Beverly Yang and Hector Garcia-Molina (2003),** Designing a super-peer network. In Proceeding of 19th International Conference on Data Engineering, page 49.

2. **F. Araujo, L. Rodrigues, N. Carvalho, and N. Carvalho (2005),** "Scalable QoS-based event routing in publish-subscribe systems," in NCA'05.

3. **S. Voulgaris, E. Riviere, A. M. Kermarrec, and M. van Steen (2006),** “Sub-2- sub: Self-organizing content-based publish subscribe for dynamic large scale collaborative networks,” in Proc. IPTPS, pp. 123–128.

4. **M. Altherr,M. Erzberger, and S. Maffeis (1999),** “iBusa software bus middleware for the Java platform,” in Proc. Int. Workshop Rel. Middleware Syst, pp. 43–53.

5. **TIBCO (1999),** “TIB/Rendezvous,” Tibco, Palo Alto, CA, White paper.

6. **Gregory Chockler, Roie Melamed , Yoav Tock(2007),** Constructing Scalable Overlay for pub-sub with Many Topics, pp.199-118.

7. **R. Baldoni, R. Beraldi, V. Quema, L. Querzoni, and S. T. Piergiovanni (2007),** “TERA: Topic-based event routing for peer-to-peer architectures,” in Proc. 1st ACM DEBS, pp. 2–13.

8. “Everything you need to know about middleware: Mission-critical interprocess communication,” Talarian Corporation, Los Altos, CA, White paper, 1999 [Online]. Available: http://www.talarian.com/

9. **L.C. Lau, J. Naor,M.R. Salavatipour, and M. Singh (2009),** Survivable Network Design with Degree or Order Constraints. In Procedings of ACM STOC’07.

10. **J. Reumann. GooPS: Pub/Sub at Google (August, 2009),** Lecture at CANOE Summer School, Oslo, Norway.

11. **Chen Chen Jacobsen, H.-A. , Vitenberg, R. (2010),** Divide and Conquer Algorithm for Publish/Subscribe Overlay Design, pp. 622-623.

12. **Chen Chen Tock, Y. , Jacobsen, H.-A. ; Vitenberg, R (2015),**Weighted Overlay Design for Topic-based Publish/Subscribe Systems on Geo-Distributed Data Centers, pp. 474-485.

13. **C. Chen, H.-A. Jacobsen, and R. Vitenberg (2010),** "Star merge and divide-and-conquer algorithms for publish/subscribe topic-connected overlay design," U. of Toronto & U. of Oslo, Tech. Rep., http://msrg.org/papers/TRCJV-DC-2010.

14. **Antonino Virgillito (2003),** Publish/Subscribe Communication System: From models to Applications, Ph.D. Thesis. , UNIVERSITÀ DEGLI STUDI DI ROMA LA SAPIENZA

15. **S. Baehni, P. T. Eugster, and E. Guerraoui (2004), "Data-aware multicast,"** in Proc. DSN, pp. 233–242.

16. **G. Chockler, R. Melamed, Y. Tock and R. Vitenberg. Spidercast (2007)** A scalable interest-aware overlay for topic-based pub/sub communication. In DEBS'07.

17. **G. Li, V. Muthusamy and H.-A. Jacobsen (2010),** "A distributed service oriented architecture for business process execution," ACM TWEB.

18. **S. Banerjee, B. Bhattacharjee, and C. Kommareddy** "Scalable application layer multicast," SIGCOMM Comput. Commun. Rev., vol. 32, no. 4, pp. 205–217.

19. **S. Bhola, R. Strom, S. Bagchi, Y. Zhao, and J. Auerbach(2002),** "Exactly-once delivery in a content-based publish-subscribe system," in Proc. DSN, pp. 7–16.

20. **A. Carzaniga, M. J. Rutherford, and A. L. Wolf (Mar. 2004),** "A routing scheme for content-based networking," in Proc. 23rd Annu. IEEE INFOCOM, Hong Kong, vol. 2, pp. 918–928.

21. **M. Onus, A. W. Richa (August 2009),** Brief Announcement: Parameterized Maximum and Average Degree Approximation in Topic-based Publish-Subscribe Overlay Network Design, 21st ACM Symposium on Parallelism in Algorithms and Architectures (SPAA), Calgary, Canada, pp.644-652.

22. **R. Chand and P. Felber (2005),** "Semantic peer-to-peer overlays for publish/subscribe networks," in Euro-Par 2005 Parallel Processing, Lecture Notes in Computer Science. New York: Springer- erlag, vol.3648, pp. 1194–1204.

23. **R. Guerraoui, S. Handurukande, and A. M. Kermarrec (2004),** "Gossip: A gossip-based structured overlay network for efficient content-based filtering,"EPFL, Lausanne, Tech. Rep. IC/2004/95.

24. **R. Levis (1999),** Advanced Messaging Applications With MSMQ and MQSeries. Indianapolis, IN: QUE.

25. **V. Ramasubramanian, R. Peterson, and E. G. Sirer 2006,** "Corona: A high performance publish-subscribe system for the World Wide Web," in Proc. 3rd NDSI, pp. 15–28.

26. **D. Sandler, A. Mislove, A. Post, and P. Druschel (2005),** "FeedTree: Sharing Web micronews with peer-to-peer event notification," in Proc. IPTPS, pp. 141–151.

27. **D. Tam, R. Azimi, and H.-A. Jacobsen (2003),** "Building content-based publish/ subscribe systems with distributed hash tables," in Proc. 1st IDBISP2P, Berlin, Germany, pp. 138–152.

28. **Y. Tock, N. Naaman, A. Harpaz, and G. Gershindky (2005),** "Hierarchical clustering of message flows in a multicast data dissemination system," in Proc. 17th IASTED Int. Conf. Parallel Distrib. Comput. Syst, pp.320-327.

29. **P.Th. Eugster, P. Felber, R. Guerraoui, and A.M. Kermarrec (2003),** The Many Faces of Publish/Subscribe, ACM Computing Surveys 35, no. 2, pp.114–131.

30. **Sheng Gao ; Sch. of Sci. & Technol., Huazhong Univ. of Sci. & Technol., Wuhan, China ; Guohui Li ; Pei Zhao (2011),** Content-Based publish over structured P2P network, pp.290-294

31. **Onus, M and Richa and A.W(2011),** Minimum Maximum-Degree Publish-Subscribe Overlay Network Design, pp. 1331-1343.

32. **Tova Milo, Tal Zur, Elad Verbin (2007),** Boosting topic-based publish-subscribe systems with dynamic clustering, pp.749-760.

33. **M. Castro, P. Druschel, A. Kermarrec, and A. Rowstron (Oct. 2002),** "SCRIBE: A large-scale and decentralized application-level multicast infrastructure," IEEE J. Sel. Areas Commun., vol. 20, no. 8, pp. 1489–1499.

34. **S. Q. Zhuang, B. Y. Zhao, A. D. Joseph, R. H. Katz, and J. D. Kubiatowicz(2001), Bayeux**(), An architecture for scalable and fault-tolerant wide-area data dissemination. 11th ACM NOSSDAV, pp. 11-20, .01.

35. **G. Banavar, T. Chandra, B. Mukherjee, J. Nagarajarao, R. E. Strom, and D. C. Sturman (1999),** An efficient multicast protocol for content-based publish-subscribe systems. In Proceedings of the 19th ICDCS, pp. 262.272.

36. **A. Carzaniga, D. S. Rosenblum, and A. L. Wolf (2001),** Design and evaluation of a wide-area event notification service. ACM Transactions on Computer Systems, pp.332.383.

37. **P. Triantafillou, A. Economides, Subscription Summarization (2004),** A New Paradigm for Efficient Publish/Subscribe Systems, In IEEE, ICDCS, pp. 562-571.

38. **W. W. Terpstra, S. Behnel, L. Fiege, A. Zeidler, and A. P. Buchmann(2003)**, A Peer-to-Peer Approach to Content-Based Publish/Subscribe, pp.1-8.

39. **P.Triantafillou, and I.Aekaterinidis, (2004),** "Content-based Publish-Subscribe Over Structured P2P Networks" pp.104-109.

40. **D.K. Bradshaw, Kevin MacDowell, Den Raphaely (2001),** Oracle9i Application Developers Guide Advanced Queuing. Redwood Shores, CA: Oracle.

41. **Choon Hoong Ding, Sarana Nutanong, and Rajkumar Buyya(2003),** (BOOK) Peer-to-Peer Networks for Content Sharing

42. **Jay Glynn, Karli Watson, Morgan Skinner (2012),** (BOOK) Professional C# 2012 and .NET 4.5 Christian Nagel, Bill Evjen, chapter 45 (Pee-to-Peer Networking)

43. **Kan, G. (2001),** (BOOK) Gnutella, Peer-to-Peer: Harnessing the Power of Disruptive Technologies, A. Oram (ed.), O’Reilly Press, USA.Survey of Publish Subscribe Event System

44. **E. De Santis, F. Grandoni, and A. Panconesi (2007),** Fast low degree connectivity of ad-hoc networks via percolation. In Proceedings of the European Symposium on Algorithms (ESA), pages 206-217.

45. **T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein (2001),** (BOOK) Introduction to Algorithms. MIT Press.

46. **H. Liu, V. Ramasubramanian, and E. G. Sirer (2005),** "Client behavior and feed characteristics of RSS, a publish-subscribe system for web micronews," in Proc. IMC, Berkeley, CA, pp. 29–34.

47. **D. Skeen (1998),** "Vitrias publish-subscribe architecture: Publish-subscribe overview," Sunnyvale, CA, [Online]. Available: http://www.vitria.com

48. **Chen Chen Tock, Y. ; Jacobsen, H.-A. ; Vitenberg, R (2010), "**Algorithms Based on Divide and Conquer for Topic-Based Publish/Subscribe Overlay Design" pp. 622-633.

APPENDICES A

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: LAYAZALI, Sina

Student ID: 201271453

Date and Place of Birth: 30 April 1989, Tehran

Marital Status: Single

Phone: +90 539 615 9002

Email: sinalayazali@gmail.com

WORK EXPERIENCE

Year	**Place**	**Enrollment**
2008- 2011	SHARP COMPANY.	Project Manager

HONOURS AND AWARDS

1. In 2010, Received Top 3 Robotic Awards in IRAN

FOREIN LANGUAGES

Advanced English, Intermediate Turkish, Beginner French

HOBBIES

Guitar Electric, Online games, Books, Swimming, Travelling, Football