2010 International Conference on Distributed Computing Systems

Parameterized Maximum and Average Degree Approximation in Topic-based Publish-Subscribe Overlay Network Design

Melih Onus
Department of Computer Engineering
TOBB University of Economics and Technology
Ankara, Turkey
Email: monus@etu.edu.tr

Andréa W. Richa
Computer Science and Engineering
School of Computing, Informatics, and Decision Systems Engineering
Arizona State University
Tempe, AZ 85287
Email: aricha@asu.edu

***Abstract*—Publish/subscribe communication systems where nodes subscribe to many different topics of interest are becoming increasingly more common. Designing overlay networks that connect the nodes subscribed to each distinct topic is hence a fundamental problem in these systems. For scalability and efficiency, it is important to keep the degree of the nodes in the publish/subscribe system low. Ideally one would like to be able not only to keep the average degree of the nodes low, but also to ensure that all nodes have equally the same degree, giving rise to the following problem: Given a collection of nodes and their topic subscriptions, connect the nodes into a graph with low average and maximum degree such that for each topic t, the graph induced by the nodes interested in t is connected. We present the first polynomial time parameterized sublinear approximation algorithm for this problem.**

We also propose two heuristics for constructing topic-connected networks with low average degree and constant diameter and validate our results through simulations. In fact, the results in this section are a refinement of the preliminary results by Onus and Richa in INFOCOM'09.

I. Introduction

In publish/subscribe (pub/sub) systems, publishers and subscribers interact in a decoupled fashion. They use logical channels for delivering messages according to the nodes subscription to the services of interest. Publishers publish their messages through logical channels that deliver the messages to the nodes that subscribed to the respective services.

Pub/sub systems can be either topic-based or content-based. In a *topic-based* pub/sub system, messages are published to "topics", where each topic is uniquely associated with a logical channel. Subscribers in a topic-based system will receive all messages published to the topics to which they subscribe. The publisher is responsible for defining the classes of messages to which subscribers can subscribe. In a *content-based* system, messages are only delivered to a subscriber if the attributes of those messages match constraints defined by the subscriber; each logical channel is characterized by a subset of these attributes. The subscriber is responsible for classifying the messages.

Given their simplicity and wide applicability, we have seen many implementations of those systems in recent years (see e.g., [1]–[4], [6]–[10], [16], [19], [24]–[28]), as well as many applications built on top of them, such as stock-market monitoring engines, RSS feeds [20], [27], on-line gaming and several others. For a survey on pub/sub systems, see [15].

We will implement a topic-based pub/sub system by designing a connected (peer-to-peer) overlay network for each pub/sub topic: more specifically, for each topic t, we will enforce that the subgraph induced by the nodes interested in t will be connected. This translates into a *fully decentralized* topic-based pub/sub system since any given topic-based overlay network will be connected and thus nodes subscribed to a given topic do not need to rely on other nodes (agents) for forwarding their messages. Such an overlay network is said to be *topic-connected*.

Low node degrees are desirable in practice for scalability and also due to bandwidth constraints. Nodes with a high number of adjacent links will have to manage all these links (e.g., monitor the availability of its neighbors, incurring in heartbeats and keep-alive state costs, and connection state costs in TCP) and the traffic going through each of the links, without being able to take great advantage of aggregating the traffic (which would also reduce the number of packet headers, responsible for a significant portion of the traffic for small messages). See [11], [21] for further motivation.

The node degrees and number of edges required by a topic-connected overlay network will be low if the node subscriptions are well-correlated. In this case, by connecting two nodes with many coincident topics, one can satisfy connectivity of many topics for those two nodes with just one edge. Several recent empirical studies suggest that correlated workloads are indeed common in practice [20], [27].

In this paper, we focus on building overlay networks with low (average *and* maximum) node degrees. The importance of minimizing both the maximum and average degree has been recognized in some network domains, such as that of survivable network design [18] and that of establishing connectivity in wireless networks [13]. To the best of our

This work was supported in part by NSF awards CCF-0830791 and CCF-0830704.

1063-6927/10 $26.00 © 2010 IEEE
DOI 10.1109/ICDCS.2010.54

IEEE computer society

knowledge, minimizing both the maximum and the average degree in topic-connected pub/sub overlay network design had not been directly addressed prior to this work.

As in many other systems, a space-time trade-off exists for pub-sub systems: On one hand, one would like the total time taken by a topic-based broadcast (which directly depends on the diameter of each topic-connected subnetwork) to be as small as possible; on the other hand, for memory and node bandwidth considerations, one would like to keep the total degree of a node small. Those two measures are often conflicting.

Most of the current solutions adopted in practice actually fail at maintaining *both* the diameter and the node degrees low. Take the naive, albeit popular, solution to topic connected-overlay network design to construct a cycle connecting all nodes interested in a topic independently for each given topic [28]: This construction results both in very large diameter for each topic-connected network (proportional to the total number of nodes subscribed to a topic) and in node degrees proportional to the nodes' subscription sizes, whereas a more careful construction, taking into account the correlations among the node subscription sets might result in much smaller node degrees (and total number of edges) and topic-based diameters. Even the more recent advances on approximating the average or maximum degree alone have been made [11], [21] still fail at approximating the diameter well.

Whereas in the main contribution of our work (see Section III), we completely neglect the diameter of the networks constructed, in Section VI, we propose some heuristics for constructing topic-connected networks with low average degree and constant diameter. In fact, the results in Section VI are a refinement of the preliminary results presented in [21].

A. *Our contributions*

In this work, we consider the problem of devising topic-based pub/sub overlay networks with low node degrees. One could argue for keeping the maximum degree of a node low or for keeping the overall average node degree low, since both are important and relevant measures of the complexity and scalability of a system [13], [18]. Unfortunately, previous attempts at minimizing either one of these degree measures alone [11], [21] resulted in a linear explosion for the other measure (see Table I).

In this work, we present the first algorithm that aims at keeping both the average and the maximum degree low. More specifically, we consider the following problem:

Low Degree Topic-Connected Overlay (Low-TCO) Problem: Given a collection of nodes V, a set of topics T, and the node interest assignment I, connect the nodes in V into a topic-connected overlay network G which has *both* low average and low maximum degree.

We present a parameterized sublinear approximation algorithm, Low Degree Overlay Design Algorithm(Low-ODA), for this problem which approximates both the average and the maximum degree well. More specifically, the Low-ODA algorithm achieves an average degree approximation of $O(\min\{k \cdot \log(n \cdot t), n\})$ and a maximum degree approximation of $O(\min\{(n/k) \cdot \log(n \cdot t), n\})$, where $n = |V|$ is the number of nodes in the network, $t = |T|$ is the number of topics and k is any parameter chosen from $[1, n]$ (See also Table I). To the best of our knowledge, this is the first overlay network design algorithm that achieves sublinear approximations on both the average and maximum degrees (e.g., for $k = \sqrt{n}$). The Low-ODA algorithm is a greedy algorithm which relies on repeatedly evaluating the trade-off of greedily adding an edge that would not increase the maximum degree versus greedily adding an edge that would lead to a small number of total edges in the final overlay network. The main contribution of this work is therefore to show that such a greedy approach can work and indeed leads to non-trivial sublinear approximation on both the average and maximum degree. We expect that the greedy parameterized template introduced by our algorithm will lead to applications in other network design domains where scalability is a key issue (see Section VII).

	Avg Degree	Max Degree
Chockler et. al. [11]	$O(\log(n \cdot t))$	$\theta(n)$
Onus and Richa [21]	$\theta(n)$	$O(\log(n \cdot t))$
This Paper	$O(k \cdot \log(n \cdot t))$	$O((n/k) \cdot \log(n \cdot t))$
Lower Bound	$\Omega(\log n)$	$\Omega(\log n)$

TABLE I

SUMMARY OF KNOWN RESULTS ON APPROXIMATION RATIOS OF MAXIMUM/AVERAGE DEGREE IN OVERLAY NETWORK CONSTRUCTION FOR PUBLISH/SUBSCRIBE COMMUNICATION (n: NUMBER OF NODES, t: NUMBER OF TOPICS, k IS ANY PARAMETER BETWEEN 1 AND n)

In addition, we present two algorithms (heuristics), Constant Diameter Overlay Design Algorithm I (CD-ODA-I) and Constant Diameter Overlay Design Algorithm II (CD-ODA-II), for building topic-based pub/sub networks where each topic-connected component is guaranteed to be of constant diameter — more specifically of diameter 2 —and where we aim at keeping the average degree low. Our experimental results show that our algorithms improve on the previous heuristic presented in [21] by a reduction of 20% on the average degree. As we mentioned earlier, keeping the node degrees and the network diameter low are key to the design of scalable topic-based pub/sub systems. We provide some preliminary results along these lines.

B. *Related Work*

Chockler et al. [11] introduced the Minimum Topic Connected Overlay (MinAv-TCO) problem [In the original paper, this problem was called Min-TCO.], which aims at minimizing the average degree alone of a topic-connected overlay network. They present an algorithm, called Greedy Merge *GM*, which achieves a logarithmic approximation on the minimum average degree of the overlay network. The GM algorithm follows the greedy approach described below:

The Greedy-Merge (GM) Algorithm [11]: The GM algorithm greedily adds the edge which maximally reduces the total number of topic-connected components at each step of the algorithm (initially we have the set of nodes V and no edges between the nodes).

While minimizing the average degree is a step forward towards improving the scalability and practicality of the pub/sub system, their algorithm may still produce overlay networks of very uneven node degrees where the maximum degree may be unnecessarily high. In [21], it is shown that GM algorithm may produce a network with maximum degree $|V|$ while a topic-connected overlay network of constant degree exists for the same configuration of I (See Table I).

In [21], the problem of minimizing the maximum degree of a topic-connected overlay network, Minimum Maximum Degree Topic Connected Overlay (MinMax-TCO), is considered, and a logarithmic approximation algorithm on the minimum maximum degree of the overlay network, Minimum Maximum Degree Overlay Design Algorithm (MinMax-ODA), is presented. The MinMax-ODA algorithm is also a greedy algorithm, as described below:

Min-Max Overlay Design Algorithm (MinMax-ODA) [21]: Initially there are no edges between the set of nodes V. At each step of the algorithm, add the edge which maximally reduces the total number of topic-connected components among the edges which increases maximum degree of the current graph minimally.

The MinMax-ODA algorithm may produce overlay networks of very high average degree: As we will show in Section II-A, this algorithm may produce a network with average degree $|V| - 2$ while a topic-connected overlay network of constant average degree exists for the same configuration of I (See Table I).

In this work, we are, to the best of our knowledge, the first to formally address the problem of minimizing both the maximum and the average degree in topic-connected pub/sub overlay network design. As we mentioned earlier, minimizing both maximum and average degree is important in many network domains, such as that of survivable network design [18] and that of establishing connectivity in wireless networks [13]. The overlay networks resulting from [2], [5], [10] are not required to be topic-connected. In [4], [9], [12], [28], topic-connected overlay networks are constructed, but they make no attempt to minimize the average or maximum node degree. The first papers to directly consider node degrees when building topic-connected pub/sub systems were [11] and [21], as we mentioned above. Minimizing the diameter in topic-connected pub/sub overlay network design first addressed in [21].

Some of the high level ideas and proof techniques of [11] and [21] have their roots in techniques used for the classical Set-Cover problem. We benefit from some of the ideas in [11], [21] and also build upon their constructions for Set-Cover, extending and modifying them to be able to handle the maximum degree and the average degree altogether.

C. Structure of the paper

In Section II, we present some definitions and restate the formal problem definition. In Section II-A, we present an outline of the related problem of minimizing the maximum node degree, namely the MinMax-TCO problem, and the corresponding logarithmic approximation algorithm MinMax-ODA proposed by Onus and Richa [21], since some of the ideas presented will be useful for the Low-TCO problem. Section III presents our topic-connected overlay design algorithm Low-ODA, whose approximation ratio is proved in Section IV. Section V presents our simulation results which validate the performance of the Low-ODA algorithm. Section VI presents our two new heuristics for the the problem of minimizing the average node degree while enforcing a 2-diameter overlay network. We conclude the paper, also presenting some future work, in Section VII.

II. Preliminaries

Let V be the set of nodes, and T be the set of topics. Let $n = |V|$. The interest function I is defined as $I : V \times T \rightarrow \{0, 1\}$. For a node $v \in V$ and topic $t \in T$, $I(v, t) = 1$ if and only if node v is subscribed to topic t, and $I(v, t) = 0$ otherwise.

For a set of nodes V, an overlay network $G(V, E)$ is an undirected graph on the node set V with edge set $E \subseteq V \times V$. For a topic $t \in T$, let $V_t = \{v \in V | I(v, t) = 1\}$. Given a topic $t \in T$ and an overlay network $G(V, E)$, the number of topic-connected components of G for topic t is equal to the number of connected components of the subgraph of G induced by V_t. An overlay network G is *topic-connected* if and only if it has one topic-connected component for each topic $t \in T$. The diameter of a graph is the length of the longest shortest path in the graph. The degree of a node v in an overlay network $G(V, E)$ is equal to the total number of edges adjacent to v in G.

A. Minimizing the maximum degree only

The MinMax-ODA algorithm (see Section I-B), proposed by Onus and Richa [21], addressed the MinMax-TCO problem, in which they aim at minimizing the maximum node degree. Unfortunately, while the MinMax-ODA algorithm produces a logarithmic approximation on the maximum node degree, it fails to approximate well the average node degree of a topic-connected overlay network: The approximation ratio on the average degree obtained by the MinMax-ODA algorithm may be as bad as $\Theta(n)$, as we show in the lemma below.

Lemma 1. *The MinMax-ODA algorithm can only guarantee an approximation ratio of $\Theta(n)$ on the average node degree, where n is number of nodes in the pub/sub system.*

Proof: Consider the example where we have n nodes $v_1, v_2, ..., v_n$, and n^2 topics $T = \{t_{i,j} | 1 \leq i, j \leq n\}$. Node v_1 is interested in all topics in T and each v_i is interested in $t_{i,j}$ and $t_{j,i}$, $2 \leq i \leq n$, $1 \leq j \leq n$. W.l.o.g., assume that n is even. The MinMax-ODA algorithm will produce an overlay network with $n(n - 2)/2$ edges, by repeatedly connecting a maximal matching of the nodes $v_1, \ldots, v_n$, $n-2$ times. The optimal overlay network with minimum number of edges is $E = \{(v_1, v_i) | 1 < i \leq n\}$, – the number of edges of this overlay network is $n-1$. Hence the approximation ratio of the MinMax-ODA algorithm can be as large as $(n(n-2)/2)/(n-1) = \Theta(n)$. ■

III. Low Degree Overlay Design Algorithm (Low-ODA)

In this section we present our overlay design algorithm (Low-ODA) for the Low-TCO problem. The weight of an edge $\{u, v\}$ is given by the reduction on the number of topic-connected components which would result from the addition of $\{u, v\}$ to the current overlay network. Let $1 \leq k \leq n$. Low-ODA starts with the overlay network $G(V, \emptyset)$. In each iteration of Low-ODA, the algorithm considers two edges:

e_1 : a maximum weight edge among the ones which minimally increases maximum degree of the current graph
e_2 : a maximum weight edge

If the weight of edge e_1 is greater than the weight of e_2 divided by k, edge e_1 is added to edge set of the overlay network; otherwise edge e_2 is added.

Let $NC(V, E)$ denote total number of topic connected components in the overlay network given by (V, E).

Algorithm 1 Low Degree Overlay Design Algorithm (Low-ODA)

```
1: OverlayEdges ← $\emptyset$
2: $V$ ← Set of all nodes
3: $G'(V, E')$ ← Complete graph on $V$
4: for $\{u, v\} \in E'$ do
5:     $w\{u, v\}$ ← Number of topics that both of nodes $u$ and $v$ have
6: end for
7: while $G(V$,OverlayEdges) is not topic-connected do
8:     Let $e_1$ be a maximum weight edge in $G'(V, E', w)$ among the ones which increase the maximum degree of $G(V$,OverlayEdges) minimally.
9:     Let $e_2$ be a maximum weight edge in $G'(V, E', w)$
10:    if $w(e_1) \geq w(e_2)/k$ then
11:        $e = e_1$
12:    else
13:        $e = e_2$
14:    end if
15:    OverlayEdges = OverlayEdges $\bigcup \{e\}$
16:    $E' \leftarrow E' - \{e\}$
17:    for $\{u, v\} \in E'$ do
18:        $w\{u, v\}$ ← NC($V$, OverlayEdges) - NC($V$,OverlayEdges $\bigcup \{u, v\}$ )
19:    end for
20: end while
```

Steps 1-6 of Low-ODA build an initial weighted graph $G'(V, E', w)$ on V, where $E' = V \times V$ and $w(\{u, v\})$ is equal to the amount of decrease in the number of topic-connected components resulting from the addition of the edge $\{u, v\}$ to the current overlay network (represented by the edges in OverlayEdges). Initially, this amount will be equal to the number of topics that nodes u and v have in common.

At each iteration of the while loop, two edges are considered: an edge (e_1) with maximum weight among the edges in E' that increase the maximum degree of the current graph minimally and an edge (e_2) with maximum weight in all of E'. If weight of the first one (e_1) is greater than or equal to weight of the second one (e_2) over k, e_1 is added to the set of overlay edges; otherwise e_2 is added. Note that the addition of an edge to OverlayEdges can either increase the maximum degree by 1 or not increase it at all. The crux in the analysis of this algorithm is to show that each of the edges will reduce the number of connected components by a "large" amount without increasing the maximum degree by too much.

While at a first glance the Low-ODA algorithm may seem like a trivial combination of the GM and MinMax-ODA algorithms, the analysis show that such a combination is far from trivial: Once we allow the algorithm (Low-ODA) to select some edges based solely on their weight (edges of type e_2), the "perfect matching" behavior of the edges selected by MinMax-ODA (basically one could show that the first $n/2$ edges selected by MinMax-ODA formed a perfect matching in G', as did the second set of $n/2$ edges, etc.) is no longer valid and the approximation ratio analysis used in MinMax-ODA (which heavily relied on this perfect matching decomposition of the edges selected) can no longer be directly used here.

Before we proceed in proving the approximation ratio on the maximum degree and the approximation ratio on the average degree guaranteed by Low-ODA, we prove that the algorithm terminates in $O(|V|^4|T|)$ time.

Lemma 2. *The Low-ODA algorithm terminates within $O(|V|^2)$ iterations of the while loop.*

Proof: At each iteration of the while loop, exactly one edge is added to the current overlay network. Hence the algorithm will terminate in at most $O(|V|^2)$ iterations. ■

Lemma 3. *The running time of Low-ODA is $O(|V|^4|T|)$.*

Proof: The weight initialization takes $O(|V|^2|T|)$ time. Updating the weight of each of the remaining edges takes $O(1)$ time([11], Lemma 6.4). Finding the edge with max weight will take at most $O(|V|^2)$ time. Since the total weight of the edges is $O(|V|^2|T|)$ at the beginning and greater than 0 at the end, Low-ODA takes $O(|V|^2|T|) \cdot O(|V|^2) = O(|V|^4|T|)$ time. ■

IV. Approximation Ratio

In this section, we will prove that our overlay design algorithm (Low-ODA) approximates the average degree by a factor $O(k \cdot \log(\sum_{v \in V} s_v))$ and the maximum degree by factor $O((n/k) \cdot \log(\sum_{v \in V} s_v))$, where $s_v = |\{t \in T | I(v, t) = 1\}|$.

As we mentioned in the previous section, the main challenge in the analysis is to overcome the fact that we can no longer think of the algorithm as selecting a sequence of perfect matchings of the nodes in V when bounding the approximation ratio on the maximum degree (the analysis of MinMax-ODA algorithm heavily relied on this "perfect matching behavior").

Now we present some definitions which will be useful for the proofs of both Theorems 1 and 2. Recall that we use s_v to denote $|\{t \in T | I(v,t) = 1\}|$. At the beginning of the algorithm, the total number of connected components is $C_{start} = \sum_{v \in V} s_v$ and at the end $C_{end} = |\{t | t \in T$ and $\exists v \in V$ such that $I(v,t) = 1\}|$. Note that since we count the connected components for each topic separately, once we get down to C_{end} components, there must exist *exactly one* component for each active topic t (i.e., each t such that there exists some v with $I(v,t) = 1$) — i.e., the overlay network is topic-connected.

Theorem 1. *The overlay network output by Low-ODA has average node degree within a factor of $O(k \cdot \log(\sum_{v \in V} s_v))$ from the minimum possible average node degree for any topic-connected overlay network on V.*

Proof: The proof follows the general lines as the proof of the logarithmic approximation ratio for the classic set cover problem (which was also the basis for the approximation ratio proof of the GM algorithm for the MinAv-TCO problem [11]). Assume we have an instance of the Low-TCO problem and that $G(V, E_{opt})$ is a solution for this instance with minimum number of edges. Let $|E_{opt}| = m$.

Let e_i be the i^{th} edge added to the set by the algorithm Low-ODA. Let n_i be total number of connected components before we add the i^{th} edge, so $n_1 = C_{start}$. Let $S_i = \{e_1, e_2, ..., e_{i-1}\}$ be the set of all edges found before the algorithm starts adding the i-th edge. Before Low-ODA starts adding the i^{th} edge, we have n_i components and we know that if we add all the edges $E_{opt} - S_i$, to the current solution, the total number of connected components will be reduced to C_{end}. Since $|E_{opt} - S_i| \le m$, there exists an edge which decreases the total number of connected components by at least $(n_i - C_{end})/m$. Since our algorithm always adds at least a $(1/k)$-optimal edge, the edge e_i that our algorithm adds must decrease the total number of connected components at that time by at least $(1/k)$ of this amount. Therefore,

$$n_i - n_{i+1} \ge (n_i - C_{end})/(m \cdot k)$$
$$\Rightarrow \quad n_{i+1} - C_{end} \le (1 - 1/(m \cdot k))(n_i - C_{end}).$$

Hence, the number of iterations of our algorithm Low-ODA is less than or equal to the smallest z which satisfies

$$1 > (n_1 - C_{end})(1 - 1/(m \cdot k))^z$$
$$\Rightarrow \quad z \le m \cdot k \ln(C_{start} - C_{end})$$
$$\Rightarrow \quad z \le m \cdot k \ln(C_{start}).$$

■

Theorem 2. *The overlay network output by Low-ODA has maximum node degree within a factor of $O((n/k) \cdot \log(\sum_{v \in V} s_v))$ from the minimum possible maximum node degree for any topic-connected overlay network on V.*

Proof: As with the proof of Theorem 1, the proof follows the general lines of the proof of the logarithmic approximation ratio for the classic set cover problem.

However, before we can apply the set cover framework, we first need to carefully relate the sequence of edges selected by the Low-ODA algorithm to a sequence of optimal matchings which reduce the current number of connected components maximally. Note that we no longer can break the sequence of edges selected by our algorithm into a sequence of perfect matchings, as in the MinMax-ODA algorithm.

Assume we have an instance of the Low-TCO problem and that $G(V, E_{opt})$ is a solution with minimum possible maximum degree for this instance. Let this maximum degree be d_{opt}. We will use the following well-known result in graph theory for the proof.

Lemma 4 ((Lemma 4 in [21])). *Given a graph $G(V,E)$ with maximum degree d, we can partition the edge set E into $d+1$ matchings M_i, $1 \le i \le (d+1)$.*

Using the lemma above, we can partition the edge set E_{opt} of the optimum solution into $d_{opt} + 1$ matchings M_i, $1 \le i \le (d_{opt} + 1)$.

At the start, all nodes have degree zero. At each iteration of the while loop, a maximum weight edge among the ones that increase the maximum degree of the current graph minimally or a maximum weight edge is added to the set of overlay edges.

After a number of iterations, the weight of a maximum weight edge among the ones that increase the maximum degree of the current graph minimally will be less than the weight of a maximum weight edge over k and we will add this maximum weight edge to the graph — this edge will increase the maximum degree of the graph by 1.

Let S_i be the edge set containing all edges added by Low-ODA from the time an edge e' increased the maximum degree of G' from $i-1$ to i until the last time an edge is added to G' without increasing its maximum degree further (i.e., without increasing the maximum degree to $i+1$). Let $h = n/(2k) + 1$. Let $R_i = S_{h\cdot(i-1)+1} \bigcup S_{h\cdot(i-1)+2} \bigcup ... \bigcup S_{h\cdot(i-1)+h}$ — i.e., R_i denotes the set of all edges added by Low-ODA while the resulting maximum degree was between $h(i-1)+1$ and hi. Let $RA_i = R_1 \bigcup R_2 \bigcup ... \bigcup R_{i-1}$ be the union of all edges added before the algorithm starts adding the set R_i. Let n_i be the total number of connected components before the algorithm adds R_i, so $n_1 = C_{start}$.

The following lemma proves that each set R_i chosen by our algorithm decreases the current total number of connected components by at least $1/3$ of the number of current components connected by any optimal matching, where an optimal matching is one that reduces the current number of connected components maximally among all possible maximal matchings in G'. Note that a matching increases the maximum degree of G' by at most 1.

Lemma 5. *Joining the set R_i to the edges of $G(V, RA_i)$ reduces the total number of connected components of $G(V, RA_i)$ by at least $1/3$ of the number of current components connected by any optimal matching.*

Proof: Let P be the edge set of the matching that reduces the total number of connected components of the $G(V, RA_i)$ by the maximum amount, which we denote by c. Let $Q = \{e_1, e_2, \dots, e_j\}$ be the edge set of R_i. Let $e_l = u_l v_l$ for $1 \leq l \leq j$. For e_a and e_b, if $a < b$, then e_a is found before e_b by our algorithm. Let Q reduce the total number of connected components of $G(V, RA_i)$ by c'. Let $G_0 = G(V, RA_i)$ and $G_l = G_{l-1} \bigcup \{e_l\}$, for $1 \leq l \leq j$. Let e_l reduce the total number of connected components of G_{l-1} by y_l. Then,

$$c' = \textstyle\sum_{1 \leq l \leq j} y_l \qquad (1)$$

Consider the case where e_l does not increase the maximum degree of the current graph, or $l = 1$, or e_l increases the maximum degree of current graph and there is no possible edge which does not increase the maximum degree of the current graph, $1 \leq l \leq j$. In this scenario, we let X_l be the set of edges in P which are incident to u_l or v_l, $1 \leq l \leq j$, and not in $X_{l'}$, $1 \leq l' \leq l-1$. Thus, X_l will have zero, one or two edges for $1 \leq l \leq j$.

Now consider the case where e_l increases the maximum degree of the current graph and there are some edges that do not increase the maximum degree of the current graph, $2 \leq l \leq j$. In this case, we define X_l to be the set of the first k maximum weight edges in P which are not in $X_{l'}$, $1 \leq l' \leq l-1$. If there are less than k elements in P which are not in $X_{l'}$, $1 \leq l' \leq l-1$, X_l will only have these edges. If there are edges which are incident to u_l or v_l and not in $X_{l'}$, $1 \leq l' \leq l$, then replace any edges from X_l with those edges (note that there may be at most two edges of this kind).

Let $P_0 = P$ and $P_l = P_{l-1} - X_l$ for $1 \leq l \leq j$. Let X_l reduce the total number of connected components of G_{l-1} by x_l for $1 \leq l \leq j$. Let P_l reduce the total number of connected components of G_l by c_l for $0 \leq l \leq j$.

If there is an edge e_l that increases the maximum degree of current graph and there is no possible edge which does not increase maximum degree of the current graph, $2 \leq l \leq j$, then for each vertex of the graph, there is at least one edge $e_{l'}$ incident to this vertex, $1 \leq l' \leq l-1$. So, union of sets $X_{l'}$, $1 \leq l' \leq l-1$, contains all the edges in P. Thus, $P_j = \emptyset$. Now, consider the case when there is no edge which satisfies these properties (so, when algorithm chooses an edge e_l which increases the maximum degree, there is always an edge which does not increase maximum degree of the current graph). Since R_i contains h sets of $S_{i'}$, there are $h - 1 = n/(2k)$ edges that increase the maximum degree of the current graph. Thus there are at least $n/(2k)$ of the sets X_l with k edges each — we call them $X'_1, \dots, X'_{n/(2k)}$ — (if X_l has less than k edges than all edges of set P are already in one of sets $X_{l'}$, $1 \leq l' \leq l-1$, and hence $P_j = \emptyset$). The union of sets X'_l has at least $(n/2k) \cdot k = n/2$ edges. On the other hand, since P is a matching, this union can have at most $n/2$ edges. Hence, $P_j = \emptyset$. Hence,

$$c_0 = c, \; c_j = 0 \qquad (2)$$

Consider the case where e_l does not increase the maximum degree of current graph or $l = 1$ or e_l increases the maximum degree of current graph and there is no possible edge which does not increase maximum degree of the current graph, $1 \leq l \leq j$. If X_l has two edges, then our algorithm did not choose one of these two edges at that step and choose e_l instead, $0 \leq l \leq j$. Since our algorithm greedily chooses the edges, e_l reduces the total number of connected components of G_{l-1} by at least as much as each of the edges in X_l. Hence, $y_l \geq x_l/2$. Similarly, if X_l has one or zero edges, then $y_l \geq x_l$.

Now consider the case where e_l increases the maximum degree of current graph and there are some edges which does not increase maximum degree of the current graph, $2 \leq l \leq j$. X_l has at most k edges. Our algorithm did not choose one of these k edges at that step and choose e_l instead, $0 \leq l \leq j$. Since our algorithm greedily chooses the edges, e_l reduces the total number of connected components of G_{l-1} by at least as much as k times any of the edges in X_l. Since X_l has at most k edges, $y_l \geq x_l$. So,

$$\begin{array}{l} y_l \geq \frac{x_l}{2},\ 1 \leq l \leq j \\ \Rightarrow \qquad \sum_{1 \leq l \leq j} y_l \geq \frac{1}{2} \sum_{1 \leq l \leq j} x_l \qquad (3) \end{array}$$

Since $P_{l+1} = P_l - X_{l+1}$ and $G_{l+1} = G_l \bigcup \{e_{l+1}\}$, $0 \leq l \leq j-1$, the amount that P_l reduces the total number of connected components of G_l is smaller than sum of the amount that P_{l+1} reduces the total number of connected components of G_{l+1} and the amount that e_{l+1} reduces the total number of connected components of G_l and the amount X_{l+1} reduces the total number of connected components of G_l. Hence,

$$c_{l+1} \geq c_l - (x_{l+1} + y_{l+1}) \text{ for } 0 \leq l \leq j-1 \qquad (4)$$

If we add all the inequalities (2) and (4), we will have

$$\textstyle\sum_{1 \leq l \leq j} x_l + \sum_{1 \leq l \leq j} y_l \geq c \qquad (5)$$

From the inequalities (3) and (5), we will have

$$3 \textstyle\sum_{1 \leq l \leq j} y_l \geq c \qquad (6)$$

From the inequalities (1) and (6), we will have

$$c' \geq c/3$$

■

Before Low-ODA starts adding the set R_i, we have n_i components and we know that if we add all the $(d_{opt} + 1)$ matchings $M_j - RA_i$, $1 \leq j \leq (d_{opt} + 1)$, to the current solution, the total number of connected components will be reduced to C_{end}. Therefore, there exists a matching $M_j - RA_i$ which decreases the total number of connected components by at least $(n_i - C_{end})/(d_{opt} + 1)$. Our algorithm always finds the set R_i that reduces the total number of connected components of $G(V, RA_i)$ by at least $1/3$ of any optimal matching which reduces the current number of connected components by the maximum amount (Lemma 5). Hence, the set R_i that our algorithm uses must decrease the total number of connected components at that time by at least $(1/3)$ of the optimal amount. Therefore,

$$\begin{aligned} & n_i - n_{i+1} \geq (n_i - C_{end})/(3(d_{opt}+1)) \\ \Rightarrow \quad & n_{i+1} - C_{end} \leq (1 - 1/(3(d_{opt}+1)))(n_i - C_{end}). \end{aligned}$$

Hence, the number of iterations for our algorithm Low-ODA is less than or equal to the smallest m which satisfies

$$\begin{aligned} & 1 > (n_1 - C_{end})(1 - 1/(3(d_{opt}+1)))^m \\ \Rightarrow \quad & m \leq 3(d_{opt}+1)\ln(C_{start} - C_{end}) \\ \Rightarrow \quad & m \leq 3(d_{opt}+1)\ln(C_{start}) \end{aligned}$$

Since $h = n/(2k)+1$, the maximum degree of resulting graph is less than or equal to $(n/(2k)+1) \cdot 3(d_{opt}+1) \cdot \ln(C_{start})$. ■

V. Experimental Results

The GM algorithm [12], the MinMax-ODA algorithm [21] and the Low-ODA algorithm are implemented in Java. These three algorithms are compared according to maximum degree and average degree in the resulting overlay graphs. Our experimental results show that the Low-ODA algorithm has better maximum degree than the GM algorithm and has better average degree than the MinMax-ODA algorithm, at the expense of a small degradation of the other corresponding degree parameter (recall that the GM algorithm has proven approximation bounds on the average degree only and the MinMax-ODA on the maximum degree only).

A. Maximum Node Degree

For these experiments, the number of nodes varies between 100 and 1000, and the number of topics is fixed at 100. Each experiment is done 1000 times. We also fix number of subscriptions to $s = 10$. For the Low-ODA algorithm, the parameter k is chosen to be equal to 3 (k is chosen as a number close to $\sqrt{8}$, since the Low-ODA algorithm behaves pretty much like the MinMax-ODA algorithm when $k = 8$ – see results in Section V-C). Each node subscribes to each topic t_i with probability p_i. The value of p_i is distributed according to a Zipf distribution ($\alpha = 0.5$). This experimental setting is similar to previous studies [12].

Figure 1 presents a comparison of the GM, MinMax-ODA and Low-ODA algorithms according to the maximum degree. When we compare the results of the algorithms, Low-ODA takes values in between the GM and the MinMax-ODA algorithms.

B. Average Node Degree

The experimental setting is the same as in the previous subsection. Figure 2 is a comparison of the GM, MinMax-ODA and Low-ODA algorithms according to the average degree. When we compare the results of the algorithms, Low-ODA takes values in between the GM and the MinMax-ODA algorithms'.

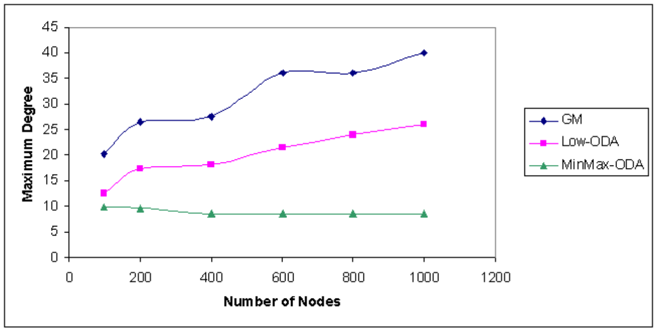

Fig. 1. Maximum node degree for GM, MinMax-ODA and Low-ODA

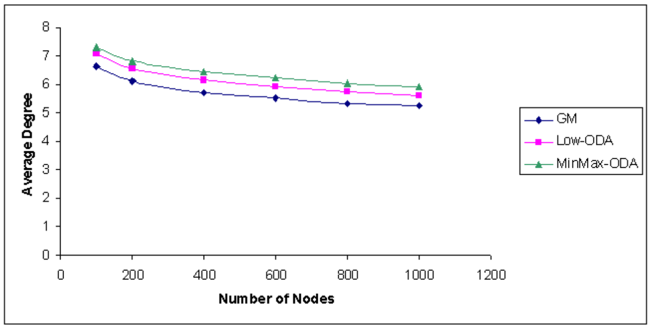

Fig. 2. Average node degree for GM, MinMax-ODA and Low-ODA

C. Different Parameters

The experimental setting is similar to previous subsections. The number of nodes is 100. The parameter k of the Low-ODA algorithm varies between 1 and 8. When $k = 1$, the Low-ODA algorithm behaves basically in the same way as the GM algorithm, and when $k = 8$, the Low-ODA algorithm behaves basically in the same way as the MinMax-ODA algorithm. As k increases, the maximum degree decreases and the average degree increases (Figure 3 and Figure 4).

VI. Constant Diameter Overlays for Publish-Subscribe

In this section, we study the following optimization problem, intially proposed in [21]:

Constant Diameter Topic-Connected Overlay (CD-TCO) Problem: [21] Given a collection of nodes V, a set of topics T, and a node–interest assignment I, connect the nodes in V into a topic-connected overlay network G which has least possible average degree and constant diameter.

We present two new overlay network construction heuristics that guarantee constant diameter and topic-connectivity, which are most important factors for efficient routing. Our heuristic also aims at keeping the average node degree low.

In [21], a heuristic (CD-ODA) is presented for this problem. CD-ODA starts with the overlay network $G(V, \emptyset)$. At each

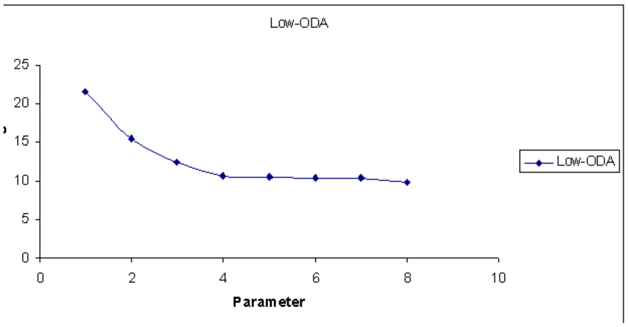

Fig. 3. Maximum node degree for different parameters

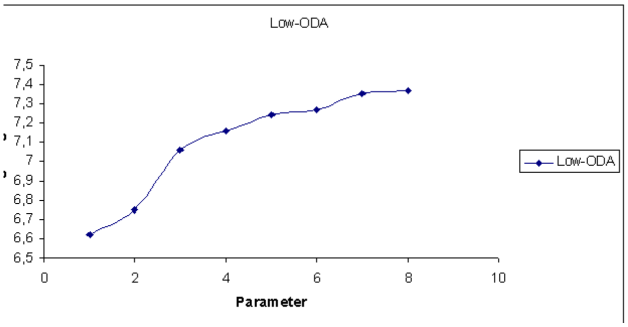

Fig. 4. Average node degree for different parameters

iteration of CD-ODA, a node which has maximum number neighbors with non-empty interest intersection is chosen. Number of neighbors is equal to $n_u = |\{v \in V | \exists t \in T, Int(v,t) = Int(u,t) = 1\}|$. After that, an edge between this node and each of its neighbors is added and the topics in this node's interest assignment are removed from the set of topics.

We present two new heuristics for this problem when the diameter is restricted to be equal to 2, validating our heuristics via experimental results. Our experimental results show that our heuristics improves CD-ODA [21] by a factor of 20%.

A. Constant Diameter Overlay Design Algorithm I(CD-ODA-I)

The first heuristic presented, the *Constant Diameter Overlay Design Algorithm I (CD-ODA-I)*, starts with the overlay network $G(V, \emptyset)$. At each iteration of CD-ODA-I, a node u which has maximum weight neighbors is chosen. The neighbor weight of a node z is equal to

$$w_z = \sum_{t \in T} |\{v \in V | Int(v,t) = Int(z,t) = 1\}|.$$

We add an edge between u and each of its neighbors and then remove the topics in this node's interest assignment from the set of topics.

B. Constant Diameter Overlay Design Algorithm II(CD-ODA-II)

The second heuristic presented, *Constant Diameter Overlay Design Algorithm II (CD-ODA-II)*, also starts with the overlay network $G(V, \emptyset)$. At each iteration of CD-ODA-II, a node u which has maximum connection density, d_u, is chosen. The connection density of a node u, d_u, is given by

$$d_u = \frac{\sum_{t \in T} |\{v \in V | Int(v,t) = Int(u,t) = 1\}|}{|\{v \in V | \exists t \in T, Int(v,t) = Int(u,t) = 1\}|}.$$

Note that $d_u = \frac{w_u}{n_u}$. We add an edge between a node u with maximum density and each of its neighbors and then remove the topics in this node's interest assignment from the set of topics.

C. Analysis of Algorithms

Lemma 6. *Both CD-ODA-I and CD-ODA-II algorithms terminate within $O(|V|^2 \cdot |T|)$ time.*

Lemma 7. *Both CD-ODA-I and CD-ODA-II algorithms generate a 2-diameter overlay for each topic.*

Proof: Since the algorithms generate a star for each topic, each topic overlay network will have diameter 2. ■

D. Experimental Results

The GM algorithm [11], the CD-ODA algorithm [21] and the two heuristics presented above are implemented in Java. These algorithms are compared according to the average degree in the resulting graph. Experiments are done 1000 times. The diameter is always 2 for our algorithms and for the CD-ODA algorithm of [21], and it may be $\theta(n)$ for the GM algorithm. When we compare the results of GM, CD-ODA, CD-ODA-I, and CD-ODA-II according to the average degree, CD-ODA require at most 2.3 times more edges than GM, and CD-ODA-II requires at most 1.8 times more edges than the GM algorithm. The CD-ODA-II algorithm improves CD-ODA [21] by a factor of 20%.

1) Average Node Degree with Varying Subscription Size: The number of nodes and the number of topics are fixed to 100. The subscription size varies between 10 and 50. Each node is interested in each topic uniformly at random. This experimental setting is similar to previous studies [11], [24].

Figure 5 is a comparison of GM, CD-ODA and our algorithms according to the average degree. The average degree of the overlay network decreases for both GM, CD-ODA and our algorithms as the subscription size increases, since the algorithms can find edges with higher correlations. When we compare the results of GM and our algorithm CD-ODA-II, our algorithm requires at most 1.8 times more edges than GM (Figure 5). CD-ODA-II improves CD-ODA by factor 20% on average and CD-ODA-I by factor 1% on average.

VII. Conclusions

In this paper, we study a new optimization problem (Low-TCO) that constructs a practical and scalable overlay network for publish/subscribe communication with many topics. We present a topic-connected overlay network design algorithm

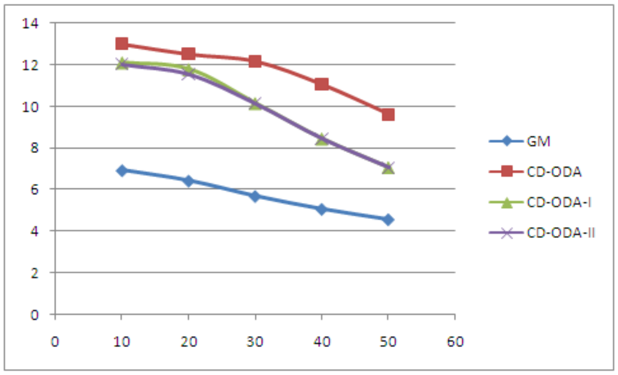

Fig. 5. Average node degree for GM, CD-ODA, CD-ODA-I and CD-ODA-II

(Low-ODA) which approximates both average and maximum degree well. We anticipate that the parameterized algorithmic framework proposed by Low-ODA will be applicable in other network design domains where, for scalability, it is important to keep both the maximum degree as well as the average degree of an overlay network low. Examples of such application domains are in the design of survivable networks [18] and in wireless networks [13].

As future work, we would like to build upon our CD-ODA-I and CD-ODA-II algorithms, by formally and experimentally evaluating the hardness of obtaining a topic-connected overlay design algorithm which achieves a “good” trade-off between low diameter and low node degree. This basically amounts to a bicriteria optimization problem and we have to be able to “quantify” the relative importance of optimizing over these two parameters (e.g., in the CD-ODA-I algorithm and the CD-ODA-II algorithm we restrict our attention to networks of diameter 2, while aiming at maintaining the average degree low). Two other important lines for future work would be to design efficient *distributed* algorithms for the Low-TCO problem, and to look at this problem under the line of a dynamic configuration of the node set V and the interest assignment I.

References

[1] *Oracle9i Application Developers Guide Advanced Queuing*, Oracle, Redwood Shores, CA.

[2] E. Anceaume, M. Gradinariu, A. K. Datta, G. Simon, and A. Virgillito, *A semantic overlay for self- peer-to-peer publish/subscribe*, In ICDCS, 2006.

[3] S. Baehni, P. T. Eugster, and E. Guerraoui, *Data-aware multicast.* In DSN, 2004.

[4] R. Baldoni, R. Beraldi, V. Quema, L. Querzoni, and S. T. Piergiovanni, *TERA: Topic-based Event Routing for Peer-to-Peer Architectures*, 1st International Conference on Distributed Event-Based Systems (DEBS). ACM, 6 2007.

[5] R. Baldoni, R. Beraldi, L. Querzoni, and A. Virgillito, *Efficient publish/subscribe through a self-organizing broker overlay and its application to SIENA*, The Computer Journal, 2007.

[6] S. Banerjee, B. Bhattacharjee, and C. Kommareddy, *Scalable application layer multicast*, SIGCOMM Comput. Commun. Rev, 32(4):205-217, 2002.

[7] S. Bhola, R. Strom, S. Bagchi, Y. Zhao, and J. Auerbach, *Exactly-once delivery in a content-based publish-subscribe system.* In DSN, 2002.

[8] A. Carzaniga, M. J. Rutherford, and A. L. Wolf, *A routing scheme for content-based networking.*, IEEE INFOCOM 2004, Hon Kong, China, Mar. 2004.

[9] M. Castro, P. Druschel, A. M. Kermarrec, and A. Rowstron, *SCRIBE: a large-scale and decentralized application-level multicast infrastructure*, IEEE J. Selected Areas in Comm. (JSAC), 20(8):14891499, 2002.

[10] R. Chand and P. Felber, *Semantic peer-to-peer overlays for publish/subscribe networks*, In Euro-Par 2005 Parallel Processing, Lecture Notes in Computer Science, volume 3648, pages 11941204. Springer Verlag, 2005.

[11] G. Chockler, R. Melamed, Y. Tock and R. Vitenberg, *Constructing scalable overlays for pub-sub with many topics*, Proc. of the 26th ACM Symp. on Principles of Distributed Computing (PODC), 2007, pp. 109–118.

[12] G. Chockler, R. Melamed, Y. Tock, and R. Vitenberg, *SpiderCast: A Scalable Interest-Aware Overlay for Topic-Based Pub/Sub Communication*, 1st International Conference on Distributed Event-Based Systems (DEBS). ACM, 6 2007.

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

[14] R. Diestel, *Graph Theory*, Springer-Verlag, 2nd edition, New York, 2000.

[15] P. T. Eugster, P. A. Felber, R. Guerraoui, and A. M. Kermarrec. *The many faces of publish/subscribe*. ACM Computing Surveys, 35(2):114131, 2003.

[16] R. Guerraoui, S. Handurukande, and A. M. Kermarrec, *Gossip: a gossip-based structured overlay network for efficient content-based filtering*, Technical Report IC/2004/95, EPFL, Lausanne, 2004.

[17] B. Korte, J. Vygen, *Combinatorial Optimization Theory and Algorithms*, Springer-Verlag, 2nd edition, 2000.

[18] L.C. Lau, J. Naor,M.R. Salavatipour, and M. Singh. *Survivable Network Design with Degree or Order Constraints*. In *Procedings of ACM STOC'07.*

[19] R. Levis, *Advanced Massaging Applications with MSMQ and MQSeries.* QUE, 1999.

[20] H. Liu, V. Ramasubramanian, and E. G. Sirer. *Client behavior and feed characteristics of rss, a publish-subscribe system for web micronews*. In Internet Measurement Conference (IMC), Berkeley, California, October 2005.

[21] M. Onus, A. W. Richa, *Minimum Maximum Degree Publish-Subscribe Overlay Network Design*, 28th Annual IEEE Conference on Computer Communications (INFOCOM), Rio de Janeiro, Brazil, April 2009.

[22] M. Onus, A. W. Richa, *Publish-Subscribe Overlay Network Design*, Technical Report, Arizona State University, Department of Computer Science and Engineering, TR-09-005, 2009, available at http://sci.asu.edu/news/technical/index.php.

[23] M. Onus, A. W. Richa, *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, August 2009.

[24] V. Ramasubramanian, R. Peterson, and E. G. Sirer. *Corona: A High Performance publish-subscribe system for the world wide web.*, In NDSI, 2006.

[25] D. Sandler, A. Mislove, A. Post and P. Druschel. *FeedTree: Sharing web micronews with peer-to-peer event notification.*, In International Workshop on Peer-to-Peer Systems(IPTPS), 2005.

[26] D. Tam, R. Azimi, and H.-A Jacobsen. *Building content-based publish/subscribe systems with distributed hash tables.*, In 1st International Workshop on Databases, Information Systems, and P2P Computing(DBISP2P), Berlin, Germany, 2003.

[27] Y. Tock, N. Naaman, A. Harpaz and G. Gershindky. *Hierarchical clustering of message flows in a multicast data dissemination system.*, In 17th IASTED International Conference Parallel and Distributed Computing and Systems, pages 320-327, 2005.

[28] S. Voulgaris, E. Riviere, A. M. Kermarrec, and M. van Steen, *Sub-2-sub: Self-organizing content-based publish subscribe for dynamic large scale collaborative networks*, In IPTPS, 2006.