Constructing Scalable Overlays for Pub-Sub with Many Topics

Problems, Algorithms, and Evaluation

Gregory Chockler Roie Melamed Yoav Tock
IBM Haifa Research Laboratory
{chockler,roiem,tock}@il.ibm.com

Roman Vitenberg
Department of Informatics,
University of Oslo, Norway
romanvi@ifi.uio.no

ABSTRACT

We investigate the problem of designing a scalable overlay network to support decentralized topic-based pub/sub communication. We introduce a new optimization problem, called *Minimum Topic-Connected Overlay (Min-TCO)*, that captures the tradeoff between the scalability of the overlay (in terms of the nodes' fanout) and the message forwarding overhead incurred by the communicating parties. Roughly, the Min-TCO problem is as follows: Given a collection of nodes and their subscriptions, connect the nodes using the minimum possible number of edges so that for each topic t, a message published on t could reach all the nodes interested in t by being forwarded by only the nodes interested in t.

We show that the decision version of Min-TCO is NP-complete, and present a polynomial algorithm that approximates the optimal solution within a logarithmic factor with respect to the number of edges in the constructed overlay. We further prove that this approximation ratio is almost tight by showing that no polynomial algorithm can approximate Min-TCO within a constant factor (unless P=NP). We show experimentally that on typical inputs, the fanout of the overlay constructed by our approximation algorithm is significantly lower than that of the overlays built by the existing algorithms, and that its running time is just a small fraction of the analytical worst case bound. As Min-TCO can be shown to capture several important aspects of most known overlay-based pub/sub implementations, our study sheds light on the inherent limitations of the existing systems and provides an insight into the best possible feasible solution.

Finally, we introduce a flexible framework that generalizes Min-TCO and formalizes most similar overlay design problems that occur in scalable pub/sub systems. We also briefly discuss several examples of such problems, and show some results with respect to their complexity.

PODC'07, August 12–15, 2007, Portland, Oregon, USA.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Network Topology*; G.2.2 [**Mathematics of Computing**]: Graph Theory—*Network problems*

General Terms

Algorithms, Theory, Experimentation

Keywords

pub/sub, overlay networks, peer-to-peer, application-level multicast, optimization problems

1. INTRODUCTION

Publish/subscribe (pub/sub) [13] is a popular communication paradigm allowing the users to interact in a decoupled fashion by publishing their messages on logical channels and receiving the messages off the channels to which they are subscribed. The pub/sub systems are classified as either *topic-based* or *content-based*. In topic-based pub/sub, the channels are specified through unique identifiers, called *topics*; whereas in content-based pub/sub, the channels are specified through a collection of attributes that could be arbitrary data types. Due to its simple interface and inherent scalability, pub/sub is commonly used to support many-to-many communication in a wide variety of popular Internet applications, such as stock-market monitoring engines [20], RSS feeds [16], on-line gaming, and many others. There are also numerous implementations of the pub/sub middleware in both industry [1, 7, 15] and academia [2, 3, 4, 6, 8, 9, 10, 14, 17, 18, 19, 20, 21, 22].

In this paper, we focus on *fully decentralized* implementations of the topic-based pub/sub systems (see e.g., [4, 10, 17, 18, 21]) where the parties do not rely on intermediate agents of any kind (such as servers or message brokers) to forward their messages but rather communicate in a peer-to-peer fashion, effectively forming an *application-level* or an *overlay* network. In this environment, an efficient publication routing protocol becomes a major factor affecting the pub/sub performance. We can thus judge the quality of a constructed overlay in terms of the complexity of the *best* possible routing scheme that can be implemented on top of it. Intuitively, this complexity is minimized if all the nodes interested in the same topic t can be organized into a

dissemination tree which both (1) consists of only the nodes interested in t, and (2) has a low diameter. In this paper, we only deal with (1), and leave (2) for future work (see Section 9).

Let G be an overlay network. The necessary pre-requisite enabling (1) is the following property of the overlay, which we call *topic-connectivity*: for each topic t, the sub-graph of G induced by the nodes interested in t is connected. Note that while it is straightforward to achieve topic-connectivity by creating a separate overlay structure (such as a tree or ring) for each topic in the system, this will result in an overlay where the number of connections maintained by each node grows linearly with the node's subscription size (being on average roughly twice as big as that). There are however, several significant costs to edges in practical overlay systems.

First, maintaining the overlay topology requires each node to continuously monitor the availability of each of its neighbors thus incurring the cost of heartbeats and keep-alive state for each connection. Furthermore, if sequential-diff-based compression scheme is used (e.g., for efficient transmission of bulk data), there is an extra cost associated with a recent history table for each link. Next, for efficiency reasons, messages posted on multiple topics are typically aggregated into a single compound message thus amortizing the header cost, which otherwise, can be quite high for small messages. Since each message forwarded over a particular link requires a separate header, the more links there are, the fewer topics can be routed over each individual links, thus diminishing the cross-topic aggregation benefits. In addition, if the edges are maintained using connection-oriented transport, such as TCP, there is the cost of connection state for each edge. Hence, it is desirable to reduce the number of connections maintained by each node to the extent possible, and in particular, try to improve upon the upper bound of twice the node subscription size.

Intuitively, this should be possible if the individual node subscriptions are well-correlated since in this case, one can improve connectivity of several topics *at once* by choosing to connect a pair of nodes which are both interested in these topics. Indeed, several recent empirical studies suggest that correlated workloads are common in practice, and in particular, occur in such popular pub/sub applications as RSS [16] and stock-market monitoring engines [20]. It is therefore interesting to inquire whether it is possible to devise a practical algorithm to exploit subscription correlation for constructing a scalable overlay for pub/sub with provable performance guarantees.

To study this question formally, we introduce a new optimization problem, called *Minimum Topic-Connected Overlay (Min-TCO)*, which is roughly as follows: given a collection of nodes V, a set of topics T, and the node interest assignment, connect the nodes in V into an overlay network G using the least possible number of edges such that G is topic-connected. We show hardness of Min-TCO by proving that the decision version of Min-TCO (called *TCO*) is NP-complete. We show this in two steps: First, we define a single node version of TCO where the degree constraint is only placed on a single given node (and not on all nodes as in TCO), and show that Set Cover is polynomially reducible to single node TCO. Since the latter can be shown to be polynomially reducible to TCO, NP-hardness of Set Cover implies the same for TCO.

On the positive side, we show that it is possible to efficiently approximate Min-TCO using a simple greedy algorithm, called *Greedy Merge (GM)*, that proceeds by adding edges to the overlay until the overlay is topic-connected while ensuring that each newly added edge merges the previously disjoint connected components for the largest number of topics. Note that a naive implementation of this scheme leads to runtime complexity as big as $O(|V|^4|T|)$. In Section 6, we utilize dynamic programming ideas to derive a much more efficient version of GM that has a running time of just $O(|V|^2|T|)$. We further show that the number of links created by GM is within a logarithmic factor of that created by the optimal solution. We prove that in most practical cases, this approximation ratio is almost tight by showing that no polynomial algorithm can approximate Min-TCO within a constant factor (unless P=NP). Our study thus demonstrates that subscription correlation can indeed be efficiently leveraged for constructing topic-connected (and thus optimal with respect to the routing cost) overlay topology with provable and (almost) tight scalability characteristics.

Note that although the Min-TCO and Set Cover problems appear to be related, they are nevertheless, fundamentally different due to an additional topological constraint (topic-connectivity) imposed by Min-TCO and not found in Set Cover. As a result, we cannot re-use the known approximability results for Set Cover directly for proving similar bounds for Min-TCO. Nevertheless, we benefit from the ideas used to obtain those results to derive similar bounds for Min-TCO.

We show experimentally (see Section 8) that exploiting interest correlation indeed results in a substantial scalability improvement. To this end, we compare the average degrees of the overlays created by GM and by a protocol in which topic-connectivity is achieved by maintaining a separate logical ring, ordered by the node identifiers, for each topic. We show that under subscription patterns derived from several common well-correlated topic popularity distributions, such as exponential and Zipf, using GM results in a significant reduction in the average overlay degree (almost 3-fold when we increase the number of nodes and upto 8-fold when we increase the number of topics). In addition, our measurements indicate that, under the above workloads, GM's running time is just a small fraction (in the order of 10^{-3}) of the worst case analytical bound ($O(|V|^2|T|)$) thus further attesting to the practicality of GM in common settings.

Finally, although topic-connectivity is essential for minimizing the routing cost, it might not always be achievable in practice (e.g., because of hard limits imposed on the node degrees due to bandwidth limitations). It is therefore interesting to explore the tradeoff between scalability and routing complexity in the case, we are ready to compromise topic-connectivity to some extent. To give a precise sense to this tradeoff we introduce a flexible framework that generalizes Min-TCO and allows for defining new problems that formally capture a multitude of constraints (such as, the node degrees, transmission rates on different topics, bandwidth and computing power available to each node, etc.), and optimization objectives, such as per-topic latency of the message dissemination, per-topic filtering overhead, etc. We illustrate our framework by giving examples of several representative problems, and mention some results with respect to their computational complexity.

2. RELATED WORK

The existing work on overlay-based pub/sub has mainly focused on implementing practical distributed systems [2, 3, 4, 5, 6, 9, 10, 17, 18, 21, 22], and to the best of our knowledge, the problem of designing an optimal overlay has never been rigorously studied.

Topic-connectivity is an explicit requirement in [4, 9, 12, 21]. In [4, 9, 21], it is achieved using a distributed protocol that maintains a separate overlay, such as a multicast tree [9] or a ring [21], for each topic. These systems however, do not attempt to minimize the average degree of the overlay, which as a result, would be roughly equal twice the average subscription size, regardless of the correlation degree exhibited by the individual node subscriptions in most inputs. Since as we argued in Section 1, many typical subscription patterns tend to exhibit a high degree of correlation, we expect that these solutions could benefit from the link reduction techniques developed in this paper. Furthermore, our GM algorithm could also be used as a baseline to estimate how close to the optimal solution their overlays are in terms of their size.

In [12], we showed experimentally that on many practical workloads exhibiting well-correlated subscription patterns, a simple distributed heuristic can be effective for constructing topic-connected overlays whose average degree is significantly smaller than twice the average subscription size. The results in [12] are however, mostly empirical.

Other systems [2, 3, 5, 10] focus on reducing the number of non-interested relays without explicitly requiring topic-connectivity. Relaxing topic-connectivity can potentially reduce the overlay degree as the nodes are no longer required to have every topic in their subscription being represented in the subscription of some of their neighbors. These systems however, do not explicitly address the tradeoff between the amount of extra overhead incurred by the nodes as a result of forwarding unwanted messages (such as the routing table size, and dissemination and filtering costs) and the overlay degree. In Section 9, we formally capture this tradeoff by introducing a parameterized family of overlay design problems, called SOC. Rigorous study of the SOC problems however, remains by and large the subject of future work.

3. PRELIMINARIES

To simplify the presentation, we do not distinguish between publishers and subscribers. In other words, in order to publish data on a topic t, a node has first to subscribe to t. We model the nodes' subscriptions formally as follows: Given a set of nodes V and a set of topics T, we define an *interest function* over $V \times T$ to be a Boolean-valued function over domain $V \times T$. Subsequently, given an interest function Int, we say that a node v is *interested* in a topic t iff $Int(v,t) = true$.

An *overlay network* (or simply, an *overlay*) over a set of nodes V is a graph of the form (W, E) such that $W = V$ and $E \subseteq V \times V$. Given an overlay network G over V, an interest function Int over $V \times T$, and a topic $t \in T$, we define the *topic-connected components* of G for t to be the connected components of the subgraph of G induced by the nodes $\{v \in V : Int(v,t)\}$[1]. We say that G is *topic-connected* if for each topic t, there is at most one topic-connected component of G for t.

[1]The subgraph of $G = (V, E)$ *induced by* $V' \subseteq V$ is the graph $G' = (V', E')$, where E' contains an edge in E iff both endpoints of this edge are in V'.

4. THE MINIMUM TOPIC-CONNECTED OVERLAY (MIN-TCO) PROBLEM

The *Minimum Topic-Connected Overlay (Min-TCO)* problem is to minimize the number of links needed to create a topic-connected overlay for the given interest assignment. Formally, given a set of nodes V, set of topics T, and an interest function Int over $V \times T$, an instance of Min-TCO(V, T, Int) is defined as follows

DEFINITION 4.1 (MIN-TCO(V, T, Int)). *Construct a topic-connected overlay network $G = (V, E)$ such that $|E| = \min_{E' \subseteq 2^E}\{|E'| : (V, E')$ is topic-connected$\}$.*

We also define TCO(V, T, Int, k), the corresponding decision problem of Min-TCO, which is to determine if it is possible to connect the nodes in V into a topic-connected overlay network G using the given number of edges $k > 0$. Formally, define a language $L_{TCO} = \{\langle V, T, Int, k\rangle :$ there exists an overlay network $G = (V, E)$ which is topic-connected for T and Int, and $|E| = k\}$.

DEFINITION 4.2 (TCO(V, T, Int, k)). *Given $inp = \langle V, T, Int, k\rangle$, decide whether $inp \in L_{TCO}$.*

Note that the above problems are defined in a centralized setting and assume that all of V, T, and Int are fixed in advance, and do not change throughout the execution. We discuss the distributed version of Min-TCO, handling dynamic changes, and the other issues arising in practical distributed systems (e.g., partial knowledge of the node interests) in Section 6.4.

5. COMPLEXITY OF TCO

We prove that TCO(V, T, Int, k) is NP-complete. First, it is easy to see that the topic-connectivity property for a given graph can be verified in polynomial time. Hence, TCO$(V, T, Int, k) \in NP$. Next, we prove the following

LEMMA 5.1. *TCO(V, T, Int, k) is NP-hard.*

The proof of Lemma 5.1 consists of two main steps: First, we define an auxiliary problem, called *Single Node Min-TCO (SN-TCO(V, T, Int, v, d))*, which is given a node $v \in V$ and the degree limit $d > 0$, decide if there is a topic-connected overlay $G = (V, E)$ such that $degree_G(v) \leq d$. Note that SN-TCO is identical to TCO except that it places a constraint on the degree of a single given node rather than all the nodes. The following statement can be proved by reduction from Set Cover (see Appendix A for a detailed proof):

LEMMA 5.2. *SN-TCO(V, T, Int, v, d) is NP-hard.*

We next show in Appendix A that SN-TCO(V, T, Int, v, d) is polynomially reducible to TCO(V, T, Int, k), which concludes the proof of Lemma 5.1. Since TCO$(V, T, Int, k) \in NP$, we conclude that the following holds:

THEOREM 5.3. *TCO(V, T, Int, k) is NP-complete.*

6. SOLVING MIN-TCO

We now present a greedy algorithm that provides an approximated solution to the Min-TCO problem, and analyze its complexity and approximation ratio. For the rest of this section, we fix V, T, and Int be a set of nodes, a set of topics, and an interest function over $V \times T$ respectively.

6.1 The Greedy Merge (GM) Algorithm

The algorithm starts with the overlay network $G = (V, \emptyset)$ so that for each topic $t \in T$, there are $|\{v : Int(v,t)\}|$ singleton topic-connected components of G (i.e., there are $\sum_{t \in T} |\{v : Int(v,t)\}|$ singleton topic-connected components overall). The algorithm proceeds by adding edges to G, thus merging topic-connected components until the resulting overlay contains at most one topic-connected component for each $t \in T$, i.e., G is topic-connected. The edge added at each step is an edge that maximally reduces the total number of topic-connected components. We keep track of the topic-connected components using array NODES over $V \times T$, where NODES$[v][t]$ holds the set of nodes belonging to the same topic-connected component for t as v (see Algorithm 1).

For each new candidate edge (v, w) to be added to the overlay, let $T_{(v,w)} \subseteq T$ be the set of topics such that for each $t \in T_{(v,w)}$, (1) $Int(v,t) \wedge Int(w,t)$, and (2) nodes v and w are the members of two different topic-connected component for t, i.e. NODES$[v][t]$ and NODES$[w][t]$ are disjoint. The addition of (v, w) causes two topic-connected components for each $t \in T_{(v,w)}$ to be merged into a single connected component $C^t =$ NODES$[v][t] \cup$ NODES$[w][t]$. Thus, the contribution of (v, w) to the reduction in the number of topic-connected components for the topics $t \in T_{(v,w)}$ is exactly $|T_{(v,w)}|$. Once edge (v, w) is added to the overlay, the sets NODES$[u][t]$ for each $u \in C^t$ are updated accordingly by setting NODES$[u][t] \leftarrow C^t$.

At each iteration, the algorithm finds the edge (v, w) that maximizes $|T_{(v,w)}|$, and adds it to the overlay. Clearly, every edge addition causes the merge of at least two topic-connected components (for at least one topic) thus reducing the overall number of topic-connected components by at least 1. Whenever an edge with $|T_{(v,w)}| > 0$ cannot be found, the algorithm stops, because this condition implies that the subgraph $G_t \subseteq G$ induced by the subscribers of topic t is connected, for every $t \in T$.

Instead of naively searching for the next best edge, the implementation presented here (see Algorithms 1, 2, and 3) uses an auxiliary array LINKCONTRIB, whose elements LINKCONTRIB$[i]$ are the subsets of the set of all possible edges $V \times V$ with contribution i, i.e., $(v, w) \in$ LINKCONTRIB$[i]$ iff $i = |T_{(v,w)}|$. Note that for each edge $e \in V \times V$, the initial contribution of e is equal to the size of the mutual interest of the nodes it connects. Thus, finding the next best edge $e = (v, w)$ to add to the overlay is easy (Algorithm 3, line 3). However, after adding e to the overlay and removing it from LINKCONTRIB, the contributions of other edges must be updated (lines 6–11). This update can be done efficiently as it is only needed to replace the edges that connect the components that have become connected due to the addition of e. For each such edge, its corresponding entry in LINKCONTRIB can be found in $O(1)$ time if we maintain a pointer to this edge's entry in LINKCONTRIB along with each edge. The NODES sets are then updated as explained above (lines 12–14) to reflect the topic-connected components that have been merged. Once LINKCONTRIB$[i] = \emptyset$ for all i, the algorithm terminates.

Algorithm 1 Data Structure

- ▶ OUTPUTOVERLAYEDGES: a set of overlay edges, initially $\emptyset$.
- ▶ NODES: a 2-dimensional array over $V \times T$ whose elements are subsets of V such that for each $v \in V$, $t \in T$: (1) $Int(v,t) = true$, and (2) for each $w \in$ NODES$[v][t]$: $Int(w,t)$ and both w and v belong to the same topic-connected component for t.
- ▶ LINKCONTRIB: an array of size $|T|$ with elements being sets of edges chosen from $V \times V$. If edge $e \in$ LINKCONTRIB$[i]$, then $e \notin$ OUTPUTOVERLAYEDGES, and adding e to the overlay at the current iteration will reduce the number of topic-connected components by i (where $1 \leq i \leq |T|$).
- ▶ HIGHESTCONTRIB: holds the biggest i for which LINKCONTRIB$[i] \neq \emptyset$.

Algorithm 2 Data Structure Initialization

```
1: for all node v do
2:    for all topic t such that Int(v,t) do
3:       NODES[v][t] ← {v}
4: for all edge e = (v,w) do
5:    contrib ← |{t ∈ T : Int(v,t) ∧ Int(w,t)}|
6:    if contrib > 0 then
7:       add e to LINKCONTRIB[contrib]
8: HIGHESTCONTRIB← max(i|LINKCONTRIB[i] not empty)
```

Algorithm 3 Overlay Construction

```
1: OUTPUTOVERLAYEDGES← ∅
2: while HIGHESTCONTRIB> 0 do
3:    e ← some edge (v,w) from LINKCONTRIB[HIGHESTCONTRIB]
4:    OUTPUTOVERLAYEDGES← OUTPUTOVERLAYEDGES⋃{e}
5:    delete e from LINKCONTRIB[HIGHESTCONTRIB]
6:    for all topic t such that Int(v,t) ∧ Int(w,t) do
7:       for all v' ∈NODES[v][t], w' ∈NODES[w][t], (v',w') ≠ (v,w)
         do
8:          locate i such that (v',w') ∈ LINKCONTRIB[i]
9:          delete (v',w') from LINKCONTRIB[i]
10:         if i > 1 then
11:            add (v',w') to LINKCONTRIB[i − 1]
12:      new_connected_component_list ← NODES[v][t]∪ NODES[w][t]
13:      for all u ∈ new_connected_component_list do
14:         NODES[u][t] ← new_connected_component_list
15:   while HIGHESTCONTRIB> 0 and LINKCONTRIB[HIGHESTCONTRIB]
      is empty do
16:      HIGHESTCONTRIB← HIGHESTCONTRIB-1
17: output (V, OUTPUTOVERLAYEDGES) and halt
```

The following lemma follows immediately from the pseudocode:

LEMMA 6.1. *For every topic t, all the nodes interested in t are organized in a single connected component in the output overlay.*

Next, we analyze the GM's running time.

LEMMA 6.2. *The algorithm terminates within $O(\min(|V|^2, \sum_{t \in T} |\{v \in V | Int(v,t)\}|))$ iterations.*

PROOF. It is easy to see by induction on the number of iterations that at each iteration of the algorithm, for each topic t, the subgraph induced by the nodes interested in t is a forest. Therefore, when the algorithm terminates, there is a tree for each topic $t \in T$. Since exactly 1 edge is added at each iteration, and the number of edges in a tree is the

number of nodes -1, the number of iterations for each topic is bounded by $|\{v \in V | Int(v,t)\}| - 1$. Summing over all the topics $t \in T$, we get the result. □

COROLLARY 6.3. *The output overlay contains* $O(\min(|V|^2, \sum_{t \in T} |\{v \in V | Int(v,t)\}|))$ *edges.*

The running time of GM is given by the following

LEMMA 6.4. *The running time of Algorithm 3 is* $O(\sum_{e=(v,w),e \in E} |\{t \in T | Int(v,t) \wedge Int(w,t)\}|) = O(|V|^2|T|)$.

PROOF. The cost of the GM's initialization (Algorithm 2) is dominated by the calculation of individual edge contribution in line 5. If the interest of each node is stored as a list of topics, the total complexity of this computation for all the edges will be $O(\sum_{e=(v,w),e \in E} |\{t \in T | Int(v,t) \wedge Int(w,t)\}|)$.

The cost of the overlay construction (Algorithm 3) is dominated by the loop in lines 6-14 that updates the contribution of edges as a result of adding an edge to the overlay. The update of each individual edge can be performed in $O(1)$, e.g., if the elements of LINKCONTRIB are implemented as a doubly-linked list, and there are two pointer kept along with each edge, one pointing to i, and the other one pointing to the location of the edge in LINKCONTRIB[i]. In order to calculate the total count of individual edge updates at all the iterations, it is sufficient to notice that every update decrements the contribution of the edge by one (lines 8–11). Algorithm 3 starts when the total contribution of all the edges is $O(\sum_{e=(v,w),e \in E} |\{t \in T | Int(v,t) \wedge Int(w,t)\}|)$ and terminates when the contribution of all the edges is reduced to zero. Therefore, the runtime of the algorithm is $O(\sum_{e=(v,w),e \in E} |\{t \in T | Int(v,t) \wedge Int(w,t)\}|)$. □

6.2 Approximation Ratio

The approximation ratio of GM is given by the following

LEMMA 6.5. *GM has an approximation ratio of at most* $\log(\Sigma_{v \in V} |\{t \in T | Int(v,t)\}|)$ *compared with the optimal solution.*

PROOF. The proof is similar to the proof of the approximation ratio for the set cover problem. Consider an instance of the Min-TCO problem, and let k be the number of overlay links produced by the optimal solution. Note that the total number of connected components is $C_s = \Sigma_{v \in V} |\{t \in T | Int(v,t)\}|$ when Algorithm 3 starts and $C_e = |\{t | t \in T \wedge \exists v \in V$ such that $Int(v,t)\}|$ when it terminates. Let $n_1 = C_s$, and let n_i be the total number of connected components before the i-th iteration of the algorithm. Denote Lg_i the set of links added to OUTPUTOVERLAYEDGES by Algorithm 3 prior to iteration i, LO the set of links in the optimal solution, and $LO_i = LO - Lg_i$. Obviously, adding all the links of LO_i to Lg_i (which would result in the output overlay $LO_i \bigcup Lg_i$) would reduce the total number of connected components to C_e. Since $|LO_i| \leq k$, there exists a link $e \in LO_i$ such that adding e to Lg_i (i.e., making Lg_{i+1} to be $Lg_i \bigcup \{e\}$) would reduce the number of connected components by at least $(n_i - C_e)/k$. Since Algorithm 3 is greedy, it picks a link that reduces at least as many connected components. Hence $n_i - n_{i+1} \geq (n_i - C_e)/k$, which can be rewritten as $n_{i+1} - C_e \leq (1 - 1/k)(n_i - C_e)$. This shows that the number of iterations used by the greedy algorithm does not exceed the smallest l that satisfies $(n_1 - C_e)(1 - 1/k)^l < 1$, which implies $l \leq k \ln(C_s - C_e) \leq k \ln C_s$. □

6.3 Tightness of Approximation Ratio

We show that the GM's approximation ratio of $\log(\Sigma_{v \in V} |\{t \in T | Int(v,t)\}|)$ established by Lemma 6.5 is tight. To this end, we construct an input example on which the greedy algorithm achieves an approximation ratio of $O(\log |T|)$.

LEMMA 6.6. *There exist an input* (V, T, Int) *on which GM achieves an approximation ratio of* $O(\log |T|)$.

PROOF. Our input example is parameterized by two integer numbers k and m, $m > 2$. T consists of $2 \cdot (2^k - 1) \cdot m$ topics, which can be visualized as organized in a cuboid of size $2 \times (2^k - 1) \times m$ (see Figure 1). There exist $2 + m + k$ nodes in the system with their interests as follows: the interest of two nodes (denoted $Stwo_1$ and $Stwo_2$) is represented by each of the two planes with $(2^k - 1) \times m$ topics, respectively, so that their combined interest covers all of T. The interest of k nodes can be described by splitting a $2 \times (2^k - 1)$ plane into k pairwise disjoint sets $Sk_1, \ldots, Sk_k$ with sizes $2, 4, 8, \ldots, 2^k$ respectively, and expanding each of the sets in the dimension of m. Therefore, the resulting size of these k interest sets is $(2 \times m), (4 \times m), \ldots, (2^k \times m)$, respectively. The interest of the last m nodes Sm_1 to Sm_m is represented by each of the m planes with $2 \times (2^k - 1)$ topics.

Observe that the following overlay induces a single connected component for every topic on this input: each of $Stwo_1$ and $Stwo_2$ nodes is connected to every Sm_i and to every Sk_i nodes. This solution uses $2m + 2k$ links. In fact, it is easy to see that this is the optimal solution for Min-TCO, even though this fact is not necessary for establishing the upper bound.

Consider the behavior of GM on this input.The intersection between each Sk_i and Sm_j is 2^i topics while the intersection between each Sk_i and $Stwo_j$ is $m2^{i-1}$ topics. The latter is larger because $m > 2$. Therefore, the algorithm will choose ($Stwo_1$, Sk_k) and ($Stwo_2$, Sk_k) links at the first two iterations. Furthermore, it will connect each $Stwo_i$ with each Sk_j at the first $2k$ iterations. At this point, however, it will start being suboptimal and connect each of Sm_i with Sk_k rather than with $Stwo_1$ and $Stwo_2$. Eventually, each of Sm_i will be linked to each of Sk_j, at which point the algorithm will stop. The total number of links added by this algorithm will be $2k + km$.

Therefore, the approximation ratio for this input will be $(2k + km)/(2k + 2m)$. If we choose $k = m$, then $|T| = O(2^k)$, and the ratio becomes $(2 + k)/4 = O(k) = O(\log |T|)$. □

6.4 Practical Considerations

Due to its relatively low time complexity (which as we show in Section 8, is only a small fraction of the worst case bound under well-correlated subscription patterns), the centralized version of the GM presented above can be used in practice, e.g., for network planning in the systems where subscriptions are relatively stable, and do not change frequently. It could also be used as a baseline for evaluating scalability of the existing overlay-based pub/sub implementations.

Note also that it is straightforward to devise a distributed version of GM, that would be exactly as above, provided each node is given a view of the other node interest assignments. However, the correctness of GM in this case rests upon the completeness and consistency of the individual node views, which could be problematic, especially in large

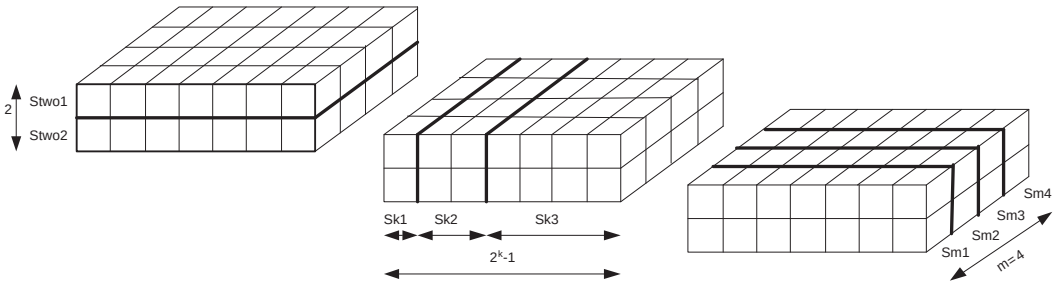


Figure 1: A geometrical illustration for specific values of $k = 3$ and $m = 4$.

and highly dynamic settings. This calls for further study to investigate the possibility of devising more adaptive variants of GM (such as e.g., an on-line version) where only a small portion of the overlay has to be re-built in response to the dynamic changes.

We also observe that it is possible to improve connectivity and reduce diameter of the topic-connected components constructed by GM, by running GM several times (depending on the desired connectivity level), with $(V \times V) \setminus E$ (where E is the set of edges already in the overlay) being the set of edges considered for inclusion at the beginning of each individual run. The precise analysis of this algorithm and its guarantees is the subject of future study.

7. INAPPROXIMABILITY OF MIN-TCO

We now show that for all practical applications, the approximation ratio achieved by GM is almost tight by establishing the following general inapproximability result:

LEMMA 7.1. *There exists no polynomial algorithm that achieves a constant factor approximation for Min-TCO unless $P = NP$.*

PROOF. Assume that such an algorithm A exists. We show how to use A to achieve a constant factor approximation for the minimal set cover problem, which is known to be impossible if $P \neq NP$.

Take an instance (U, S) of the minimal set cover problem. Denote $S_{opt} \subseteq S$ an optimal solution for this instance (S_{opt} cannot be found algorithmically in polynomial time if $P \neq NP$). We build an instance Min-TCO(V, T, Int) using a construction technique that is similar to the example above and that is also parameterized by integer m: we take T with $m \times |U|$ topics so that each element u of universe U corresponds to m topics that are denoted $t_i(u), 1 \leq i \leq m$. We take V of size $|S|+m$: V includes m new nodes $vm_i, 1 \leq i \leq m$ and $|S|$ nodes so that each set $s \in S$ corresponds one-to-one to node $v(s)$. Int is constructed as follows: $\forall i, 1 \leq i \leq m, vm_i$ is interested in $\bigcup_{u \in U} t_i(u)$ and $\forall s \in S, v(s)$ is interested in $\bigcup_{u \in s, 1 \leq i \leq m} t_i(u)$.

Consider the following overlay for Min-TCO(V, T, Int): $\forall i, 1 \leq i \leq m, \forall s \in S_{opt}$, there is a link between vm_i and $v(s)$. In addition, there is a link between every pair of $v(s)$, $v(s')$ nodes, $s, s' \in S$. This overlay uses $|S_{opt}|m + |S|^2$ links and induces a single connected component for every topic. This gives an upper bound on the number of links in an optimal solution for Min-TCO(V, T, Int). Since A achieves a constant factor approximation, $\exists c$ such that A will produce a number of links that is $\leq c(|S_{opt}|m + |S|^2)$.

Observe that any solution for Min-TCO(V, T, Int) can be used to derive m set covers for U as follows: assume an overlay $G = (V, E)$ that induces a single connected component for each topic. For each i, $1 \leq i \leq m$, let $E_i = \{(vm_i, v(s)) \in E | s \in S\}$. By topic-connectivity, each $C_i = \{s | (vm_i, v(s)) \in E_i\}$ is a set cover of U. Since all E_i's are pairwise disjoint, any algorithm guarantees that $\sum_{i=1}^{m} |C_i| \leq |E|$. Therefore, there exists at least one i such that $|C_i|m \leq |E|$. Hence, A can be used to produce a set cover S_A for the original instance of the minimal set cover problem so that $|S_A|m \leq |E| \leq c(|S_{opt}|m + |S|^2)$. Thus, $|S_A| \leq c(|S_{opt}| + |S|^2/m)$. If we take $m = |S|^2$, the latter expression becomes $c(|S_{opt}| + 1) \leq 2c|S_{opt}|$. Therefore, A allows us to obtain a constant factor approximation for the minimal set cover problem. □

8. EVALUATION

We implemented a centralized version of GM as appears in Algorithm 3. As in other studies, e.g., [17], in each of our experiments, both the number of topics and the number of nodes are fixed. We varied the number of topics from 100 to 200, and the number of nodes from 1000 to 10,000. Each node is subscribed to s topics. We run experiments with several values of s and we got similar trends in all the experiments. Due to space limitations, we only report on the experiments with s=10, except for Section 8.3 in which we also report on the experiments with s=20.

We construct our workloads as follows: given a collection of topics T, each topic $t_i \in T$ is associated with probability p_i, $\sum_i p_i$=1, so that each node subscribes to t_i with a probability p_i. The value of p_i is distributed according to a Zipf, an exponential, or a uniform distribution. We use a Zipf distribution with $\alpha = 0.5$ (i.e., $p_i \propto \frac{1}{i^{0.5}}$). Our choice of this topic popularity distribution is based on a recent study [16] that shows it faithfully describes the feed popularity distribution in RSS. In the exponential distribution we use, the probability to choose one of the 10% most popular topics is set 0.55. This distribution was shown in [20] to be a good predictor of stock popularity in the New York Stock Exchange (NYSE). Below, we study the effects of the number of nodes and topics and the topic popularity distribution on the average node degree (see Section 8.1), and the running time (see Section 8.2). In Section 8.3, we study scalability benefits of exploiting subscription correlation.

8.1 The Average Node Degree

Figure 2 depicts the average node degree (i.e., 2 × overall number of edges/number of nodes) in experiments with GM. As the figure shows, with *all* the topic popularity distribu-

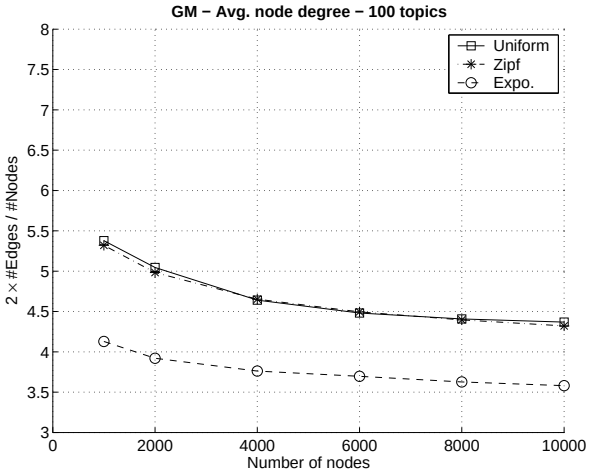


Figure 2: Average node degree for different number of nodes and topic popularity distributions.

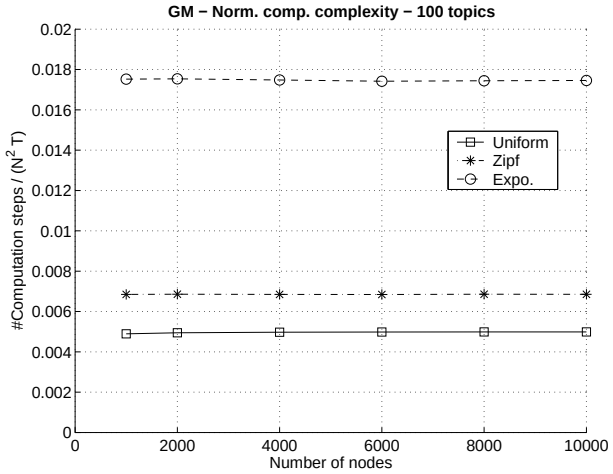


Figure 3: Running time for different number of nodes and topic popularity distributions.

tions, the average node degree *decreases* with the number of nodes. The reason for that lies in the randomness of the subscription patterns. In particular, increasing the number of nodes increases the chances for a given node to find a neighbor with a bigger interest overlap thus reducing overall number of neighbors needed to maintain topic-connectivity. Likewise, the overlay constructed in the experiments with the exponential popularity distribution is smaller than that in the experiments with uniform and Zipf distributions because of a substantially higher subscription correlation exhibited by the exponential workload. We also observe that increasing the number of topics also increases the average node degree as that results in a workload with less correlated subscriptions.

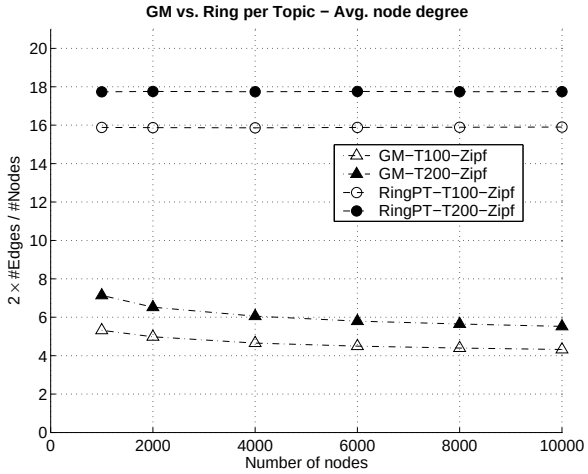
8.2 The Running Time

Recall that the running time of GM is $O(|V|^2|T|)$. Figure 3 depicts the actual number of processing steps it takes GM until termination divided by $|V|^2|T|$. Remarkably, in *all* of our experiments, GM's running time is just a small fraction (in the order of 10^{-3}) of the worst case analysis bound. This implies that GM is feasible for many practical workloads. In addition, we observe that the GM's running time behaves like $|V|^2$, and decreases with the number of topics. This is because, as we noted in the previous section, increase in the number of topics with the rest of the parameters being fixed results in less correlated subscriptions. We also note that, interestingly, for a given number of nodes and topics, although exponential workload results in a smaller overlay (see Figure 2), the GM's running is nevertheless the highest on this workload. The reason for that lies in a higher subscription correlation exhibited in the exponential workload which results in a higher cost of updating the connected components after each individual edge addition.

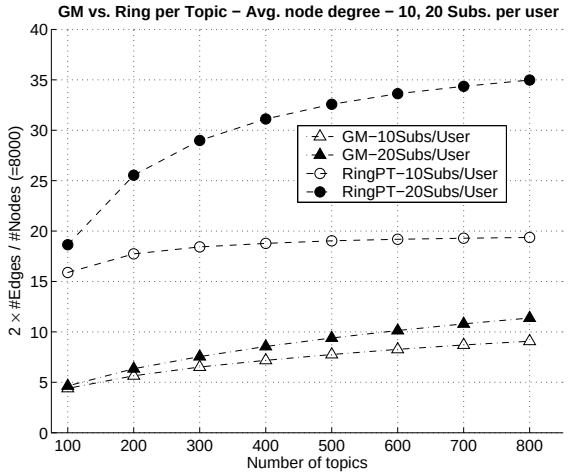
8.3 Scalability Benefits of Exploiting Subscription Correlation

To see that exploiting interest correlation could indeed result in a substantial scalability improvement, we compare the average degrees of the overlays created by GM and by the *Ring-Per-Topic (RingPT)* protocol (similar to [21]) in which topic-connectivity is achieved by maintaining a separate logical ring, ordered by the node identifiers, for each topic.

We compare the average node degrees obtained in the experiments with GM with those in obtained with RingPT. Due to space limitations, we only report on experiments with the Zipf topic popularity distribution (similar results were



(a) Average node degree for different number of nodes, 10 subscriptions per-node.



(b) Average node degree for different number of topics, 10 and 20 subscriptions per-node.

Figure 4: Average node degree - The GM algorithm vs. ring per-topic algorithm.

obtained for the exponential and the uniform distributions as well). As Figure 4(a) shows, the average node degree in the experiments with RingPT is roughly 3 times the average node degree in the experiments with GM. In Figure 4(b), we study the effect of the average subscription size on the average node degree. Remarkably, with GM, increasing the number of subscriptions per-node from 10 to 20 results in just 25% increase in the average node degree at the worst, whereas with RingPT, doubling subscription size results in roughly doubling the average node degree. This further attests to our claim that maintaining topic-connectivity without exploiting correlation results in the average node degree being roughly equal to twice the average subscription size.

9. OTHER OVERLAY DESIGN PROBLEMS FOR PUB/SUB

The Min-TCO problem discussed so far focuses on a single (though important) aspect of the overlay design for pub/sub systems, namely, constructing a topic-connected overlay using the least possible number of links. It is a natural question to ask to what extent we could improve scalability if we are ready to compromise the full topic connectivity to some extent. It is also interesting to further investigate this tradeoff with respect to additional parameters affecting the message routing efficiency, such as propagation latency, filtering overhead, robustness, etc. In this section, we propose a formal framework that extends and generalizes Min-TCO to capture a multitude of the objectives and constraints affecting performance and scalability of the overlay-based pub/sub systems. Namely, we introduce a parameterized definition of the *Scalable Overlay Construction (SOC)* problem for pub/sub with multiple topics, discuss several representative SOC problems, and outline complexity results for some of them.

The SOC Problem: Intuitively, the objective of the SOC problem is to capture a tradeoff between the overlay scalability, represented as an abstract *degree function* δ, and the cost of message dissemination, represented as an abstract *routing cost* function ρ. Formally, let V, T, Int be a set of nodes, set of topics, and an interest function over $V \times T$ respectively. Fix δ to be a function from all possible overlay networks over V to $\mathbb{R}$; and ρ to be a function mapping pairs (overlay over V, Int) to $\mathbb{R}$. An instance of the SOC decision problem, $SOC(V, T, Int, d, C, \delta, \rho)$, is to determine if there is an overlay network G over V such that (1) $\delta(G) \leq d$ and (2) $\rho(G, Int) \leq C$.

We believe that the SOC problem will be a valuable tool for formally studying various overlay design problems for pub/sub, whose definition and analysis is mostly the subject of future work. Below, we give two representative examples of the problems that can be derived from the general SOC problem above:

Filtering: Determine if there is an overlay G over V whose average (or maximum) degree $\leq d$, and the cumulative cost of message filtering taken over all the topics in T does not exceed C. Here, the filtering cost for a topic $t \in T$ is defined to be the number of nodes not interested in t in a Steiner tree over G whose terminals are the nodes interested in t. In the full version of the paper [11], we prove that Filtering is NP-hard.

Diameter: Determine if there is an overlay G over V such that the average (or maximum) degree of $G \leq d$, and the average (or maximum) topic diameter (where the topic diameter for $t \in T$ is defined to be the maximum distance in G between any pair of nodes interested in t) does not exceed C.

10. CONCLUSIONS

We initiated a formal study of the problem of designing a scalable overlay network to support efficient decentralized pub/sub communication. We introduced a new optimization problem, Min-TCO whose objective is to minimize the number of links needed to create a topic-connected (and thus optimal with respect to the routing cost) overlay network.

We showed hardness of Min-TCO by proving that its decision version (TCO) is NP-complete. On the positive side, we presented the Greedy Merge (GM) algorithm, which has a polynomial running time and a logarithmic approximation

ratio. We proved that GM is almost tight for most practical uses by showing that no polynomial algorithm can approximate Min-TCO within a constant factor (unless P=NP).

Our experimental results demonstrate that under realistic workloads, the overlay networks constructed by GM are significantly more scalable than those constructed by straightforward implementations of topic-connectivity. Our results also highlight the impact of interest correlation among the nodes and emphasize the fact the GM algorithm exploits this correlation in a manner that improves scalability.

In order to formally study scalability of overlay constructions in the context of other parameters affecting pub/sub performance, we introduced a new parameterized family of decision problems, called SOC, and gave examples of two representative problems. The further study of the SOC problems as well as extending GM to dynamic and/or distributed settings is the subject of future work.

11. ACKNOWLEDGMENTS

The authors would like to thank John Doucier, Hana Chockler, and the anonymous reviewers whose many comments and suggestions resulted in a substantial improvement of the presentation and technical exposition.

12. REFERENCES

[1] *Oracle9i Application Developers GuideAdvanced 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. In *1th 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. In *Proceedings of IEEE INFOCOM 2004*, Hong 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):1489–1499, 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 1194–1204. Springer Verlag, 2005.

[11] G. Chockler, R. Melamed, Y. Tock, and R. Vitenberg. Constructing scalable overlay networks for pub/sub with many topics. Manuscript in progress, 2007.

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

[13] P. T. Eugster, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec. The many faces of publish/subscribe. *ACM Computing Surveys*, 35(2):114–131, 2003.

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

[15] R. Lewis. *Advanced Messaging Applications with MSMQ and MQSeries*. QUE, 1999.

[16] 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.

[17] V. Ramasubramanian, R. Peterson, and E. G. Sirer. Corona: A high performance publish-subscribe system for the world wide web. In *NSDI*, 2006.

[18] 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.

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

[20] Y. Tock, N. Naaman, A. Harpaz, and G. Gershinsky. Hierarchical clustering of message flows in a multicast data dissemination system. In *17th IASTED Int'l Conf. Parallel and Distributed Computing and Systems*, pages 320–327, 2005.

[21] 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.

[22] S. Zhuang, B. Zhao, A. Joseph, R. Katz, and J. Kubiatowicz. Bayeux: An architecture for scalable and fault tolerant wide-area data dissemination. In *11th International Workshop Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, pages 11–20, June 2001.

APPENDIX

A. NP-HARDNESS OF TCO

Proof of Lemma 5.2. The proof goes by reducing the problem of set cover to SN-TCO. The decision problem of set cover is defined as follows: given a universe set U and a collection S of subsets of U, a set cover is a subcollection $C \subseteq S$ such that $\bigcup_{s \in S} s = U$. The problem is given a tuple (U, S, k), $k \in \mathbb{N}$ to determine if there is a set cover C of size $|C| = k$. The set cover problem is one of the first problem discovered to be NP-complete.

Given an instance (U, S, k) of the set cover problem, we

construct an instance SN-TCO(V, T, Int, v, d) in the following way: we take T with topics that one-to-one correspond to the elements of U. We take V of size $|S| + 1$: V includes one special node v and $|S|$ nodes that one-to-one correspond to the elements of S. We build the interest binary relation Int in the following way: v is interested in all $|T|$ topics. For any other node w, we construct its interests based on the data of the corresponding subset $s \in S$: w is interested in a topic t iff s includes an element $u \in U$ that corresponds to t. Finally, we take $d = k$. We now prove that there is an overlay matching this instance of the SN-TCO problem if and only if there is a set cover matching the original instance of the set cover problem.

Suppose there is an overlay G such that $degree_G(v) \leq d$ and for every topic, all the nodes interested in this topic create a single connected component of G. Denote by VN the set of neighbors of v in G: $VN = \{w | w \in V, (v, w) \in E\}$. Consider $C = \{s | s \in S, s \text{ corresponds to } w, w \in VN\}$. Note that $\forall t \in T, \exists w \in V, w \neq v, Int(w, t)$ because $\bigcup_{s \in S} s = U$. Therefore, $\forall t \in T, \exists w \in VN, Int(w, t)$ because otherwise v and the other nodes interested in t will not form a single connected component. Hence, $\bigcup_{s \in C} s = U$, and C is a set cover. Furthermore, $|C| = |VN| = degree_G(v) \leq d = k$ (it is possible to add other elements of S to C to make $|C| = k$ if necessary).

Suppose there is a set cover C of size k. Consider an overlay G in which all nodes other than v are connected to each other: $\forall w \in V, \forall w' \in V, w \neq v, w' \neq v \rightarrow (w, w') \in E$, and v connected to those and only those nodes that correspond to the elements of C. Obviously, $degree_G(v) = |C| = k = d$. Furthermore, for every topic t, there is a neighbor of v that is interested in t because $\bigcup_{s \in C} s = U$. Therefore, all the nodes interested in t create a single connected component of G.

While this concludes the proof, it is interesting to note that there is a similar reverse reduction from SN-TCO to the set cover problem. Thus, the two problems are isomorphic. □

Proof of Lemma 5.1. The proof goes by reducing the problem of SN-TCO to TCO. Assume $d \geq 3$ (in the special case of $d \leq 2$, SN-TCO is trivially in P). Given an instance (V, T, v, Int, d) of the SN-TCO problem, we construct an instance (V', T', Int', d') of the TCO problems as follows. We take $T' = T$ and $d' = d$. V' includes v and every other node $w \neq v$ of V duplicated $|V| - 1$ times. Thus, the size of V' is $1 + (|V| - 1) \times (|V| - 1)$. Denote $Gr(w)$ the set of $|V| - 1$ duplicated nodes in V' that correspond to the node w of V. We will call w the spawning node of $w', w' \in Gr(w)$ and denote it by $spawning(w')$. Int' is constructed as follows: Int' retains the interests of v in Int so that $\forall t \in T, Int'(v, t)$ iff $Int(v, t)$. For all nodes of V' other than v, their interests are identical to those of their spawning nodes in V: $\forall t \in T, \forall w' \in V', w' \neq v, Int'(w', t) \leftrightarrow Int(spawning(w'), t)$.

We now prove that there is an overlay matching this instance of the TCO problem if and only if there is an overlay matching the original instance of the SN-TCO problem. Suppose there is an overlay $G' = (V', E')$ such that $degree(G') \leq d$ and for every topic, all the nodes interested in this topic create a single connected component of G'. Consider $G = (V, E)$ wherein E is constructed by connecting all nodes of V other than v to each other and connecting v as follows: $\forall w \in V, w \neq v, (v, w) \in E$ iff $\exists w' \in V'$ such that $w = spawning(w') \wedge (v, w') \in E'$. In other words, we obtain the edges of v in G by grouping the edges of v in G' and make a full clique of all nodes of G other than v. Note that $degree_G(v) \leq degree_{G'}(v) \leq d$. Assume there is a topic t such that all the nodes interested in t do not create a single connected component in G. This is only possible if $Int(v, t) \wedge \exists w \in V, w \neq v, Int(w, t) \wedge \forall w \in V, Int(w, t) \rightarrow (v, w) \notin E$. Then, $\exists w' \in V', w' \neq v, Int'(w', t) \wedge \forall w' \in V', Int'(w', t) \rightarrow (v, w') \notin E'$. Therefore, all the nodes of G' interested in t do not create a single connected component in G', which is a contradiction.

Suppose there is an overlay $G=(V, E)$ such that $degree_G(v) \leq d$ and for every topic, all the nodes interested in this topic create a single connected component of G. Consider $G' = (V', E')$ wherein E' is constructed in the following manner. For each $w \in V, w \neq v$, all $|V| - 1$ nodes of $Gr(w)$ are connected to form a cycle. Let us impose an order on the cycles and on the nodes within each cycle. Each node of $Gr(w)$ may only have a single edge besides the two edges of the cycle, so that $\forall w' \in V', w' \neq v, degree_{G'}(w') \leq 3 \leq d$. Specifically, for every two cycles Gr_i and Gr_j, we connect node w_i^j of Gr_i with node w_j^i of Gr_j. This way, for every topic t, all the nodes of G' other than v that are interested in t form a single connected component. For every cycle Gr_i, its node w_i^i, which does not have an edge to any other cycle, is used to form a connection to v: $\forall w \in V, w \neq v, Gr_i = Gr(w), (v, w_i^i) \in E'$ iff $(v, w) \in E$.

Note that $degree_{G'}(v) = degree_G(v) \leq d$. Since for every other node, its degree in G' is limited by 3, $degree(G') \leq d$. Assume there is a topic t such that all the nodes of G' interested in t do not create a single connected component in G'. This is only possible if $Int(v, t) \wedge \exists w' \in V', w' \neq v, Int'(w', t) \wedge \forall w' \in V', Int'(w', t) \rightarrow (v, w') \notin E'$. Then, $\exists w \in V, w \neq v, Int(w, t) \wedge \forall w \in V, Int(w, t) \rightarrow (v, w) \notin E$. Therefore, all the nodes of G interested in t do not form a single connected component in G, which is a contradiction. □