Constructing Fault-Tolerant Overlay Networks for Topic-based Publish/Subscribe

Chen Chen
University of Toronto
chenchen@eecg.toronto.edu

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

Hans-Arno Jacobsen
University of Toronto
jacobsen@eecg.toronto.edu

Abstract – We incorporate fault tolerance in designing reliable and scalable overlay networks to support topic-based publish/subscribe communication. We propose a new family of optimization problems, named MinAvg-kTCO, that captures the trade-offs among several key dimensions including fault tolerance, scalability, performance, and message dissemination. Roughly speaking, the MinAvg-kTCO problem is: use the minimum number of edges to create a *k-topic-connected overlay* ($kTCO$) for pub/sub systems, i.e., for each topic the sub-overlay induced by nodes interested in the topic is k-connected.

We prove the NP-completeness of MinAvg-kTCO and show a lower-bound for the hardness of its approximation. With regard to the MinAvg-2TCO problem, we present the first polynomial time algorithm, namely GM2, with a guaranteed approximation factor relative to the optimum. We show experimentally that on representative publish/subscribe workloads, the GM2 algorithm outputs $2TCO$ at the cost of an empirically insignificant increase in the average node degree, which is around 1.65 times that of $1TCO$ produced by the algorithm with the best known approximation ratio. Besides, GM2 reduces the topic diameters around 50% as compared to those in the baseline $1TCO$.

With regards to the MinAvg-kTCO problem, where $k \geq 2$, we propose a simple and efficient heuristic algorithm, namely HararyPT, that aligns nodes across different sub-overlays. We show the practical scalability of HararyPT for highly correlated pub/sub workloads in terms of the number of nodes, the number of topics, and the number of subscriptions per node.

I. Introduction

Publish/Subscribe (pub/sub) systems constitute an attractive choice as the communication paradigm and messaging substrate for building large-scale distributed systems. Many real-world applications are using pub/sub for message dissemination, such as application integration across data centers [1], [2], file synchronization in distributed storage systems [3], financial data dissemination [4], RSS feed aggregation, filtering, and distribution [5], business process management [6], and algorithmic trading [7].

In the topic-based pub/sub model, a publisher associates its publication message with a specific topic and subscribers register their interest in a subset of all topics. Topic-based pub/sub is adopted by many large-scale systems and applications [1], [2], [3], [5].

A distributed topic-based pub/sub system is often organized as an application-level overlay of brokers (e.g., servers or simply referred to as nodes) connected in a federated or in a peer-to-peer manner [8]. The overlay infrastructure directly impacts the pub/sub system's performance and scalability, such as the message routing cost. Constructing a high-quality broker overlay is a fundamental problem that has received attention both in industry [1], [2] and academia [9], [10], [11], [12], [13], [14], [15].

Gregory Chockler *et al.* define a *topic-connected overlay* (TCO), as an overlay, where all nodes (i.e., pub/sub brokers) interested in the same topic are organized in a connected dissemination sub-overlay [9]. A TCO ensures that nodes not interested in a topic never need to contribute to disseminating information on that topic. Publication routing atop TCOs saves bandwidth and computational resources otherwise wasted on forwarding messages of no interest to the node. Topic-connectivity also results in more efficient routing protocols, a simpler matching engine design, and smaller forwarding tables. From a security perspective, TCOs are desirable when messages are to be shared across a network among a set of trusted users without leaving this set.

Unfortunately, topic-connectivity per se does not address critical reliability requirements for the pub/sub overlay. In particular, there is no guarantee that topic-connectivity is preserved under even a single node crash. That is, all the desirable properties about TCOs are fragile and easily break in a dynamic environment. The root cause for this lies in the definition of TCO and TCO-related problems [9], [10], [11]. These definitions make an implicit assumption that the pub/sub overlay is reliable and robust, i.e., nodes and links in the network are fault-free.

In order to address this shortcoming, we propose a problem of constructing a k-topic-connected overlay ($kTCO$): topic-connectivity still holds as long as fewer than k nodes fail simultaneously on the same topic (see Def. 1 in §IV). The extension from TCO to $kTCO$ captures the overlay's resilience to churn by introducing a safety factor, k. This safety factor is important from an engineering perspective because pub/sub systems are dynamic in nature. Node churn may occur due to administrative maintenance or inevitable failures, such as hardware faults, misconfigurations, or software bugs [16]. In practice, the set of active machines in a data center shows non-negligible variations over time [17]. Furthermore, the advent of new pub/sub applications, e.g., in sensor networks [18], [19] or mobile networks [20], [21], makes it increasingly important and challenging to enable the overlay's reliability. In these scenarios, overlay nodes are not necessarily dedicated servers or brokers, and the pub/sub system is subjected to growing dynamism and additional resource constraints.

Advocates for TCO-structured pub/sub overlays might argue that $kTCO$ is not necessary. In principle, the TCO can always be reconstructed in the presence of churn. However, this is impractical and wasteful since state-of-the-art algorithms suffer from a high computational complexity [9], [10], [11], [14], [15], [22]. On the other hand, a few pub/sub systems (e.g., [23], [24], [25], [26]) have explored the problem of dynamically maintaining the TCO. Basically, these approaches constantly make incremental adjustments of the overlay in presence of churn. However, the overlays they produced are not as optimal in terms of the node degree as the centralized algorithms for TCO construction, as corroborated by experimental studies, e.g., in [25]. Besides, approaches for incremental overlay maintenance can be applied to $kTCO$ as well to produce even more reliable solutions.

Furthermore, $kTCO$ can lead to better performance. First, $kTCO$ indicates that k disjoint data paths exist from end to end for each topic (see Merger's Theorem [27]). Thus, we can harvest network intelligence in the routing protocols on top of $kTCO$ by steering the traffic among multiple alternate paths in a more optimized and secure manner. Second, it is possible to reduce the diameter of the overlay, as we improve its connectivity [9]. With lower diameter, message delays are likely to be diminished because fewer hops are needed for message delivery.

Nevertheless, these merits of $kTCO$ come with a price – additional links are required. Intuitively, a sparse overlay is unlikely to be $kTCO$, while a dense overlay is sub-optimal with respect to node degree. However, it is also imperative for a pub/sub overlay network to have low node degrees. This is because it costs a lot of resources to maintain adjacent links for a high-degree node (i.e., monitor links and neighbors [9], [11]). For a typical pub/sub system, each link would also have to accommodate a number of protocols, service components, message queues, and so on. While overlay designs for different applications might be principally different, they all strive to maintain bounded node degrees, e.g., DHTs [28], wireless networks [29], and survivable network designs [30].

In this paper, we formally study the fundamental trade-offs between attaining the $kTCO$ property while preserving low node degrees. Our main contributions are as follows:

- We propose the MinAvg-kTCO problem of devising $kTCO$ with the minimum number of links (see Problem 1 in §IV). Formally, we prove the NP-completeness of the MinAvg-kTCO problem. We also show that MinAvg-kTCO is difficult to approximate within a logarithmic ratio (§IV).
- We design two algorithms for the MinAvg-kTCO problem. First, with regards to the MinAvg-2TCO problem, we present the first polynomial-time approximation algorithm, namely the GM2 algorithm in §V. We provide an approximation ratio for GM2, which almost meets the lower bound on the approximation ratio for the problem. Our proof of GM2's approximation ratio exhibits novelties in several respects, including the concept of ear decomposition based on an edge sequence, the amortized analysis to measure the progress of the algorithm, the estimate of edge contribution towards $2TCO$, the charging argument against the optimal solution, and the mathematical analysis using number theory (see the detailed proof in §VI). Second, with regards to the MinAvg-kTCO problem, where $k \geq 2$, we propose a simple and efficient heuristic algorithm, namely HararyPT, that aligns nodes across different sub-overlays (see §VII).
- In §VIII, we validate both GM2 and HararyPT with comprehensive experiments under a variety of characteristic pub/sub workloads of up to 1 000 nodes, 1 000 topics, and 100 subscriptions per node. GM2 requires an empirically small amount of additional edges to obtain a $2TCO$, whose average node degree is around 1.5 times that of the $1TCO$ produced by the algorithm with the best known approximation ratio. Besides, GM2 improves the topic diameters, which are around 0.5 of those in the baseline $1TCO$. To achieve $kTCO$ ($k \geq 2$) for highly correlated pub/sub workloads, we show the practical effectiveness and scalability of the HararyPT algorithm with respect to the number of nodes, the number of topics, and the number of subscriptions per node.

II. Related Work

A significant body of research has been considering the construction of an overlay topology underlying pub/sub systems such that network traffic is minimized (e.g., [9], [10], [12], [13], [14], [11], [24], [15], [26]). Topic-connectivity is a required property in [31], [23], [25], [26]. It is an implicit requirement in [32], [33], [13], [34], [24], which all aim to reduce the number of intermediate overlay hops for a message to travel in the network.

Gregory Chockler *et al.* explored the MinAvg-TCO problem of constructing a TCO with a minimum number of connections [9]. Following this direction, a number of problems were formulated in constructing TCO while optimizing node degrees and other criteria [9], [10], [11], [14], [15], [22]. Unfortunately, this body of work did not address the critical reliability requirements for a pub/sub overlay.

Some pub/sub systems (e.g., [23], [24], [26]) build and maintain the TCO in a decentralized manner. These systems implement non-coordinated decentralized overlay construction protocols such that each node decides upon its own neighbors. These protocols are generally efficient for handling dynamism, because they operate with only local and partial knowledge. Since these approaches are TCO-based, our algorithms to build $kTCO$ have the potential to complement them, thus achieving more reliability, robustness, and adaptiveness in pub/sub overlays.

The classical graph theory about *connectivity* [27] serves as solid bedrock for us to tackle the reliability of pub/sub overlays, including problem formulation, algorithm design, and performance analysis.

III. Background

In this section, we present some notation and background information, essential for the problem formulation, the algorithm design and analyses, and evaluations in this paper.

Let $I(V, T, Int)$ represent an input instance, where V is the set of nodes, T is the set of topics, and Int is the interest function such that $Int : V \times T \rightarrow \{true, false\}$. Since the domain of the interest function is a Cartesian product, we also refer to this function as an interest matrix. Given an interest function Int, we say that a node v is interested in some topic t if and only if $Int(v, t) = true$. We also say that node v subscribes to topic t.

We denote a *topic-based pub/sub overlay network* (TPSO) as $TPSO(V, T, Int, E)$. A $TPSO(V, T, Int, E)$ can be illustrated as an undirected graph $G = (V, E)$ over the node set V with the edge set $E \subseteq V \times V$. Given $TPSO(V, T, Int, E)$, the sub-overlay *induced* by $t \in T$ is a subgraph $G^{(t)} = (V^{(t)}, E^{(t)})$ such that $V^{(t)} = \{v \in V | Int(v, t)\}$ and $E^{(t)} = \{(v, w) \in E | v \in V^{(t)} \wedge w \in V^{(t)}\}$. A *topic-connected component* (*TC-component*) on topic $t \in T$, is a maximal connected subgraph in $G^{(t)}$. A $TPSO$ is called *topic-connected* if for each topic $t \in T$, $G^{(t)}$ has at most one *TC-component*. We denote the *topic-connected overlay* as $TCO(V, T, Int, E)$, TCO for short.

IV. The parameterized MinAvg-kTCO problem and its complexity

The definition of a k-connected graph [27] can be directly applied to the sub-overlay induced by a topic $t \in T$. We call a $TCO(V, T, Int, E)$ **k-connected for topic** $t \in T$ if $G^{(t)} = (V^{(t)}, E^{(t)})$ is k-connected, i.e., $|V^{(t)}| > k$ and $G^{(t)} - X = (V^{(t)} - X, E^{(t)} \backslash \{e(v, w) | \text{either } v \in X \text{ or } w \in X\})$ is connected for every $X \subseteq V^{(t)}$ with $|X| < k$.

We want to extend the definition of k-connectivity to a $TPSO$ considering all topics in T. However, given a parameter k, $|V^{(t)}|$ might be smaller than k for some topic $t \in T$; in these cases, "k-connectivity" is not defined in classic graph theory, but we need to adopt a convention for $TPSO$. Intuitively, for a fixed k, a k-topic-connected overlay should have the property that the $TPSO$ can still provide pub/sub service (for all topics) as long as fewer than k nodes fail simultaneously on the same topic $t \in T$. If $|V^{(t)}| < k$, the removal of $(k-1)$ nodes on t implies that none subscribes to t any more, and thus the overlay no longer serves t. To ensure the pub/sub service continues with topic t under other cases, we need to make sure $G^{(t)}$ has no separate set, i.e., $G^{(t)}$ is a complete graph. With this convention, we formally give Def. 1 and Problem 1.

Definition 1. *A $TCO(V, T, Int, E)$ is **k-topic-connected** if for any $t \in T$, $G^{(t)} = (V^{(t)}, E^{(t)})$ is either* (1) *k-connected or* (2) *a clique if $|V^{(t)}| \leq k$. We denote a **k-topic-connected overlay** by $kTCO(V, T, Int, E)$ (or $kTCO$).*

Problem 1. *The MinAvg-kTCO(V, T, Int) problem parameterized by an integer k is defined as: Given a set of nodes V, a set of topics T, and the interest function Int, construct a $kTCO$ that has the least possible total number of edges, i.e., the minimum average node degree.*

For brevity, we often omit "parameterized by k" and just refer to the problem as MinAvg-kTCO. The MinAvg-TCO problem is the base case of MinAvg-kTCO where $k = 1$. We have the Greedy Merge (GM) algorithm for MinAvg-TCO [9]. The GM algorithm starts with $TPSO(V, T, Int, E)$ where $E = \emptyset$ and proceeds by iteratively adding edges to E until topic-connectivity is attained. At each iteration, GM greedily selects an edge e with the highest GM-*edge-contribution*, which is defined as the number of *TC-components* reduced if an edge e is added to the current overlay. The GM algorithm achieves a logarithmic approximation ratio, which is the lowest among all known polynomial-time algorithms. We use GM as the baseline for developing, analyzing, and evaluating new algorithms for the more generalized problem of MinAvg-kTCO.

We summarize the complexity analysis of the MinAvg-kTCO problem in Theorem 1. The proof is in Appx. A.

Theorem 1. *For any given positive integer k, the MinAvg-kTCO problem parameterized by k is NP-complete and can not be approximated in polynomial time within a factor of $O(\log |V|)$ unless $P = NP$.*

V. The GM2 Algorithm to build $2TCO$

For the MinAvg-2TCO problem, we devise Greedy Merge for the $2TCO$ algorithm, GM2 for short. Although GM2 is structurally similar to the GM and other existing centralized algorithms that build TCO [9], [10], [11], GM2 uses a principally different *progress measure* (see Line 5 of Alg. 1), which we will elaborate upon §VI.

Given a $TPSO(V, T, Int, E)$, the *2-topic-connected component* on topic $t \in T$, is a maximal 2-connected subgraph induced on topic t (i.e., it is not contained in any larger 2-connected subgraph induced on t). We also call it *topic-biconnected component* or *topic-connected block*, *TC-block* for short. Thus, each *TC-block* on $t \in T$ is either a maximal 2-topic-connected subgraph, a bridge (including its endpoints), or an isolated node in $G^{(t)}$. Also,

every such subgraph is a *TC-block* in $G^{(t)}$. Due to their maximality property, different *TC-blocks* on $t \in T$ overlap in at most one node in $G^{(t)}$. Hence, every edge $e \in E^{(t)}$ lies in a unique *TC-block* on t in $G^{(t)}$.

As specified in Alg. 1, GM*2* starts with the overlay $TPSO(V, T, Int, E)$ where $E = \emptyset$, so that there are $\big|\{v|Int(v,t)\}\big|$ singleton *TC-blocks* for each topic $t \in T$. The total number of *TC-blocks* at the start is

$$B_{start} = \sum_{t \in T} \big|\{v \in V | Int(v,t)\}\big| = O(|V||T|) \ . \quad (1)$$

The algorithm carefully adds an edge to E iteration by iteration until $TPSO(V, T, Int, E)$ contains at most one *TC-block* for each $t \in T$, i.e., *2*-topic-connected, and the total number of *TC-blocks* at the end is reduced to

$$B_{end} = \big|\{t \in T | \exists v \in V \text{ s.t. } Int(v,t) = \textit{true}\}\big| \ . \quad (2)$$

Alg. 1 The GM*2* algorithm for *2TCO*

GM*2*(V, T, Int)
Input: V, T, Int
Output: A *2*-topic-connected overlay $2TCO(V, T, Int, E_{\mathsf{GM}2})$
1: $E_{\mathsf{GM}2} \leftarrow \emptyset$
2: $E_{pot} \leftarrow V \times V$
3: **while** $TPSO(V, T, Int, E_{\mathsf{GM}2})$ is not *2*-topic-connected **do**
4: **for all** $e = (v, w) \in E_{pot}$ **do**
5: $estimate(e, E_{\mathsf{GM}2}) \leftarrow \big|\{t \in T | Int(v,t) \wedge Int(w,t) \wedge$ no *TC-block* in $G^{(t)}$ contains both v and $w\}\big|$
6: $e \leftarrow$ find e s.t. $estimate(e, E_{\mathsf{GM}2})$ is maximum among E_{pot}
7: $E_{\mathsf{GM}2} \leftarrow E_{\mathsf{GM}2} \cup \{e\}$
8: $E_{pot} \leftarrow E_{pot} - \{e\}$
9: return $2TCO(V, T, Int, E_{\mathsf{GM}2})$

Lemma 1 summaries the correctness and running time of Alg. 1. We provide the proof in Appx. B.

Lemma 1. *Alg. 1 outputs a 2TCO with time complexity* $O(|V|^4|T|)$.

VI. Approximation Ratio of GM*2*

While both, GM*2* and GM, are greedy algorithms employing similar heuristics, the analysis of GM*2* is much more complex as compared to that of GM [9]. The crux lies in the measure of *progress* each algorithm employs – namely, a quantity that strictly decreases (or increases) with every edge addition up to an absolute limit. The limit can be used to bound the number of edges produced by the algorithm. For example, GM defines the progress measure to construct *1TCO* as the number of *TC-components* in the resulting overlay. The number of *TC-components* decreases every time GM adds an edge, and the number of *TC-components* is an integer-valued function on the current edge set. Based on the well-defined *progress measure*, the key is to establish a lower bound on the optimum – since the optimum solution must cover a complete round of *progress* (i.e., attaining full topic-connectivity), it needs at least a certain number of edges.

Unfortunately, the techniques for GM do not directly apply to the design of GM*2*. We need to overcome three major challenges: (1) find a meaningful measure of progress towards *2TCO* (see §VI-A), (2) estimate the progress as the algorithm proceeds (see §VI-B), and (3) compare the output to the unknown optimum (see §VI-C).

A. Progress measure towards the construction of 2TCO

Probably the most natural progress measure to construct *2TCO* would be the number of *TC-blocks*. However, the number of *TC-blocks* does not always decrease when we add an edge at each iteration. Suppose some algorithm adds edges one by one as illustrated in Fig. 1, i.e., e_i is added in the i-th iteration. The addition of edge e_2, e_3, or e_4 does not decrease the number of *TC-blocks* – the number of *TC-blocks* remains 4. Adding edge e_5 leads to a reduction from 4 to 1 in the number of *TC-blocks*, but not all of them should be accredited to e_5.

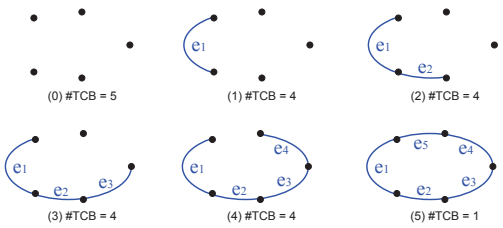


Fig. 1: An edge sequence forms *2TCO* on t.

The progress toward *2TCO* is *amortized* over a sequence of edges added. We can still use the number of *TC-blocks* as a rough progress measure, but the task of comparing to the (unknown) optimum is more difficult. We will use a more subtle calculation (or estimation) of the progress measure that captures each edge contribution.

To present our progress measure we adopt the notation of sequences. Given an instance of the MinAvg-*2*TCO problem $I(V, T, Int)$, we look at an edge set $E \subseteq V \times V$. An **edge sequence**, $\mathbb{E}$, is an ordered list of edges in E, denoted by $\mathbb{E} = \langle e_1, e_2, ..., e_m \rangle$, where each edge $e_i (1 \leq i \leq m)$ is distinct. The *length* of an edge sequence $\mathbb{E}$, denoted by $|\mathbb{E}|$, is the number of ordered edges in $\mathbb{E}$. So $|E| = |\mathbb{E}|$. An edge sequence $\mathbb{F}$ is a *subsequence* of $\mathbb{E}$, if $\mathbb{F}$ can be be derived from $\mathbb{E}$ by deleting some edges without changing the order of the remaining edges.

Suppose E is the output edge set of some algorithm $\mathcal{A}$ that adds edges iteratively one by one and produces a $2TCO$. The **$\mathcal{A}$-edge-sequence**, $\mathbb{E} = \langle e_1, e_2, ..., e_m \rangle$, indicates that $\mathcal{A}$ adds e_i in the i-th iteration. Given an edge $e \in E$, $ind_{\mathbb{E}}(e)$, is the sequence index of e in $\mathbb{E}$, i.e., the iteration number of adding e in algorithm $\mathcal{A}$.

Consider $G^{(t)} = (V^{(t)}, E^{(t)})$ induced on topic $t \in T$, $\mathbb{E}^{(t)}$ is a subsequence of $\mathbb{E}$ that keeps the linear ordering of edge additions of $\mathcal{A}$. $G^{(t)} = (V^{(t)}, E^{(t)})$ is 2-connected, which is equivalent to say that $G^{(t)}$ admits an *ear decomposition* [27]. Below, we adopt some additional concepts from graph theory and provide formal definitions for their use in our context, including the ear decomposition.

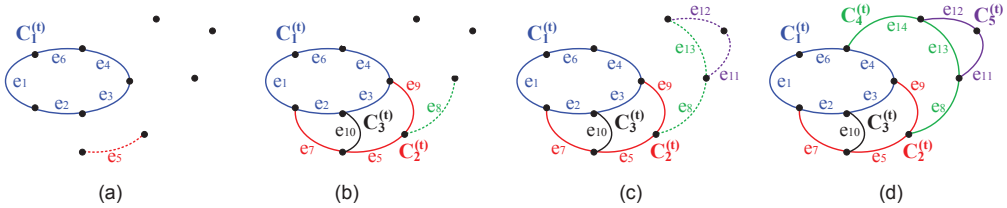
Definition 2. *A **path** in a graph $G^{(t)} = (V^{(t)}, E^{(t)})$ is a nonempty set of edges $Y \subseteq E^{(t)}$ such that (1) edges in Y can be linearly ordered as an edge sequence $\mathbb{Y} = \langle (v_0, v_1), ..., (v_{n-1}, v_n) \rangle$ to connect a sequence of nodes $\mathbb{X} = \langle v_0, ..., v_n \rangle$, and (2) $v_0, ..., v_{n-1}$ are distinct and $v_1, ..., v_n$ are distinct. We call v_0 and v_n the* terminal nodes *of the path Y and the other nodes $v_1, ..., v_{n-1}$ (which may not exist) are* internal nodes. *A* closed *path Y with $v_0 = v_n$ is called a **cycle**, otherwise Y is* noncyclic.

Definition 3. *Given $G^{(t)} = (V^{(t)}, E^{(t)})$ and a nonempty subset of edges $P^{(t)} \subseteq E^{(t)}$, let $W^{(t)} = \{v \in V^{(t)} | \exists e \in P^{(t)}$ s.t. e is incident to $v\}$. An $P^{(t)}$**-ear** in $G^{(t)}$, denoted by $C^{(t)}$, is a noncyclic path in $G^{(t)}$ such that the two terminal nodes are in $W^{(t)}$ and the internal nodes are in $\left(V^{(t)} \backslash W^{(t)}\right)$. The length of the ear $C^{(t)}$ is the number of edges in the path, which we denote as $|C^{(t)}|$. A* trivial *ear contains only one edge. We define the* sequence index *of $C^{(t)}$ with regards to $\mathbb{E}$ as $ind_{\mathbb{E}}(C^{(t)}) = \max\{ind_{\mathbb{E}}(e) | e \in C^{(t)}\}$.*

$G^{(t)}$ contains at least one cycle, otherwise it is not 2-connected. $\mathcal{A}$ adds edges one by one according to $\mathbb{E}$, so one cycle would at first be formed in $G^{(t)}$. This cycle has the minimum sequence index with regards to $\mathbb{E}$. Moreover, given $\mathbb{E}$, we can construct a corresponding ear decomposition for $G^{(t)}$.

Definition 4. *The **$\mathbb{E}$-ear-decomposition on topic** $t \in T$, denoted by $\mathbf{D}^{(t)} = \left[C_1^{(t)}, ..., C_z^{(t)}\right]$, is a partition of $G^{(t)} = (V^{(t)}, E^{(t)})$ into an ordered collection of edge-disjoint paths $C_1^{(t)}, ..., C_z^{(t)}$, such that:*

- ▷ *$S_1^{(t)} = C_1^{(t)}$ is the cycle in $G^{(t)}$ with the minimum sequence index with regards to $\mathbb{E}$.*
- ▷ *For all $1 \leq j \leq z$, let $S_j^{(t)} = C_1^{(t)} \cup ... \cup C_j^{(t)}$, then $C_j^{(t)}$ is the $S_{j-1}^{(t)}$-ear with the shortest length among all $S_{j-1}^{(t)}$-ears that have the minimum sequence index with regards to $\mathbb{E}$. In other words, if C' is any other $S_{j-1}^{(t)}$-ear in $G^{(t)}$, then either* (1) *$ind_{\mathbb{E}}(C_j^{(t)}) < ind_{\mathbb{E}}(C')$ or* (2) *$\left(ind_{\mathbb{E}}(C_j^{(t)}) = ind_{\mathbb{E}}(C') \wedge |C_j^{(t)}| \leq |C'|\right)$.*
- ▷ *$S_z^{(t)} = \bigcup_{j=1}^{z} C_j^{(t)} = E^{(t)}$*



$C_j^{(t)}$	$\lvert C_j^{(t)} \rvert$	number of *TC-blocks*	reduction
$\emptyset$	0	$B(V^{(t)}, \emptyset) = 10$	N/A
$C_1^{(t)}$	5	$B(V^{(t)}, S_1^{(t)}) = 6$	4
$C_2^{(t)}$	3	$B(V^{(t)}, S_2^{(t)}) = 4$	2
$C_3^{(t)}$	1	$B(V^{(t)}, S_3^{(t)}) = 4$	0
$C_4^{(t)}$	3	$B(V^{(t)}, S_4^{(t)}) = 2$	2
$C_5^{(t)}$	2	$B(V^{(t)}, S_5^{(t)}) = 1$	1

Fig. 2: Example of the $\mathbb{E}$-ear-decomposition on topic t: **(a)** The initial cycle $C_1^{(t)}$ is formed after adding e_6. **(b)** $C_2^{(t)}$ is formed after adding e_9, and $C_3^{(t)}$ is formed after adding e_{10}. Note that $C_3^{(t)}$ is *trivial* because it contains only one edge e_{10}. **(c)** The addition of e_{13} forms a cycle $\{e_{11}, e_{12}, e_{13}\}$, but this cycle does not make a $S_3^{(t)}$-ear as required in Def. 4. **(d)** $C_4^{(t)}$ and $C_5^{(t)}$ are formed after adding e_{14}. The addition of edge e_{14} makes two $S_3^{(t)}$-ears: $\{e_8, e_{13}, e_{14}\}$ and $\{e_8, e_{11}, e_{12}, e_{14}\}$. By Def. 4, we set $C_4^{(t)}$ by $\{e_8, e_{13}, e_{14}\}$ because it has the shortest length, and consequently $C_5^{(t)}$ is $\{e_{11}, e_{12}\}$.

Def. 4 serves as the basis to define the progress measure of algorithm $\mathcal{A}$. We consider the number of *TC-blocks* reduced by adding all edges in each ear. Given $P^{(t)} \subseteq E^{(t)}$, we denote by $B(V^{(t)}, P^{(t)})$ the number of *TC-blocks* in the subgraph $(V^{(t)}, P^{(t)})$ of $G^{(t)} = (V^{(t)}, E^{(t)})$. As illustrated in the table on the right hand side of Fig. 2, the number of *TC-blocks* on topic t reduced by adding $C_j^{(t)}$ is $|C_j^{(t)}| - 1$, where $1 \leq j \leq z = 5$. Formally, we generalize these observations as Claim 1, where we define $S_0^{(t)} = C_0^{(t)} = \emptyset$. We prove it inductively in Appx. C.

Claim 1. *The ear* $C_j^{(t)}$ *reduces the number of TC-blocks on topic* $t \in T$ *in* $\left(V^{(t)}, S_{j-1}^{(t)}\right)$ *by* $|C_j^{(t)}| - 1$, *i.e.,*

$$B\left(V^{(t)}, S_{j-1}^{(t)}\right) - B\left(V^{(t)}, S_j^{(t)}\right) = |C_j^{(t)}| - 1 \ , \ \forall \, C_j^{(t)} \text{ in } \mathbf{D}^{(t)} = \left[C_1^{(t)}, ..., C_z^{(t)}\right] \ . \tag{3}$$

As each ear $C_j^{(t)}$ is formed, edges in this ear account for the reduction in the number of *TC-blocks*. It is natural to distribute the reduction over all edges in the newly formed ear: Each edge $e \in C_j^{(t)}$ contributes $\frac{|C_j^{(t)}|-1}{|C_j^{(t)}|}$ to the reduction of *TC-blocks*. The intuitive meaning of the ratio $\frac{|C_j^{(t)}|-1}{|C_j^{(t)}|}$ is the *amortized* contribution of e toward 2-connectivity on topic t. Furthermore, each edge in $E^{(t)}$ belongs to only one ear in the $\mathbb{E}$-ear-decomposition on topic $t \in T$. So, given $\mathbb{E}$, we can define the *edge contribution* of $e \in C_j^{(t)}$ on topic t based on Def. 4 and Claim 1:

$$contrib^{(t)}(e, \mathbb{E}) = \frac{|C_j^{(t)}| - 1}{|C_j^{(t)}|}, \text{ where } e \in C_j^{(t)} \ . \tag{4}$$

The overall *edge contribution* is defined as $$contrib\,(e, \mathbb{E}) = \sum\nolimits_t contrib^{(t)}(e, \mathbb{E}) \ . \tag{5}$$

Furthermore, we define potential function $\Phi(i, \mathbb{E})$ as the progress measure for $\mathcal{A}$ after adding the i-th edge of $\mathbb{E}$:

$$\Phi(i, \mathbb{E}) = B_{start} - \sum_{j=1}^{i} contrib(e_j, \mathbb{E}), 0 \le i \le m \ . \tag{6}$$

$\Phi(0, \mathbb{E}) = B_{start}$, $\Phi(i, \mathbb{E})$ monotonously decreases as i increases, and in the end $\Phi(m, \mathbb{E}) = B_{end}$ based on Claim 1.

B. The estimate of edge contribution

With the edge contribution and potential function defined in Eq. (4), (5) and (6), we could accurately tell the progress of algorithm $\mathcal{A}$ at each iteration – if we knew the output sequence of $\mathcal{A}$. Unfortunately, we do not have the output sequence until the algorithm returns, which makes the decision at each iteration of the algorithm more difficult. To circumvent this dilemma, we first find the bounds for the contribution of an edge currently considered for addition, with regard to all possible extensions of the edge sequence added up until the current iteration. Next, we use these bounds as a bookkeeping device to estimate each edge contribution.

After the i-th iteration of $\mathcal{A}$, we denote by P_i the set of edges added to the overlay and by $\mathbb{P}_i = \langle e_1, ..., e_i \rangle$ the corresponding edge sequence. Given another edge sequence $\mathbb{Q} = \langle e'_1, ..., e'_{|\mathbb{Q}|} \rangle$ where $e'_j \in (V \times V) \backslash P_i$, $\mathbb{P}_i \diamond \mathbb{Q}$ means $\mathbb{P}$ *concatenates with* $\mathbb{Q}$, i.e., $\mathbb{P}_i \diamond \mathbb{Q} = \langle e_1, ..., e_i, e'_1, ..., e'_{|\mathbb{Q}|} \rangle$. Let $\mathbb{R} = \mathbb{P}_i \diamond \mathbb{Q}$, then $\mathbb{R}$ is an *extension* of $\mathbb{P}_i$. The *extension set* of $\mathbb{P}_i$ is

$$\mathscr{E}(\mathbb{P}_i) = \{\mathbb{R} | \mathbb{R} \text{ is an extension of } \mathbb{P}_i \text{ and produces a } 2TCO \text{ for } I(V, T, Int)\}. \tag{7}$$

Note $\mathbb{E} \in \mathscr{E}(\mathbb{P}_i)$. Given some $\mathbb{R} \in \mathscr{E}(\mathbb{P}_i)$, we analyze the range of $contrib(e, \mathbb{R})$. Looking at topic $t \in T$, let $H_i^{(t)} = \left(V^{(t)}, P_i^{(t)}\right)$ be the current topic-induced subgraph on t produced by $\mathcal{A}$ after the i-th iteration, then $contrib^{(t)}(e, \mathbb{R})$ is the edge contribution of e on topic $t \in T$ with regards to $\mathbb{R}$. Let us consider an edge $e(v, w) \in \left(V^{(t)} \times V^{(t)}\right) \backslash P_i^{(t)}$:

- If there exists some *TC-block* on t that contains both v and w, then e would form a trivial $P_i^{(t)}$-ear of length one. By Eq. (4), $contrib^{(t)}(e, \mathbb{R}) = 0$ (e.g., e_{10} in Fig. 2).
- If no *TC-block* on t contains both v and w (e.g., any edge except e_{10} in Fig. 2), Eq. (4) implies that $contrib^{(t)}(e, \mathbb{R}) > 0$. By Def. 4, the length of the ear containing e in the $\mathbb{R}^{(t)}$-ear-decomposition is at least 2, so $contrib^{(t)}(e, \mathbb{R}) \ge \frac{1}{2}$. Besides, $contrib^{(t)}(e, \mathbb{R})$ has an obvious upper bound of 1.

$$\text{Thus, } \ contrib^{(t)}\left(e(v, w), \mathbb{R}\right) = \begin{cases} 0, \text{ if some block in } \left(V^{(t)}, P_i^{(t)}\right) \text{ contains both } v \text{ and } w \\ \in \left[\frac{1}{2}, 1\right), \text{ otherwise} \end{cases} , \mathbb{R} \in \mathscr{E}(\mathbb{P}_i) \ . \tag{8}$$

Claim 2 shows the contributions of an edge with regards to different sequences. The proof is in Appx. C.

Claim 2. *Given* $\mathbb{P}_i = \langle e_1, ..., e_i \rangle$, $\forall \mathbb{E}, \mathbb{R} \in \mathscr{E}(\mathbb{P}_i)$, $contrib(e_j, \mathbb{E}) \le contrib\,(e_j, \mathbb{R}) \le 2 contrib(e_j, \mathbb{E})$, $1 \le j \le i$.

We now instantiate $\mathcal{A}$ by GM2. Given the current edge set P_i, Line 5 of Alg. 1 defines the *estimate* of e's contribution on topic t:

$$estimate^{(t)}(e(v,w),P_i)=\begin{cases}0, \text{ if some block in } \left(V^{(t)},P_i^{(t)}\right) \text{ contains both } v \text{ and } w\\ 1, \text{ otherwise}\end{cases} \tag{9}$$

The overall *edge estimate* is defined as

$$estimate(e,P_i)=\sum\nolimits_t estimate^{(t)}(e,P_i)\ . \tag{10}$$

Claim 3 gives us the bounds for an edge estimate in terms of the edge contribution. The proof is in Appx. C.

Claim 3. *Given P_i and a corresponding $\mathbb{P}_i$, $\forall \mathbb{R}\in\mathscr{E}(\mathbb{P}_i)$, $contrib(e,\mathbb{R})\le estimate(e,P_i)\le 2contrib(e,\mathbb{R})$.*

C. Comparison against the unknown optimum

We now complete our plan to compare the *2TCO* produced by Alg. 1 to the optimal one.

Lemma 2. *The approximation ratio of Alg. 1 is $O(U+\ln|V||T|)$, where $U=\max\{|V^{(t)}|,t\in T\}$.*

Proof: Given an instance $I(V,T,Int)$, suppose E is the output edge set of Alg. 1 and $\mathbb{E}$ is the GM2-edge-sequence, where $|E|=|\mathbb{E}|=m$. Let E^* be the optimal solution where $|E^*|=m^*$.

Recall that P_i is the edge set added to the overlay after the i-th iteration, and $\mathbb{P}_i=\langle e_1,...,e_i\rangle$ is the corresponding edge sequence. Let $Q_i=E^*-P_i$ and $\mathbb{Q}_i$ be an edge sequence of Q_i with an arbitrary order, i.e., $\mathbb{Q}_i=\langle e_1^*,...,e_{|\mathbb{Q}_i|}^*\rangle$. Let $R_i=P_i\cup Q_i$ where $m_i=|R_i|$, and $\mathbb{P}_i\diamond\mathbb{Q}_i=\langle e_1,...,e_i,e_1^*,...,e_{|\mathbb{Q}_i|}^*\rangle$, which means $\mathbb{P}_i$ *concatenates* $\mathbb{Q}_i$.

Since $E^*\subseteq R_i$, $\mathbb{R}_i$ would produce a *2TCO*, by Eq. (7),

$$\mathbb{R}_i\in\mathscr{E}(\mathbb{P}_i)\ . \tag{11}$$

Adding $\mathbb{Q}_i$ immediately after $\mathbb{P}_i$ reduces the values of potential function from $\Phi(i,\mathbb{R}_i)$ to B_{end}. Since $|Q_i|\le|E^*|$, there exists an edge $e'\in Q_i$ such that

$$contrib(e',\mathbb{R}_i)\ge\frac{\Phi(i,\mathbb{R}_i)-\Phi(m_i,\mathbb{R}_i)}{|E^*|}=\frac{\Phi(i,\mathbb{R}_i)-B_{end}}{m^*}. \tag{12}$$

Line 7 of Alg. 1 specifies the edge selection rule: Always choosing the edge with the highest estimate. At the $(i+1)$-th iteration, Alg. 1 picks e_{i+1} so that

$$\begin{aligned} estimate(e_{i+1},P_i) &\ge estimate(e',P_i) && \text{// by greediness}\\ &\ge contrib(e',\mathbb{R}_i) && \text{// by Claim 3}\\ &\ge \frac{\Phi(i,\mathbb{R}_i)-B_{end}}{m^*} && \text{// by Eq. (12)} \end{aligned} \tag{13}$$

According to Eq. (6), $\Phi(i,\mathbb{R}_i)-2\Phi(i,\mathbb{E})=-B_{start}+\sum_{j=1}^{i}(2contrib(e_j,\mathbb{E})-contrib(e_j,\mathbb{R}_i))$. With Claim 2 and Eq. (11), $\sum_{j=1}^{i}(2contrib(e_j,\mathbb{E})-contrib(e_j,\mathbb{R}_i))\ge 0$. So,

$$\Phi(i,\mathbb{R}_i)\ge 2\Phi(i,\mathbb{E})-B_{start}\ . \tag{14}$$

Further,

$$\begin{aligned} \Phi(i,\mathbb{E})-\Phi(i+1,\mathbb{E})=contrib(e_{i+1},\mathbb{E}) &\ge 1/2\cdot estimate(e_{i+1},P_i) && \text{// by Claim 3}\\ &\ge \frac{\Phi(i,\mathbb{R}_i)-B_{end}}{2m^*} && \text{// by Eq. (13)}\\ &\ge \frac{2\Phi(i,\mathbb{E})-(B_{start}+B_{end})}{2m^*} && \text{// by Eq. (14)} \end{aligned} \tag{15}$$

By Derivation C.1 in Appx. C, $\left(\Phi(i+1,\mathbb{E})-\widetilde{B}\right)\le(1-1/m^*)\left(\Phi(i,\mathbb{E})-\widetilde{B}\right)$, where $\widetilde{B}=\frac{B_{start}+B_{end}}{2}$ (16)

Eq. (16) shows the progression of the potential function value within successive iterations in GM2 as compared to the optimal solution. Based on Eq. (16), we derive the bound on the number of iterations of Alg. 1 (i.e., the number of edges in E) relative to m^*. We take $\left(\Phi(i,\mathbb{E})-\widetilde{B}\right)$ as a function of i, and it decreases as GM2 adds an edge at each iteration. Initially, $\Phi(0,\mathbb{E})-\widetilde{B}=\frac{B_{start}-B_{end}}{2}>0$, and finally, $\Phi(m,\mathbb{E})-\widetilde{B}=-\frac{B_{start}-B_{end}}{2}<0$. So at some iteration λ_0, the function turns from positive to negative. We have a sequence of the function values as:

$$\Big\langle \underbrace{\left(\Phi(1,\mathbb{E})-\widetilde{B}\right),\ldots,\left(\Phi(\lambda_0,\mathbb{E})-\widetilde{B}\right)}_{>0,\text{ denote the number of such elements by }\lambda_0.}, \underbrace{\left(\Phi(\lambda_0+1,\mathbb{E})-\widetilde{B}\right),\ldots,\left(\Phi(m,\mathbb{E})-\widetilde{B}\right)}_{\le 0,\text{ denote the number of such elements by }\lambda_1=m-\lambda_0.}\Big\rangle \tag{17}$$

Further, by Derivation C.2 in Appx. C,

$$\lambda_0\le m^*\cdot O(\ln B_{start})\ , \tag{18}$$

$$\lambda_1\le m^*\cdot O(U+\ln B_{start}),\text{ where } U=\max\{|V(t)|,t\in T\}\ . \tag{19}$$

Therefore, $m=\lambda_0+\lambda_1=m^*\cdot O(U+\ln B_{start})$. ■

VII. The HararyPT Algorithm to Build $kTCO$

With regard to the MinAvg-kTCO problem, we design the Harary-Per-Topic Algorithm (HararyPT) to build the $kTCO$, as specified in Alg. 2.

HararyPT stems from graph theory about vertex connectivity and Harary graphs. Function `buildHarary`$(k, \mathbb{V}^{(t)})$ (Line 3 of Alg. 2) represents the standard procedure to construct the k-connected Harary graph for a given sequence of nodes $\mathbb{V}^{(t)}$. HararyPT invokes `buildHarary`() for each topic $t \in T$ (Lines 2-3). Since the Harary graph $H_{k,n}$ is known to be k-connected with the minimum number of edges $\lceil kn/2 \rceil$ [27], we can derive Lemma 3.

Alg. 2 Harary-Per-Topic for $kTCO$

HararyPT$(I(V, T, Int), k)$
Input: $I(V, T, Int), k$
Output: $kTCO(V, T, Int, E_{\text{HPT}})$
1: $\mathbb{V} \leftarrow$ get an arbitrary sequence for V
2: **for all** $t \in T$ **do**
3: $\quad E^{(t)} \leftarrow$ `buildHarary`$(k, \mathbb{V}^{(t)})$
4: $E_{\text{HPT}} \leftarrow \bigcup_{t \in T} E^{(t)}$
5: return $kTCO(V, T, Int, E_{\text{HPT}})$

Lemma 3. *Alg. 2 produces a $kTCO$ with time complexity* $O\left(k \cdot \sum_{t \in T} |V^{(t)}|\right) = O(k|V||T|)$.

The most straightforward approach is perhaps to build the sub-overlay (i.e., Harary graph) independently for each topic. One serious drawback is that the probability of any two nodes sharing the edge in more than one sub-overlay is small [26]. Thus, the output overlay has an unnecessarily high average node degree. We also evaluate this naive approach in §VIII. In order to promote edge sharing across different sub-overlays, we first obtain a node sequence for all the nodes in Line 1 of Alg. 2. The HararyPT algorithm adopts the same linear ordering for all Harary constructions across all topics. By sharing the determined node sequence, these Harary graphs are likely to converge a lot of edges, especially when the workloads are highly correlated. As a consequence, the output $kTCO$ tends to have a low node degree. Although we do not have an approximation ratio for the HararyPT algorithm, we can assume that subscriptions are highly correlated in typical pub/sub workloads. More specifically, the study of representative pub/sub workloads used in actual applications observes the "Pareto 80-20" rule: Most nodes subscribe to a relatively small number of topics [23]. Besides, many pub/sub workloads are modelled by a power law distribution in both topic popularity and subscription size per node [5], [35]. Our experimental findings in §VIII and Appx. D further demonstrate that the HararyPT algorithm significantly reduces the number of edges while offering a high degree of topic-connectivity for typical pub/sub workloads in practice.

VIII. Evaluation

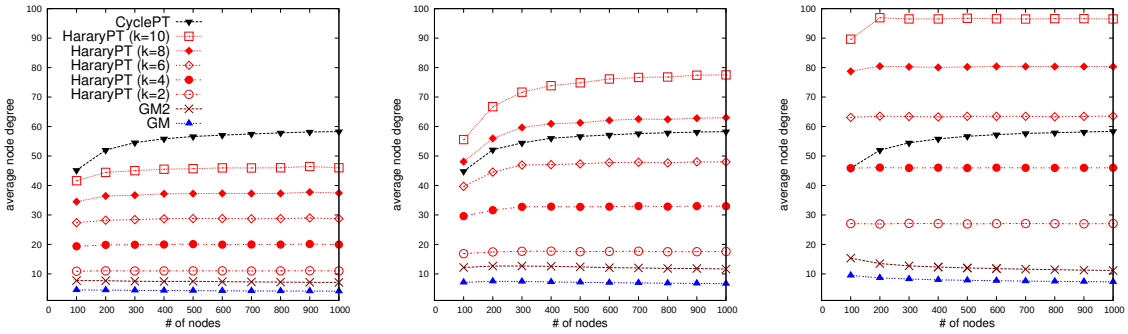
We implemented GM2, HararyPT, and other auxiliary algorithms in Java. We use GM as a baseline, because it produces a $1TCO$ with the lowest average node degree among all known polynomial-time algorithms [9]. We also develop the *Cycle-Per-Topic* algorithm (CyclePT) that mimics the common practice of building a separate overlay for each topic independently (usually a tree but we use a cycle that has the same average node degree and achieves 2-topic-connectivity). By CyclePT, all nodes interested in the same topic form a cycle, and cycles for different topics are merged into a single $2TCO$. Note that CyclePT is fundamentally different from HararyPT, because CyclePT does not exploit the correlations in the workload and often ends up with an overlay with unnecessarily high node degrees. We evaluated the HararyPT algorithm with different parameters, i.e., $k \in [2, 14]$. We only plot the representative results for $k = 2, 4, 6, 8, 10$, but we report additional results.

We mainly compare the average node degrees in the output overlays produced by different algorithms. For a specific algorithm $\mathcal{A}$, we denote by $\overline{d}_{\mathcal{A}}$ the average node degree produced by $\mathcal{A}$. We denote by $T(v)$ the topic set which node v subscribes to, and we call $|T(v)|$ the *subscription size* of node v.

We use the following value ranges as input: $|V| \in [100, 1\,000]$, $|T| \in [100, 1\,000]$, and $|T(v)| \in [10, 100]$, where each node has a fixed the subscription size. Each topic $t \in T$ is associated with probability $p(t)$, $\sum_{t \in T} p(t) = 1$, and each node $v \in V$ subscribes to t with a probability $p(t)$ until $|T(v)|$ is reached. The value of $p(t)$ is distributed according to either an exponential, a Zipfian (with α=2.0), or a uniform distribution, which we call Expo, Zipf, or Unif for short. According to [23], these distributions are representative of actual workloads used in industrial pub/sub systems today. Expo is used by stock-market monitoring engines for the study of stock popularity in the New York Stock Exchange [35], and Zipf faithfully describes the feed popularity distribution in RSS feeds [5].

A. The impact of the number of nodes

Fig. 3 depicts the comparison among GM2, HararyPT, GM, and CyclePT with regards to the number of nodes under different distributions. We set $|T| = 200$, $|T(v)| = 30$, and $|V| \in [100, 1\,000]$.



(a) under Expo (b) under Zipf (c) under Unif

Fig. 3: GM2 vs. HararyPT vs. GM vs. CyclePT wrt. $|V|$

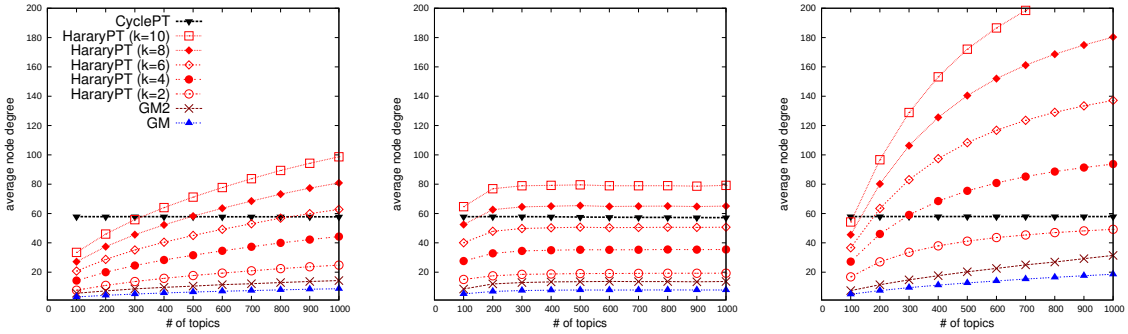
We look at GM2 in Fig. 3. First, $\overline{d}_{\mathsf{GM}2}$ and $\overline{d}_{\mathsf{GM}}$ are quite close under all conditions. More specifically, $\overline{d}_{\mathsf{GM}2}$ is smaller than $1.66 \cdot \overline{d}_{\mathsf{GM}}$ on average, across all three distributions. GM2 is capable of constructing a $2TCO$ with a marginal increase in the average node degree as compared to $1TCO$ produced by GM. Second, $\overline{d}_{\mathsf{CyclePT}}$ is roughly equal to twice the subscription size, which is about 5-times higher than $\overline{d}_{\mathsf{GM}2}$ on average. Third, $\overline{d}_{\mathsf{CyclePT}}$ tends to increase with the number of nodes, while both $\overline{d}_{\mathsf{GM}2}$ and $\overline{d}_{\mathsf{GM}}$ decrease as the number of nodes scales up. The decrease of $\overline{d}_{\mathsf{GM}2}$ and $\overline{d}_{\mathsf{GM}}$ lies in the fact that increasing the number of nodes leads to higher chances for both GM2 and GM to find neighbors with more interest overlap, thus reducing overall number of neighbors needed. All these results demonstrate the scalability of GM2 with regards to the number of nodes.

We look at HararyPT in Fig. 3. First, GM2 outperforms HararyPT both theoretically and empirically for constructing $2TCO$s. However, HararyPT allows the overlay to have more reliability commitments over $2TCO$ by setting $k > 2$. Second, $\overline{d}_{\mathsf{HararyPT}}$ increases as we increase k under all three distributions, and the HararyPT algorithm tends to output better average node degrees in more skewed distributions. Third, HararyPT significantly reduces unnecessary redundancy as compared to CyclePT. With fewer edges than $2TCO$s produced by CyclePT, HararyPT can achieve $12\,TCO$ under Expo, $7TCO$ under Zipf, and $5TCO$ under Unif.

Please see other evaluation analysis for HararyPT in Appx. D. The rest of §VIII places more emphasis on GM2.

B. The impact of the number of topics

Fig. 4 depicts how GM2 and HararyPT perform as compared to GM and CyclePT when the number of topics varies under different topic popularities. We set $|V| = 800$, $|T(v)| = 30$, and $|T| \in [100, 1\,000]$.



(a) under Expo (b) under Zipf (c) under Unif

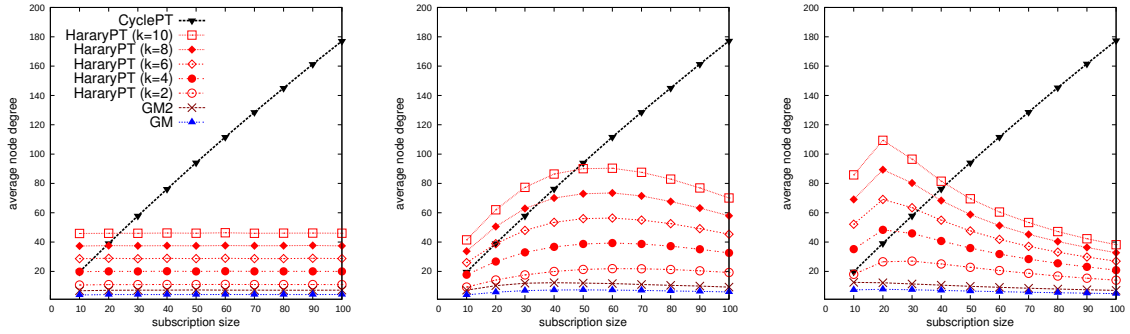
Fig. 4: GM2 vs. HararyPT vs. GM vs. CyclePT wrt. $|T|$

Referring to the GM2 algorithm in Fig. 4, it can be seen that the average node degrees of both GM2 and GM increase with the number of topics. Note that increasing the number of topics leads to reduced correlation, i.e., the probability of having two nodes interested in the same topic drops as the number of topics increases, and with reduced correlation the edge contribution at each iteration of GM2 (and GM) tends to be lower. This reduction in the correlation is more pronounced for Unif as compared to skewed distributions, like Expo or Zipf. Yet, the increase in $\overline{d}_{\mathsf{GM}2}$ is slow paced. In particular, $\overline{d}_{\mathsf{GM}2}$ is no more than $1.66 \cdot \overline{d}_{\mathsf{GM}}$ on average. Next, the gap between

$\overline{d}_{\mathsf{CyclePT}}$ and $\overline{d}_{\mathsf{GM}2}$ remains significant: $\overline{d}_{\mathsf{CyclePT}} - \overline{d}_{\mathsf{GM}2} \geq 43.1$ on average, across all experiment instances under various distributions. Besides, GM*2* exhibits more advantages over CyclePT for highly correlated workloads.

C. The impact of subscription size

Fig. 5 depicts the impact of the subscription size on the GM*2*, HararyPT, GM, and CyclePT algorithms. We fix $|V| = 800$, $|T| = 200$, and $|T(v)| \in [10, 100]$.



(a) under Expo (b) under Zipf (c) under Unif

Fig. 5: GM*2* vs. HararyPT vs. GM vs. CyclePT wrt. the subscription size

We focus on the GM*2* algorithm in Fig. 5. First, under all three distributions, GM*2* and GM produce quite close overlays in terms of the average node degrees. As the subscription size increases, both $\overline{d}_{\mathsf{GM}2}$ and $\overline{d}_{\mathsf{GM}}$ decrease, and the difference $(\overline{d}_{\mathsf{GM}2} - \overline{d}_{\mathsf{GM}})$ shrinks. This decrease occurs because the growth of subscription size causes increased correlation across the nodes. Upon bigger correlation, an edge addition to the overlay has on average a higher contribution toward $1TCO$ (or $2TCO$) because the nodes share more comment interests. Therefore, a smaller number of edges are needed until the overlay becomes topic-(bi)connected. Second, the average node degree of CyclePT increases linearly with the subscription size, $\overline{d}_{\mathsf{CyclePT}}$ is roughly equal to twice the subscription size of the workload. The above two facts render GM*2* increasingly important when the subscription size scales up.

D. Topic diameter of the overlay

We also look at *topic diameters* in the output overlays. Given $2TCO(V, T, Int, E)$, the topic diameter for $t \in T$ is $diam^{(t)} = diam(G^{(t)})$, where $diam(G^{(t)})$ is the maximum shortest distance between any two nodes in $G^{(t)} = \left(V^{(t)}, E^{(t)}\right)$. We denote the maximum and average topic diameter across all topics as $Diam$ and $\overline{diam}$, respectively. Fig. 6 shows that GM*2* significantly outperforms GM in terms of both $Diam$ and $\overline{diam}$. Under Unif, $Diam_{\mathsf{GM}2}$ is $0.40 \cdot Diam_{\mathsf{GM}}$, and $\overline{diam}_{\mathsf{GM}2}$ is $0.50 \cdot \overline{diam}_{\mathsf{GM}}$, on average. Besides, the gap grows as the input instances scale up from 100 nodes to 1000: as $|V| = 1000$, $Diam_{\mathsf{GM}} - Diam_{\mathsf{GM}2} = 17.5$, and $\overline{diam}_{\mathsf{GM}} - \overline{diam}_{\mathsf{GM}2} = 10.0$. (Additional results under Expo and Zipf are available in Appx. D-C.)

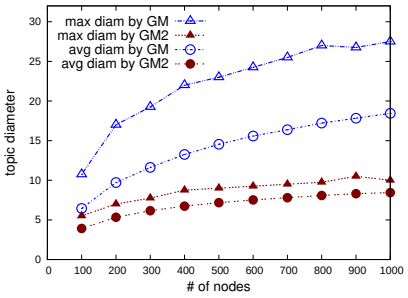


Fig. 6: Topic diameters under Unif

IX. Conclusion

We study a new family of optimization problems MinAvg-kTCO that constructs reliable overlay networks for topic-based pub/sub. We present a polynomial-time overlay design algorithm, GM*2*, which approximates MinAvg-*2*TCO within a proven bound. We provide a novel proof for the approximation ratio of GM2, which is almost tight since no logarithmic approximation polynomial-time algorithm can exist for the MinAvg-*2*TCO problem unless P=NP. Furthermore, we design a heuristic algorithm for the MinAvg-kTCO problem, namely the HararyPT algorithm, especially for highly correlated pub/sub workloads.

Our experimental results validate our formal analysis for the GM*2* algorithm: The average node degree of the $2TCO$ produced by GM*2* is about 1.65 times that of the $1TCO$ generated by the baseline algorithm, GM. We also show the practical scalability of HararyPT for representative pub/sub workloads in terms of the number of nodes, the number of topics, and the subscription size. In sum, our designed algorithms are capable of achieving more reliable topic-connectivity by compromising the average node degrees insignificantly.

REFERENCES

[1] J. Reumann, "Pub/Sub at Google," Lecture & Personal Communications at EuroSys & CANOE Summer School, Oslo, Norway, Aug'09.

[2] B. F. Cooper, R. Ramakrishnan, U. Srivastava, A. Silberstein, P. Bohannon, H.-A. Jacobsen, N. Puz, D. Weaver, and R. Yerneni, "PNUTS: Yahoo!'s hosted data serving platform," *Proc. VLDB Endow.*, 2008.

[3] A. Adya, J. Dunagan, and A. Wolman, "Centrifuge: integrated lease management and partitioning for cloud services," in *NSDI'10*.

[4] "TIBCO Rendezvous," http://www.tibco.com.

[5] H. Liu, V. Ramasubramanian, and E. G. Sirer, "Client behavior and feed characteristics of RSS, a publish-subscribe system for web micronews," in *IMC'05*.

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

[7] M. Sadoghi, M. Labrecque, H. Singh, W. Shum, and H.-A. Jacobsen, "Efficient event processing through reconfigurable hardware for algorithmic trading," *Proc. VLDB Endow.'10*.

[8] B. Koldehofe, F. Dürr, M. A. Tariq, and K. Rothermel, "The Power of Software-defined Networking: Line-rate Content-based Routing Using OpenFlow," in *Proceedings of the 7th MW4NG Workshop of the 13th International Middleware Conference*. ACM, 2012, Workshop Paper, pp. 1–6.

[9] G. Chockler, R. Melamed, Y. Tock, and R. Vitenberg, "Constructing scalable overlays for pub-sub with many topics: Problems, algorithms, and evaluation," in *PODC'07*.

[10] M. Onus and A. W. Richa, "Minimum maximum degree publish-subscribe overlay network design," in *INFOCOM'09*.

[11] ——, "Parameterized maximum and average degree approximation in topic-based publish-subscribe overlay network design," in *ICDCS'10*.

[12] M. A. Jaeger, H. Parzyjegla, G. Mühl, and K. Herrmann, "Self-organizing broker topologies for publish/subscribe systems," in *SAC'07*.

[13] R. Baldoni, R. Beraldi, L. Querzoni, and A. Virgillito, "Efficient publish/subscribe through a self-organizing broker overlay and its application to SIENA," *Comput. J.*, vol. 50, no. 4, 2007.

[14] C. Chen, H.-A. Jacobsen, and R. Vitenberg, "Divide and conquer algorithms for publish/subscribe overlay design," in *ICDCS'10*.

[15] C. Chen, R. Vitenberg, and H.-A. Jacobsen, "Scaling construction of low fan-out overlays for topic-based publish/subscribe systems," in *ICDCS'11*.

[16] D. Yuan, S. Park, P. Huang, Y. Liu, M. M. Lee, X. Tang, Y. Zhou, and S. Savage, "Be conservative: enhancing failure diagnosis with proactive logging," in *OSDI*, 2012, pp. 293–306.

[17] "Google Cluster Data." [Online]. Available: http://code.google.com/p/googleclusterdata/

[18] M. Li, F. Ye, M. Kim, H. Chen, and H. Lei, "A scalable and elastic publish/subscribe service," in *IPDPS '11*.

[19] R. K. Ganti, N. Pham, H. Ahmadi, S. Nangia, and T. F. Abdelzaher, "Greengps: a participatory sensing fuel-efficient maps application," in *MobiSys '10*.

[20] R. Meier and V. Cahill, "Steam: Event-based middleware for wireless ad hoc network," in *ICDCSW'02*.

[21] G. P. Picco, G. Cugola, and A. L. Murphy, "Efficient content-based event dispatching in the presence of topological reconfiguration," in *ICDCS'03*.

[22] C. Chen, R. Vitenberg, and H.-A. Jacobsen, "A generalized algorithm for publish/subscribe overlay design and its fast implementation," in *DISC*, 2012, pp. 76–90.

[23] G. Chockler, R. Melamed, Y. Tock, and R. Vitenberg, "Spidercast: A scalable interest-aware overlay for topic-based pub/sub communication," in *DEBS'07*.

[24] S. Girdzijauskas, G. Chockler, Y. Vigfusson, Y. Tock, and R. Melamed, "Magnet: practical subscription clustering for internet-scale publish/subscribe," in *DEBS'10*.

[25] V. Setty, M. van Steen, R. Vitenberg, and S. Voulgaris, "Poldercast: Fast, robus, and scalable architecture for p2p topic-based pub/sub," in *Middleware'12*.

[26] M. Matos, A. Nunes, R. Oliveira, and J. Pereira, "Stan: exploiting shared interests without disclosing them in gossip-based publish/subscribe," in *IPTPS'10*.

[27] D. B. West, *Introduction to Graph Theory*, 2nd ed. Prentice Hall, 2000.

[28] D. Liben-Nowell, H. Balakrishnan, and D. Karger, "Analysis of the evolution of peer-to-peer systems," in *PODC'02*.

[29] E. De Santis, F. Grandoni, and A. Panconesi, "Fast low degree connectivity of ad-hoc networks via percolation," in *ESA'07*.

[30] L. C. Lau, J. S. Naor, M. R. Salavatipour, and M. Singh, "Survivable network design with degree or order constraints," in *Proc. ACM STOC'07*.

[31] R. Baldoni, R. Beraldi, V. Quema, L. Querzoni, and S. Tucci-Piergiovanni, "TERA: topic-based event routing for peer-to-peer architectures," in *DEBS'07*.

[32] M. Castro, P. Druschel, A.-M. Kermarrec, and A. Rowstron, "SCRIBE: A large-scale and decentralized application-level multicast infrastructure," *JSAC*, 2002.

[33] E. Baehni, P. Eugster, and R. Guerraoui, "Data-aware multicast," in *DSN'04*.

[34] R. Chand and P. Felber, "Semantic peer-to-peer overlays for publish/subscribe networks," in *EUROPAR'05*.

[35] Y. Tock, N. Naaman, A. Harpaz, and G. Gershinsky, "Hierarchical clustering of message flows in a multicast data dissemination system," in *IASTED PDCS*, 2005.

[36] J. Gross and J. Yellen, *Graph theory and its applications*. Boca Raton, FL, USA: CRC Press, Inc., 2005.

[37] D. Angluin, J. Aspnes, and L. Reyzin, "Inferring social networks from outbreaks," in *Algorithmic Learning Theory, 21st International Conference, ALT 2010*, pp. 104–118.

[38] "Prime number theorem." [Online]. Available: http://en.wikipedia.org/wiki/Prime_number_theorem

APPENDIX A
COMPLEXITY OF THE PARAMETERIZED MINAVG-kTCO PROBLEM

Theorem A.1. *MinAvg-2TCO is NP-complete.*

Proof: First, MinAvg-2TCO is in NP. We consider the decision version of MinAvg-2TCO: Each instance of the problem has $I(V, T, Int)$ and a constant m_0 We need to answer the question of whether there exists a $2TCO$ such that the number of edges is no more than m_0, i.e., $\leq m_0$. If we are given a candidate overlay $TPSO(V, T, Int, E)$, we can verify in polynomial time (1) whether this candidate is $2TCO$ by computing the *blocks* of each sub-graph $G(t)$ where $t \in T$ [27], [36], and (2) whether $|E| \leq m_0$.

Second, we prove MinAvg-2TCO is NP-hard by a reduction from MinAvg-TCO to MinAvg-2TCO. We look at the decision versions for both problems. Given an instance $I(V, T, Int, m)$ for the decision version of MinAvg-TCO, we need to give a yes/no answer to the question of whether there exists $TCO(V, T, Int, E)$ such that $|E| \leq m$. Without loss of generality, we can denote $|V| = n$ and number all nodes as $V = \{v_1, v_2, ..., v_n\}$. We construct an instance $I'(V', T', Int', m')$ for the decision version of MinAvg-2TCO, which asks whether there exists $2TCO'(V', T', Int', E')$ such that $|E'| \leq m'$.

The construction can be achieved in polynomial time as follows (see Fig. 7):

- V' includes all nodes in V and adds a new node v', i.e., $V' = V \cup \{v'\} = \{v_1, v_2, ..., v_n, v'\}$.
- T' includes all topics in T and adds a topic t'_i for each node $v_i \in V$, i.e., $T' = T \cup \{t'_1, t'_2, ..., t'_n\}$.
- $Int'(v_i, t) = Int(v_i, t), \forall t \in T$; $Int'(v_i, t'_i) =$ ***true***, $Int'(v_i, t'_j) =$ ***false*** if $i \neq j$; $Int'(v', t) =$ ***true***, $\forall t \in T'$.
- $m' = m + n$.

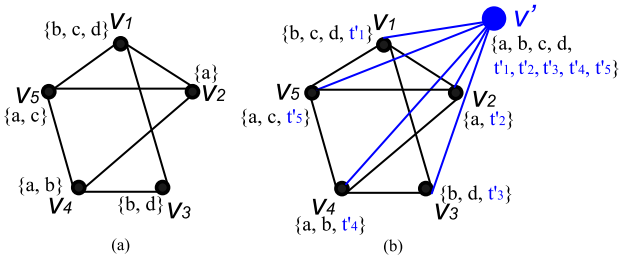


Fig. 7: The construction of a $2TCO$ instance from a $1TCO$ instance: (a) $1TCO(V, T, Int)$; (b) $2TCO'(V', T', Int')$.

Suppose there is some $TCO(V, T, Int, E)$, for instance, I such that $|E| \leq m$, then there is $2TCO'(V', T', Int', E')$, for instance, I' such that $|E'| \leq m' = m + n$. Consider the edge set $E' = E \cup \{(v', v_1), (v', v_2), ..., (v', v_n)|\}$, then $|E'| \leq m + n$ and $G'(V', E')$ satisfies 2-topic-connectivity for $I'(V', T', Int')$, because 1) $G'(V', E')$ will remain topic-connected when removing v' from the graph since the remaining graph is identical to $G(V, E)$; 2) $G'(V', E')$ will remain topic-connected when removing any $v_i \in V$ because v' connects to every other node.

On the other hand, if there is some $2TCO'(V', T', Int, E')$, for instance, I' such that $|E'| \leq m'$, then there is $TCO(V, T, Int, E)$ such that $|E| \leq m = m' - n$. In $2TCO'$, v' has to connect to every other nodes to attain topic-connectivity for the newly introduced topics $\{t'_1, t'_2, ..., t'_n\}$, i.e., E' contains all edges $\{(v', v_1), (v', v_2), ..., (v', v_n)\}$. We construct $E = E' \backslash \{(v', v_1), (v', v_2), ..., (v', v_n)\}$, then $|E| \leq m$ and $G(V, E)$ satisfy topic-connectivity with regards to $I(V, T, Int)$, because $G'(V', E')$ remains topic-connected when removing v' from the graph, which turns out to be $G(V, E)$. ■

The lower bound on the approximability of MinAvg-TCO was proven to be $\Omega(\log |V|)$ unless $P = NP$ [37]. Based on this result, we provide Theorem A.2 for the inapproximability of MinAvg-2TCO.

Theorem A.2. *MinAvg-2TCO can not be approximated in polynomial time within a factor of* $O(\log |V|)$ *unless* $P = NP$.

Proof: We follow the same reduction and notation as presented in the proof of Theorem A.1. Denote m_{opt} as the minimum number of edges for the optimal solution of MinAvg-TCO(V, T, Int) and m'_{opt} as the minimum number of edges for the optimal solution of MinAvg-2TCO(V, T, Int), then

$$m_{opt} = m'_{opt} - n \tag{20}$$

Suppose, by contradiction, that there is a polynomial algorithm which achieves an approximation ratio of $O(\log n)$ for MinAvg-2TCO. Let the output be $2TCO'(V', T', Int', E')$ and $|E'| = m'$, then there exists a constant C s.t.

$$m' \leq m'_{opt} \cdot C \log n \tag{21}$$

With regards to the corresponding $TCO(V,T,Int,E)$ where $|E|=m$, we have

$$\begin{aligned} m = m' - n &\leq m'_{opt} \cdot C\log n - n \\ &= (m_{opt}+n)\cdot C\log n - n \\ &= m_{opt}\cdot C\log n - (n - C\log n) \\ &= m_{opt}\cdot O(\log n) \end{aligned} \tag{22}$$

So, $TCO(V,T,Int,E)$ achieves an approximation ratio of $O(\log|V|)$, which contradicts Theorem 1 in [37]. ■

We further generalize the results for MinAvg-kTCO by induction on k with the same proof techniques.

Theorem A.3. *For any given positive integer k, the MinAvg-kTCO problem parameterized by k is NP-complete and can not be approximated in polynomial time within a factor of $O(\log|V|)$ unless $P=NP$.*

APPENDIX B
CORRECTNESS AND RUNNING TIME OF THE GM2 ALGORITHM

Lemma B.1. *Alg. 1 outputs a $2TCO$ after at most $|V|^2$ iterations of the while loop in Lines 3-8.*

Proof: It follows directly from the pseudo code that Alg. 1 always outputs a $2TCO$. At each iteration of the while loop in Lines 3-8, one edge is added to the current overlay network. Hence, the algorithm terminates in $|E_{\mathsf{GM}2}|$ iterations, which is bounded by $|V|^2$. ■

Lemma B.2. *The running time of Alg. 1 is $O(|V|^4|T|)$.*

Proof: Consider the runtime cost of each iteration in the while loop in Lines 3-8. For each topic $t\in T$, a Depth-First-Search-based algorithm can find all blocks in the current topic-induced sub-overlay with $O(|V|^2)$ [36]. Thus, it takes $O(|V|^2)$ time to compute the *edge estimate* on topic t for each potential edge. The time complexity at each iteration is $O(|V|^2|T|)$ across all topics in T. According to Lemma B.1, the running time of Alg. 1 is $O(|V|^4|T|)$. ■

APPENDIX C
CLAIMS AND BUILDING BLOCKS TO COMPLETE THE PROOF OF GM2'S APPROXIMATION RATIO

Claim C.1. *The ear $C_j^{(t)}$ reduces the number of TC-blocks on topic $t\in T$ in $\left(V^{(t)},S_{j-1}^{(t)}\right)$ by $|C_j^{(t)}|-1$, i.e.,*

$$B\left(V^{(t)},S_{j-1}^{(t)}\right) - B\left(V^{(t)},S_j^{(t)}\right) = |C_j^{(t)}|-1\ ,\ \forall\ C_j^{(t)}\ in\ \mathbf{D}^{(t)} = \left[C_1^{(t)},...,C_z^{(t)}\right]\ .$$

Proof: Given an instance $I(V,T,Int)$ and an edge sequence $\mathbb{E}$ that produces a $2TCO$, we prove this claim by induction on j, the index for the ears in the $\mathbb{E}$-ear-decomposition $\mathbf{D}^{(t)}$.

- *Base case*: $j=1$. When the edge set is $\emptyset$, there are $|V^{(t)}|$ singleton *TC-blocks* in $\left(V^{(t)},\emptyset\right)$, i.e.,

$$B\left(V^{(t)},\emptyset\right) = |V^{(t)}|\ . \tag{23}$$

Consider the first cycle $C_1^{(t)}$ in the $\mathbf{D}^{(t)}$: $C_1^{(t)}=S_1^{(t)}$ has $|C_1^{(t)}|$ edges to connect $|C_1^{(t)}|$ nodes, and all $|C_1^{(t)}|$ nodes belong to one *TC-block* in $\left(V^{(t)},C_1^{(t)}\right)$. Apart from the *TC-block* that is composed of $|C_1^{(t)}|$ nodes, there are $\left(|V^t|-|C_1^{(t)}|\right)$ singleton *TC-blocks* in $\left(V^{(t)},C_0^{(t)}\right)$. Thus the total number of *TC-blocks* in $\left(V^{(t)},S_0^{(t)}\right)$ is $(|V^t|-|C_0^{(t)}|+1)$, i.e.,

$$B\left(V^{(t)},S_1^{(t)}\right) = |V^t| - |C_1^{(t)}| + 1\ . \tag{24}$$

Therefore, edges in $S_1^{(t)}$ belong to one *TC-block* in $\left(V^{(t)},S_1^{(t)}\right)$, and

$$B\left(V^{(t)},\emptyset\right) - B\left(V^{(t)},S_1^{(t)}\right) = |C_1^{(t)}|-1\ . \tag{25}$$

- *Inductive hypothesis*: assume inductively that all edges in $S_r^{(t)}$ belong to one *TC-block* in $\left(V^{(t)}, S_r^{(t)}\right)$, and

$$B\left(V^{(t)}, S_{r-1}^{(t)}\right) - B\left(V^{(t)}, S_r^{(t)}\right) = |C_r^{(t)}| - 1 \ , \ \forall\, r \leq j-1 \ , \text{ where } 1 < j \leq z \ .$$

- *Inductive step*: Based on the inductive hypothesis, all edges in $S_{j-1}^{(t)}$ belong to one *TC-block* in $\left(V^{(t)}, S_{j-1}^{(t)}\right)$ and we denote the node set in this block as

$$W_{j-1}^{(t)} = \{v \in V^{(t)} | \exists\ e \in S_{j-1}^{(t)} \text{ s.t. } e \text{ is incident to } v\} \ .$$

So the number of *TC-blocks* in $\left(V^{(t)}, S_{j-1}^{(t)}\right)$ is

$$B\left(V^{(t)}, S_{j-1}^{(t)}\right) = 1 + \left(|V^{(t)}| - |W_{j-1}^{(t)}|\right) \ . \tag{26}$$

Adding ears preserves 2-connectedness (see the Whitney Theorem in [27]), so all edges in $S_j^{(t)} = S_{j-1}^{(t)} \cup C_j^{(t)}$ belong to one *TC-block* in $\left(V^{(t)}, S_j^{(t)}\right)$, which we denote as

$$W_j^{(t)} = \{v \in V^{(t)} | \exists\ e \in S_j^{(t)} \text{ s.t. } e \text{ is incident to } v\} \ .$$

Similar to Eq. (26), the number of *TC-blocks* in $\left(V^{(t)}, S_j^{(t)}\right)$ is

$$B\left(V^{(t)}, S_j^{(t)}\right) = 1 + \left(|V^{(t)}| - |W_j^{(t)}|\right) \ . \tag{27}$$

The ear $C_j^{(t)}$ has $|C_j^{(t)}|$ edges to connect $\left(|C_j^{(t)}| + 1\right)$ nodes: 2 terminal nodes are in $W_{j-1}^{(t)}$ and $\left(|C_j^{(t)}| - 1\right)$ are in $\left(V^{(t)} \backslash W_{j-1}^{(t)}\right)$. As compared to $W_{j-1}^{(t)}$, $W_j^{(t)}$ contains additional $\left(|C_j^{(t)}| - 1\right)$ nodes from the ear $C_j^{(t)}$, so

$$|W_j^{(t)}| - |W_{j-1}^{(t)}| = |C_j^{(t)}| - 1 \tag{28}$$

Combining Eq. (26), (27) and (28), edges in $S_j^{(t)}$ belong to one *TC-block* in $\left(V^{(t)}, S_j^{(t)}\right)$ and

$$B\left(V^{(t)}, S_{j-1}^{(t)}\right) - B\left(V^{(t)}, S_j^{(t)}\right) = |W_j^{(t)}| - |W_{j-1}^{(t)}| = |C_j^{(t)}| - 1 \ . \tag{29}$$

■

Claim C.2. *Given* $\mathbb{P}_i = \langle e_1, ..., e_i \rangle$, $\forall\, \mathbb{E}, \mathbb{R} \in \mathscr{E}(\mathbb{P}_i)$, $contrib(e_j, \mathbb{E}) \leq contrib\,(e_j, \mathbb{R}) \leq 2 contrib(e_j, \mathbb{E})$, $1 \leq j \leq i$.

Proof: Since $\mathbb{E}, \mathbb{R} \in \mathscr{E}(\mathbb{P}_i)$, by Eq. (8), for any $t \in T$, we have either

$contrib^{(t)}(e_j, \mathbb{E}) = contrib^{(t)}(e_j, \mathbb{R}) = 0$, $1 \leq j \leq i$

or

$contrib^{(t)}(e_j, \mathbb{E}), contrib^{(t)}(e_j, \mathbb{R}) \in [\frac{1}{2}, 1)$, $1 \leq j \leq i$

As a result,

$$contrib^{(t)}(e_j, \mathbb{E}) \leq contrib^{(t)}\,(e_j, \mathbb{R}) \leq 2 contrib^{(t)}(e_j, \mathbb{E}),\ 1 \leq j \leq i,\ \forall\ t \in T \tag{30}$$

Furthermore,

$$\sum_{t \in T} contrib^{(t)}(e_j, \mathbb{E}) \leq \sum_{t \in T} contrib^{(t)}\,(e_j, \mathbb{R}) \leq 2 \sum_{t \in T} contrib^{(t)}(e_j, \mathbb{E}),\ 1 \leq j \leq i \tag{31}$$

By the definition in Eq. 5, we have

$$contrib(e_j, \mathbb{E}) \leq contrib\,(e_j, \mathbb{R}) \leq 2 contrib(e_j, \mathbb{E}),\ 1 \leq j \leq i \tag{32}$$

■

Claim C.3. *Given P_i and its corresponding $\mathbb{P}_i$, $\forall \mathbb{R} \in \mathscr{E}(\mathbb{P}_i)$, $contrib(e, \mathbb{R}) \leq estimate(e, P_i) \leq 2 \cdot contrib(e, \mathbb{R})$.*

Proof: Given $\mathbb{R} \in \mathscr{E}(\mathbb{P}_i)$, we first fix some $t \in T$. Based on the definition in Eq. (9), for any $e \in \left(V^{(t)} \times V^{(t)}\right)$, we have either

$$contrib^{(t)}(e, \mathbb{R}) = estimate^{(t)}(e, P_i) = 0$$

or

$$contrib^{(t)}(e_j, \mathbb{R}) \in \left[\frac{1}{2}, 1\right) \text{ and } estimate^{(t)}(e, P_i) = 1$$

As a result,

$$contrib^{(t)}(e, \mathbb{R}) \leq estimate^{(t)}(e, P_i) \leq 2 \cdot contrib^{(t)}(e, \mathbb{R}),\ \forall\ t \in T, \text{ where } \mathbb{R} \in \mathscr{E}(\mathbb{P}_i) \tag{33}$$

Furthermore,

$$\sum_{t \in T} contrib^{(t)}(e, \mathbb{R}) \leq \sum_{t \in T} estimate^{(t)}(e, P_i) \leq 2 \sum_{t \in T} contrib^{(t)}(e, \mathbb{R}), \text{ where } \mathbb{R} \in \mathscr{E}(\mathbb{P}_i) \tag{34}$$

By the definition in equations (5) and (10), we have

$$contrib(e, \mathbb{R}) \leq estimate(e, P_i) \leq 2 \cdot contrib(e, \mathbb{R}),\ \forall\ e \in \left(V^{(t)} \times V^{(t)}\right), \text{ where } \mathbb{R} \in \mathscr{E}(\mathbb{P}_i) \tag{35}$$

■

Derivation C.1. *The derivation from Eq. (15) to Eq. (16) in the proof of Lemma 2.*

Proof:

$$\begin{aligned}
&\Phi(i, \mathbb{E}) - \Phi(i+1, \mathbb{E}) \geq \frac{2\Phi(i, \mathbb{E}) - (B_{start} + B_{end})}{2m^*} \\
&\Rightarrow \Phi(i, \mathbb{E}) - \frac{1}{m^*}\Phi(i, \mathbb{E}) \geq \Phi(i+1, \mathbb{E}) - \frac{1}{m^*}\widetilde{B}\ , \text{ where } \widetilde{B} = \frac{B_{start} + B_{end}}{2} \\
&\Rightarrow \left(1 - \frac{1}{m^*}\right)\Phi(i, \mathbb{E}) - \left(1 - \frac{1}{m^*}\right)\widetilde{B} \geq \Phi(i+1, \mathbb{E}) - \frac{1}{m^*}\widetilde{B} - \left(1 - \frac{1}{m^*}\right)\widetilde{B} \\
&\Rightarrow \left(1 - \frac{1}{m^*}\right)\left(\Phi(i, \mathbb{E}) - \widetilde{B}\right) \geq \Phi(i+1, \mathbb{E}) - \widetilde{B}
\end{aligned} \tag{36}$$

■

Derivation C.2. *The derivation for the bounds of λ_0 and λ_1 in the proof of Lemma 2, i.e., Eq. (18) and (19).*

Proof: Eq. (36) shows the progression of the potential function value within successive iterations in GM2 as compared to the optimal solution. Based on Eq. (36), we derive the bound on the number of iterations of Alg. 1 (i.e., the number of edges in E) relative to m^*.

Given $\mathbb{E}$, we take $\left(\Phi(i, \mathbb{E}) - \widetilde{B}\right)$ as a function of i, and it decreases as GM2 adds an edge at each iteration. Initially, $\Phi(0, \mathbb{E}) - \widetilde{B} = \frac{B_{start} - B_{end}}{2} > 0$, and finally, $\Phi(m, \mathbb{E}) - \widetilde{B} = -\frac{B_{start} - B_{end}}{2} < 0$. So at some iteration λ_0, the function turns from positive to negative. We have the values of the function as a sequence with regard to i:

$$\Bigg\langle \underbrace{\left(\Phi(1, \mathbb{E}) - \widetilde{B}\right), \ldots, \left(\Phi(\lambda_0, \mathbb{E}) - \widetilde{B}\right)}_{> 0, \text{ denote the number of such elements by } \lambda_0.}, \underbrace{\left(\Phi(\lambda_0 + 1, \mathbb{E}) - \widetilde{B}\right), \ldots, \left(\Phi(m, \mathbb{E}) - \widetilde{B}\right)}_{\leq 0, \text{ denote the number of such elements by } \lambda_1 = m - \lambda_0.} \Bigg\rangle \tag{37}$$

The difference between successive elements in Eq. (37) is

$$\left(\Phi(i, \mathbb{E}) - \widetilde{B}\right) - \left(\Phi(i+1, \mathbb{E}) - \widetilde{B}\right) = \Phi(i, \mathbb{E}) - \Phi(i+1, \mathbb{E}) = contrib(e_{i+1}, \mathbb{E})\ ,\ 0 \leq i < m. \tag{38}$$

We look at the edge contribution at each iteration. Based on Eq. (4) and (8), every edge contribution on $t \in T$ is a value in $\left\{\frac{1}{2}, \frac{2}{3}, ..., \frac{U-1}{U}\right\}$, where $U = \max\{|V(t)|, t \in T\}$. More specially,

$$contrib^{(t)}(e, \mathbb{E}) \in \left\{\frac{1}{2}, \frac{2}{3}, ..., \frac{U-1}{U}\right\} \text{ , where } U = \max\{|V(t)|, t \in T\} \ . \tag{39}$$

The overall edge contribution $contrib(e, \mathbb{E})$ is the sum of at most $|T|$ values chosen from $\left\{\frac{1}{2}, \frac{2}{3}, ..., \frac{U-1}{U}\right\}$. Thus,

$$contrib(e, \mathbb{E}) \geq \frac{1}{2} \text{ , } \forall e \text{ in } \mathbb{E} \ , \tag{40}$$

and

$$\begin{aligned} \left|\Phi(i, \mathbb{E}) - \widetilde{B}\right| &= \left|B_{start} - \sum_{j=1}^{i} contrib(e_j, \mathbb{E}) - \frac{B_{start} + B_{end}}{2}\right| \\ &= \left|\frac{B_{start} - B_{end}}{2} - \sum_{j=1}^{i} contrib(e_j, \mathbb{E})\right| \\ &= \begin{cases} \text{or} \geq 1/LCM(1, 2, ..., U), \\ \text{either } 0, \end{cases} \quad \forall\, 1 \leq i \leq m \end{aligned} \tag{41}$$

where $LCM(1, 2, ..., U)$ is the Least Common Multiple of $1, 2, ..., U$.

We now return to the sequence in Eq. (37). Let l_0^* be the smallest sequence index in Eq. (37) where the function value is smaller than $1/2$, i.e.,

$$l_0^* = \min\left\{l_0 \mid \left(\Phi(l_0, \mathbb{E}) - \widetilde{B}\right) \leq \left(\Phi(0, \mathbb{E}) - \widetilde{B}\right)\left(1 - \frac{1}{m^*}\right)^{l_0} \leq \frac{1}{2}\right\} \ . \tag{42}$$

By Eq. (40), at the $(l_0^* + 1)$-th iteration,

$$contrib(e_{l_0+1}, \mathbb{E}) = \left(\Phi(l_0^*, \mathbb{E}) - \widetilde{B}\right) - \left(\Phi(l_0^* + 1, \mathbb{E}) - \widetilde{B}\right) \geq 1/2 \ . \tag{43}$$

So we have either $0 < \left(\Phi(l_0^*, \mathbb{E}) - \widetilde{B}\right) \leq 1/2$ and $\left(\Phi(l_0^* + 1, \mathbb{E}) - \widetilde{B}\right) \leq 0$, (44)

or

$$\left(\Phi(l_0^*, \mathbb{E}) - \widetilde{B}\right) \leq 0 \tag{45}$$

In any case of Eq. (44) and (45),

$$\lambda_0 \leq l_0^*, \tag{46}$$

because $\left(\Phi(\lambda_0, \mathbb{E}) - \widetilde{B}\right) > 0$ and $\left(\Phi(\lambda_0 + 1, \mathbb{E}) - \widetilde{B}\right) \leq 0$ (see the sequence in Eq. (37)).

We look back at Eq. (42). According to Eq. (36), $\Phi(l_0, \mathbb{E}) - \widetilde{B} \leq \left(\Phi(0, \mathbb{E}) - \widetilde{B}\right)\left(1 - \frac{1}{m^*}\right)^{l_0}$ always holds, so by definition, l_0^* is the smallest l_0 that satisfies

$$\begin{aligned} &\left(\Phi(0, \mathbb{E}) - \widetilde{B}\right)\left(1 - \frac{1}{m^*}\right)^{l_0} \leq \frac{1}{2} \\ &\Leftrightarrow (B_{start} - B_{end})\left(1 - \frac{1}{m^*}\right)^{l_0} \leq 1 \\ &\Leftrightarrow \ln(B_{start} - B_{end}) + l_0 \cdot \ln\left(1 - \frac{1}{m^*}\right) \leq 0 \\ &\Leftrightarrow l_0 \cdot \ln\left(1 - \frac{1}{m^*}\right) \leq -\ln(B_{start} - B_{end}) \\ &\Leftrightarrow l_0 \geq \frac{\ln(B_{start} - B_{end})}{-\ln\left(1 - \frac{1}{m^*}\right)} \qquad \text{// since } \ln\left(1 - \frac{1}{m^*}\right) < 0 \end{aligned} \tag{47}$$

Further, by the definition of l_0^* in Eq. (42),

$$l_0^* = \left\lceil \frac{\ln(B_{start} - B_{end})}{-\ln\left(1 - \frac{1}{m^*}\right)} \right\rceil \tag{48}$$

By Taylor Expansion,

$$\begin{aligned} & \ln\left(1 - \frac{1}{m^*}\right) = -\frac{1}{m^*} - \frac{1}{2(m^*)^2} - \frac{1}{3(m^*)^3} - \cdots \text{, where } m^* > 0 \\ \Rightarrow \quad & \ln\left(1 - \frac{1}{m^*}\right) < -\frac{1}{m^*} < 0 \\ \Leftrightarrow \quad & -\ln\left(1 - \frac{1}{m^*}\right) > \frac{1}{m^*} > 0 \\ \Leftrightarrow \quad & \frac{1}{-\ln\left(1 - \frac{1}{m^*}\right)} < m^* \end{aligned} \tag{49}$$

Putting Eq. (49) into Eq. (47),

$$l_0^* \leq \lceil m^* \cdot \ln(B_{start} - B_{end}) \rceil \ . \tag{50}$$

Combining, Eq. (46) and (50),

$$\lambda_0 \leq l_0^* \leq \lceil m^* \cdot \ln(B_{start} - B_{end}) \rceil \leq \lceil m^* \cdot \ln B_{start} \rceil = m^* \cdot O(\ln B_{start}) \ . \tag{51}$$

Now we try to derive the bound for λ_1. By Eq. (37), $\Phi(\lambda_0+1, \mathbb{E}) - \widetilde{B} \leq 0$. We just consider $\Phi(\lambda_0+1, \mathbb{E}) - \widetilde{B} < 0$ is the first negative element in the sequence of Eq. (37) – for the case of $\Phi(\lambda_0+1, \mathbb{E}) - \widetilde{B} = 0$, $\Phi(\lambda_0+2, \mathbb{E}) - \widetilde{B} < 0$ would be the first negative element, and we could use the same technique to derive the bound for $(\lambda_1 - 1)$.

Since $\widetilde{B} - \Phi(\lambda_0 + 1, \mathbb{E}) > 0$, λ_1 does not exceed the smallest l_1 that satisfies

$$\left(\widetilde{B} - \Phi(\lambda_0 + 1, \mathbb{E})\right)\left(1 - \frac{1}{m^*}\right)^{l_1} \geq \frac{B_{start} - B_{end}}{2} \ . \tag{52}$$

Similar to the derivation for Eq. (51), we can obtain

$$\lambda_1 \leq \left\lceil m^* \cdot \ln \frac{B_{start} - B_{end}}{2(\widetilde{B} - \Phi(\lambda_0 + 1, \mathbb{E}))} \right\rceil \ . \tag{53}$$

By Eq. (41),

$$\widetilde{B} - \Phi(\lambda_0 + 1, \mathbb{E}) \geq 1/LCM(1, 2, ..., U) \ . \tag{54}$$

Recall that $LCM(1, 2, ..., U)$ is the Least Common Multiple of $1, 2, ..., U$.

The prime number theorem [38] implies that

$$LCM(1, 2, ..., U) = e^{U(1+O(1))} \text{ as } U \to \infty \ . \tag{55}$$

Putting Eq. (54) and (55) into (53), we have,

$$\lambda_1 \leq m^* \cdot O(U + \ln B_{start}) \ . \tag{56}$$

■

APPENDIX D
EVALUATION

As a complement to §VIII, we present additional experiments and analyses in this section.

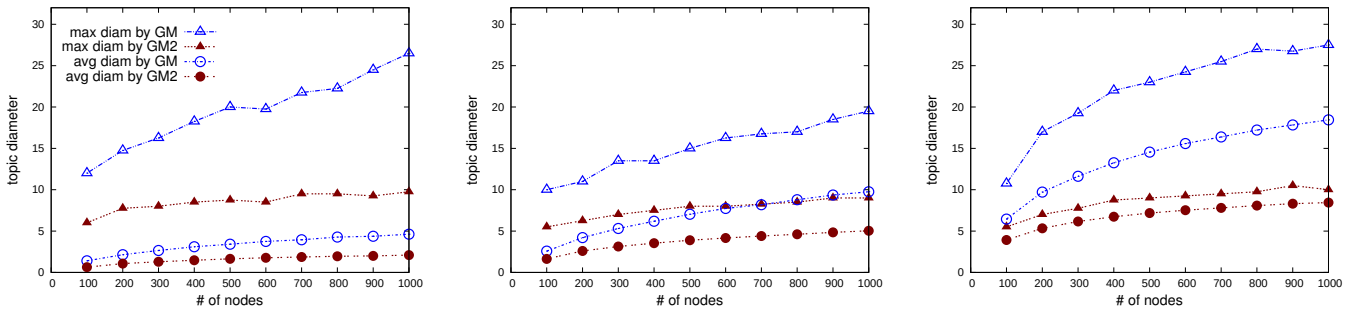
A. The impact of the number of topics

Referring to the HararyPT algorithm in Fig. 4, first, the average node degree of HararyPT increases with the number of topics due to the reduction in the correlation. This reduced correlation has an considerable effect under Unif: To achieve $10\,TCO$, $\overline{d}_{\text{HararyPT}}$ is 54.22 when $|T| = 100$ and 222.84 when $|T| = 1000$ under Unif. As the number of topics increases, the input instances (especially those under Unif) are deviating from our assumption about the high correlation embedded in pub/sub workloads. As a result, HararyPT tends to lose its advantages of aligning the nodes. Second, HararyPT always outputs a $2TCO$ with fewer edges than that produced by CyclePT. Under Expo and Zipf, HararyPT can even achieve $6TCO$ with almost the same number of edges as required by CyclePT.

B. The impact of subscription size

We focus on the HararyPT algorithm in Fig. 5. First, the average node degree increases as the subscription size varies from 10 to 50 under Zipf (or from 10 to 20 under Unif), since each node has more topics to cover. However, when the subscription size exceeds some threshold (e.g., around 50 under Zipf and around 20 under Unif), the average node degrees start to *decrease*. We can explain this phenomenon by the trend that the correlation becomes increasingly dominant as the subscription size increases. Second, HararyPT (for all k values) outperforms CyclePT significantly. For example, when $|T(v)| = 100$, HararyPT can produce $12\,TCO$ with $\overline{d}_{\text{HararyPT}} = 60.20$ on average, across all distributions, whereas CyclePT outputs only $2TCO$ with $\overline{d}_{\text{CyclePT}} = 177.23$.

C. Topic diameters of the overlay



(a) under Expo (b) under Zipf (c) under Unif

Fig. 8: Diameters of GM2 vs. GM wrt. $|V|$

We also compare another important metric in the overlays produced by different algorithms, namely the *topic diameter*. Overlay diameters impact many performance factors for efficient routing in pub/sub, e.g., message latency. Given $2TCO(V, T, Int, E)$, the topic diameter for $t \in T$ is $diam^{(t)} = diam(G^{(t)})$, where $diam(G^{(t)})$ is the maximum shortest distance between any two nodes in $G^{(t)} = \left(V^{(t)}, E^{(t)}\right)$. We denote the maximum and average topic diameter across all topics as $Diam$ and $\overline{diam}$, respectively. Fig. 8 shows that GM2 significantly outperforms GM in terms of both $Diam$ and $\overline{diam}$: $Diam_{\text{GM2}}$ is $0.45 \cdot Diam_{\text{GM}}$, and $\overline{diam}_{\text{GM2}}$ is $0.51 \cdot \overline{diam}_{\text{GM}}$, on average across all three distributions. Besides, the gap grows as the input instances scale up from 100 nodes to 1000: as $|V| = 1000$, $Diam_{\text{GM}} - Diam_{\text{GM2}}$ is 16.75 under Expo, $Diam_{\text{GM}} - Diam_{\text{GM2}}$ is 10.5 under Zipf, and $Diam_{\text{GM}} - Diam_{\text{GM2}}$ is 17.5 under Unif, respectively.