Overlay Design for Topic-based Publish/Subscribe under Node Degree Constraints

Chen Chen
Middleware System Research Group
University of Toronto
chenchen@eecg.toronto.edu

Yoav Tock
IBM Research - Haifa
tock@il.ibm.com

Hans-Arno Jacobsen
Middleware System Research Group
University of Toronto
jacobsen@eecg.toronto.edu

***Abstract*—It is important to build overlays for topic-based publish/subscribe (pub/sub) under resource constraints. In a *topic-connected overlay* (TCO), each topic t induces a connected sub-overlay among all nodes interested in t. Existing work merely consider how to optimize a complete TCO and implicitly commit the unrealistic assumption of unlimited resources.**

In contrast, we make maximum use of restricted node degree budgets to build a *partial* TCO. We formalize the notion of *TCO support* to capture the quality of the pub/sub overlay. Furthermore, we demonstrate that partial TCOs usually exhibit significantly better cost-effectiveness in practice.

We propose two problems of maximizing TCO support in a partial TCO: (1) PTCOA with a bounded *average* node degree and (2) PTCOM under the *maximum* node degree constraint. We design two greedy algorithms, which achieve the constant approximation ratios of $(1 - e^{-1})$ for PTCOA and $(1 - e^{-1/6})$ for PTCOM, respectively.

Empirical evaluation demonstrates the scalability of our algorithms under a variety of pub/sub workloads. Given practical data sets extracted from Facebook and Twitter, our algorithms produce an 80% TCO with fewer than 20% of the node degree budget as a complete TCO. We also show experimentally that it is promising to design decentralized protocols to compute a partial TCO for pub/sub.

***Index Terms*—topic-connected overlay; partial TCO; pub/sub**

I. Introduction

Publish/subscribe (pub/sub) is a popular communication paradigm that provides a loosely coupled form of interaction among many publishing data sources and many subscribing data sinks [15], [20], [21], [31], [34], [36]. This work focuses on *topic-based* pub/sub: publishers associate each publication message with one or more specific topics, and subscribers register their interests in a subset of all topics.

A distributed pub/sub system often organizes nodes (i.e., brokers or servers) as an application-level overlay in a federated or peer-to-peer manner. The overlay infrastructure forms the foundation for distributed pub/sub and directly impacts the system's performance and scalability, e.g., message latency and routing cost. Constructing a high-quality overlay for distributed pub/sub is a key challenge and fundamental problem that has received attention in both industry [15], [31] and academia [12], [13], [14], [27], [32].

Topic-connected overlay (TCO) was introduced in [13] as an abstract model for pub/sub overlays. TCO organizes all nodes interested in each topic t in a directly connected dissemination sub-overlay. First, TCO enables the propagation of publications on each topic t to all subscribers of t without using non-interested nodes as intermediate relays. Publication routing atop TCO saves bandwidth and computational resources otherwise wasted on forwarding and filtering out irrelevant messages. Second, TCO leads to more efficient pub/sub routing, e.g., simpler matching engine implementations and smaller forwarding tables.

Unfortunately, the TCO property is at odds with the requirement of low node degrees in a pub/sub overlay, which may grow proportionally to the number of nodes (or topics) in some cases [14], [32]. While overlay designs for different applications might be principally different, they all strive to control node degrees to be logarithmic or constant, e.g., the *distributed hash table* (DHT) [23], [33], small-world overlays [8], [30], and application-level multicast [25]. First, low-degree overlays reduce active connections and ambient traffic relating to pings, keep-alive messages, and monitoring information [13], [14], [27]. For a typical pub/sub system, each link would also have to accommodate a number of protocols, service components, message queues, etc. Second, in low-degree overlays, the set of nodes that participate in coordination of distributed tasks (e.g., overlay maintenance and load balancing) tends to be smaller. Particularly for pub/sub routing, the node degree directly influences the sizes of the routing tables, the complexity of matching, and the efficiency of message delivery.

To address these issues, [13] and [27] proposed the problems of constructing a TCO with the smallest average and maximum node degrees, proved NP-hardness, and devised greedy algorithms with logarithmic approximation ratios, respectively. Other TCO design problems have also been considered [11], [12], [14], [28], [32]. The basic setting of existing TCO designs is to find a minimum cost overlay that satisfy the TCO requirement. This work studies a more general class of pub/sub overlay design that regards resource constraints. We aim to optimize a *partial* TCO with strict budgets in node degrees, because it is not always feasible or in many cases too costly to build a complete TCO.

First, by targeting the optimization of node degrees or other metrics for a TCO, existing approaches [12], [13], [14], [27], [28], [32] implicitly assume that network resources are infinite and that a TCO is always achievable. The assumption of unlimited node degree budgets does not hold for most large-scale distributed systems. For example, small-world networks, such as DHTs [23], [33] and other structured overlays [8], [30], often demand each node to have only $\Theta(\log N)$ neighbors given the network size N. Viceroy [22] further enforces a constant-degree DHT by emulating traditional butterfly networks. Araneola [25] mathematically resembles k-regular random graphs, where the node degree is either k or $(k+1)$. In BitTorrent [1], each node has to specify a constant limit for upload/download with peers, five by default. While resource constraints are obvious and mandatory in the context of peer-to-peer, mobile or sensor networks, the original motivation for TCO [13], [14] stem from a bounded-degree requirement in enterprise environments, where full-mesh overlays proved unscalable to connect application servers in an IBM production data centre (see also [7]).

Second, even if abundant resources were available, we prefer to be economical in our use of the network, because each connection incurs a non-negligible expense, including maintenance fees, energy costs, traffic charges, etc. We show in §IX that, for many real-world pub/sub applications, well-structured overlays exhibit the *Pareto principle* (a.k.a. the 80-20 *rule*): over 80% of the TCO is supported by fewer than 20% of all edges. Hence, a partial TCO is often more cost-effective than a complete one, and we should focus our optimization efforts on the 20% of vital consequence.

Third, we can compensate for the imperfection of a partial TCO by incorporating effective routing protocols on top of the pub/sub overlay. In [10] , we combine both routing and overlay in one practical pub/sub system and empirically show that a carefully constructed partial TCO can reduce over 30% of the costs in both routing overhead and message latency. Further improvements are possible if we have a thorough understanding about partial TCO design.

This work focuses on centralized algorithm design and analysis with a global knowledge and strives to reveal the fundamental trade-offs in designing partial TCOs under node degree constraints. We are motivated by considerations resulting from the development of a federated pub/sub system within one large-scale data centre. A reference scenario in the context of *Internet of Things* (IoT, see also [3]) would be a data centre hosting a few thousands of MQTT servers [4]. This large cluster of servers collectively constitutes a cloud platform which caters to tens or hundreds of millions of *IoT devices* (e.g. planted sensors). The pub/sub layer on top of the overlay implements the control plane of the system, and therefore the number of topics is proportional to the number of servers, rather than the number of devices. We target at the overlay design within the data centre, namely the topological structure that spans all cluster servers. In such environment, it is usually common to have a centralized entity with global knowledge, such as masters in Google clouds [9], [16], the root node in Hadoop [2], and controllers in *software defined networking* (SDN) [24]. The global knowledge of pub/sub-server subscriptions can easily fit into the main memory of one modern machine. For example, if each of the $10,000$ servers produces 1000 subscriptions on average, then we end up with a manageable amount of $10,000,000$ pub/sub-server subscriptions. A capable machine can easily accommodate this amount in main memory and still have plenty of resources for other jobs, such as executing algorithms, exerting its control over the cluster of pub/sub servers, and so on. Please note that pub/sub-server subscriptions are different from the subscriptions of IoT devices, which are defined by MQTT (or other pub/sub specifications) and handled differently. Each pub/sub server aggregates all of its client subscriptions in a Bloom filter [6] and passes it around [29]. If we allocate 10 bits per element (for a 1% false positive probability), then the entire cluster can represent 10^{10} IoT subscriptions (assuming $100,000,000$ devices $\times$ 100 topics), which consume approximately 12.5GB of memory in the centralized master node. Our master is only responsible for a limited set of centralized operations (e.g., overlay design) and does not reside in the critical data flow paths for pub/sub message dissemination. This lightweight master design greatly simplifies our system development and works well in the targeted application scenarios, where the churn rate is low and the subscriptions are stable.

We bring up these back-of-the-envelope calculations to convince the readers that central control for such a large cluster is no longer a pipe dream. Nevertheless, it is worth mentioning that our proposed algorithms can also serve as comparison baselines and inspire building blocks for other approaches. It is crucial to compare against a centralized algorithm when designing hierarchical organizations, partitioning, or fully distributed architectures. Besides, we often gain principal insights from centralized algorithms, which form bedrocks for more sophisticated solutions. In fact, our empirical evaluation (in §IX) shows that a decentralized protocol based on [14] comes quite close to the performance of our centralized algorithms – a proof that a theoretically sound baseline is a powerful tool and important guideline to distributed algorithm designers.

We summarize our main contributions as follows:

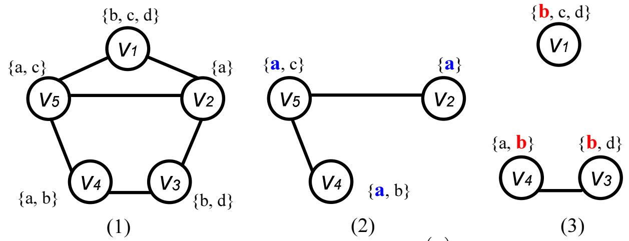
1. We propose the problem of PtcoA that respects the *average* node degree constraint for a partial TCO (§IV). We devise a greedy algorithm GPA, which achieves the approximation ratio of $(1 - e^{-1})$ for PtcoA (§V).

2. We propose the problem of PtcoM that respect the *maximum* node degree constraint for a partial TCO (§VI). We devise a greedy algorithm, GPM, which achieve the approximation ratio of $(1 - e^{-1/6})$ for PtcoM (§VII).

3. §IX thoroughly evaluates our partial TCO design algorithms under a variety of characteristic pub/sub instances of up to $10,000$ nodes and $10,000$ topics. Given data sets extracted from Facebook [36] and Twitter [20], GPA and GPM successfully deliver up to over 80% TCO support with fixed small node degrees, which are lower than 20% of the whole budget in the state-of-the-art complete TCOs.

II. Related Work

A significant body of research has been considering the overlay topology underlying pub/sub systems such that network traffic is minimized [12], [13], [19], [29], [27], [30]. TCO is explicitly enforced in [5], [14], [29], [32] and implicitly



(1) (2) (3)

Figure 1: (1) An overlay G. (2) Subgraph $G^{(a)}$ is completely topic-connected. (3) Subgraph $G^{(b)}$ is partially topic-connected.

manifest in [8], [19], [30], and they all strive to reduce intermediate overlay hops for message delivery.

Many existing approaches build a separate dissemination sub-overlay per topic independently [5], [8], [29], [32], thereby attaining a TCO. Although these construction methods are efficient and easy to implement, the resulting overlays often suffer from prohibitively high node degrees. The theoretical formulation of pub/sub overlay design originated in [13], which proposed the MinAvgTCO problem of constructing a complete TCO with minimum number of edges and designed the GM algorithm with a logarithmic approximation bound. However, GM does not balance degrees across all nodes and often yields the maximum node degree unnecessarily high. To minimize the maximum node degree for a full TCO, [27] explored the MinMaxTCO problem and devised a logarithmic approximation algorithm, MinMaxODA. Following this direction, a number of problems were formulated in constructing complete TCOs while optimizing node degrees and other criteria [12], [14], [27], [28], [32]. Unfortunately, this body of work simply assumes that the network resources are unbounded and ignore practical restrictions in the overlay topologies. To our knowledge, this work is the first attempt to optimize a partial TCO with bounded node degrees.

The core ideas of [12], [13], [27], [28] are rooted in the greedy algorithm for the canonical *minimum set cover* problem. In contrast, we apply *submodular set functions* [18], [26], [37] to the pub/sub overlay construction.

III. Background

A. Topic-connected overlay (TCO) [11], [12], [13], [27]

Let (V, T, I) be an instance: V is the set of nodes, T is the set of topics, and I is the interest function, $I : V \times T \rightarrow \{0, 1\}$. A node v is interested in some topic t iff $I(v, t) = 1$. We also say that node v subscribes to topic t.

Given (V, T, I) and an edge set $E \subseteq K$, the undirected graph $G = (V, E)$ presents a topological structure for the pub/sub overlay, where K is the ground set of all possible edges among V, i.e., $K = V \times V$. The sub-overlay *induced* by topic $t \in T$ is a subgraph $G^{(t)} = (V^{(t)}, E^{(t)})$ such that $V^{(t)} = \{v \in V | I(v, t) = 1\}$ and $E^{(t)} = \{(u, v) \in E | u \in V^{(t)} \wedge v \in V^{(t)}\}$. A *topic-connected component* (TCC) on topic $t \in T$ is a maximal connected subgraph in $G^{(t)}$. E forms a *topic-connected overlay* (TCO) if each topic $t \in T$ induces at most one TCC in $G^{(t)}$ (see Fig. 1).

Aiming to achieve TCO while optimizing node degrees has resulted in the formulation of various problems, such as MinAvgTCO for the average degree [13] and MinMaxTCO for the maximum degree [27].

Problem 1: MinAvgTCO(V, T, I): given (V, T, I), find $E \subseteq K$ that forms a TCO with the smallest *average* degree, i.e., the total number of edges is minimum.

$$\min_{E \subseteq K} \{|E| : E \text{ forms a } TCO \text{ for } (V, T, I)\} \tag{1}$$

Problem 2: MinMaxTCO(V, T, I): given (V, T, I), find $E \subseteq K$ that forms a TCO with the smallest *maximum* degree.

$$\min_{E \subseteq K} \{\Delta(V, E) : E \text{ forms a } TCO \text{ for } (V, T, I)\} \tag{2}$$

where $\Delta(V, E)$ is the maximum node degree of $G = (V, E)$.

Both Problem 1 and 2 are NP-complete. GM [13] and MinMaxODA [27] achieve logarithmic approximation ratios for MinAvgTCO and MinMaxTCO, respectively.

B. Submodular set function [18], [26], [37]

Let a ground set K be finite and nonempty. We consider an integral valued nonnegative function

$$\mu : 2^K \rightarrow \mathbb{N}^0 = \{0, 1, 2, \ldots\}$$

Definition 1: A set function $\mu : 2^K \rightarrow \mathbb{N}^0$ is
(a) *nondecreasing* if $\forall E \subseteq F \subseteq K$,

$$\mu(E) \leq \mu(F) \tag{3}$$

(b) *submodular* if $\forall E \subseteq F \subseteq K, \forall e \in K$,

$$\mu(E + e) - \mu(E) \geq \mu(F + e) - \mu(F) \tag{4}$$

We use e and $\{e\}$ interchangeably if there is no ambiguity.

Given a submodular function $\mu : 2^K \rightarrow \mathbb{N}^0$, we define $\mu'(S|E)$ as the *discrete derivative* of μ for S at E:

$$\mu'(S|E) = \mu(E + S) - \mu(E), \forall S \subseteq \overline{E}, \forall E \subseteq K \tag{5}$$

where $\overline{E}$ is the *complement* of E (w.r.t. the ground set K), i.e., $\overline{E} = K \backslash E$. We can rewrite Eq. (4) as

$$\mu'(e|E) \geq \mu'(e|F), \forall E \subseteq F \tag{6}$$

Submodular set functions naturally have the property of *diminishing returns*, which makes them suitable for approximation algorithm design, as this work illustrates.

IV. PtcoA - Maximizing Partial TCO under Average Node Degree Constraint

Our first objective is to optimally build a partial TCO with a bounded average node degree. Before formally presenting the problem statement, we introduce several key definitions.

Given (V, T, I) and $E \subseteq K$, $TCC(E)$ stands for the total number of TCCs in $G = (V, E)$ over all topics T,

$$TCC(E) = \sum_{t \in T} \left(\#\text{TCCs in } G^{(t)} = (V^{(t)}, E^{(t)}) \right) \tag{7}$$

By definition,

$$TCC(\emptyset) = \sum_{t \in T} \left| V^{(t)} \right| \tag{8}$$

$$TCC(K) = \left| \{ t \in T : V^{(t)} \neq \emptyset \} \right| \tag{9}$$

$$TCC(E) = TCC(K) \text{ iff } E \subseteq K \text{ forms a TCO} \tag{10}$$

We use $TCC(E)$ to measure the progress towards TCO. Suppose E is initially empty and grows by adding edges one by one, then $TCC(E)$ starts from $TCC(\emptyset)$ and decreases with every edge addition down to an absolute limit, $TCC(K)$.

We define the energy function $\mu : 2^K \to \mathbb{N}^0$:

$$\mu(E) = TCC(\emptyset) - TCC(E), \forall E \subseteq K \tag{11}$$

Function $\mu(E)$ represents the *contribution* of edge set $E \subseteq K$ (towards TCO) with respect to the empty set $\emptyset$, i.e., the amount of TCCs that E reduces from $\emptyset$. Function $\mu(E)$ linearly depends on $TCC(E)$, since $TCC(\emptyset)$ is a constant for a given instance. As a result, $\mu(E)$ is equivalent to $TCC(E)$ as a progress measure towards TCO: as the edge set E grows from empty, $\mu(E)$ starts from $\mu(\emptyset) = 0$ and increases with every edge addition up to $\mu(K) = TCC(\emptyset) - TCC(K)$, the amount of TCCs that E needs to reduce to attain a TCO.

According to Eq. (10) and (11), an edge set $E \subseteq K$ forms a (complete) TCO for (V, T, I) iff $\mu(E) = \mu(K)$. We say that E forms a *partial* TCO for (V, T, I) iff $\mu(E) \leq \mu(K)$.

With Eq. (11) and Eq. (5), we have

$$\begin{aligned} &\mu'(e|E) = \mu(E+e) - \mu(E) \\ &= TCC(E) - TCC(E+e), \forall e \in \overline{E}, \forall E \subseteq K \end{aligned} \tag{12}$$

This discrete derivative $\mu'(e|E)$ defines the *contribution* (towards TCO) of edge e with respect to the current edge set E, which is the number of TCCs that would be reduced by adding e upon E.

Further, the accumulated contribution of an edge set $S \subseteq \overline{E}$ with regard to the current edge set $E \subseteq K$ is

$$\begin{aligned} &\mu'(S|E) = \mu(E+S) - \mu(E) \\ &= TCC(E) - TCC(S+E), \forall S \subseteq \overline{E}, \forall E \subseteq K \end{aligned} \tag{13}$$

We formalize the problem of optimizing a partially constructed TCO with a global budget in the cardinality of the edge set, namely Maximizing Partial TCO under the Average Node Degree Constraint, PTCOA for short.

Problem 3: PTCOA(V, T, I, m): given (V, T, I) and an integer m, find $E \subseteq K$ that maximizes $\mu(E)$ where $|E| \leq m$, i.e., the *average* degree of $G = (V, E)$ never exceeds $\frac{2m}{|V|}$.

$$\max_{E \subseteq K} \{\mu(E) : |E| \leq m\} \tag{14}$$

Problem 3 of PTCOA genetically relates to Problem 1 of MINAVGTCO [13]. PTCOA is in the form of *maximum packing* which chooses a subset of items maximizing the total benefits subject to capacity constraints; meanwhile, MINAVGTCO falls into the class of *minimum covering*, that is to find a set of items with minimum cost to accomplish a target.

V. GPA: GREEDY APPROXIMATION FOR PTCOA

Alg. 1 specifies GPA, greedy algorithm for PTCOA. Alg. 1 starts from an empty edge set $E = \emptyset$ and greedily adds to E the edge with the highest contribution to the TCO construction at each iteration (Line 4), until reaching the cardinality constraint or attaining an integral TCO (Line 2).

We can regard GPA as an extension of GM [13]: GPA behaves exactly as GM until running out of the total number of edges allowed. Given a sufficient budget to build a complete TCO, GPA is equivalent to GM. Although GM only attains a logarithmic approximation ratio of $\mathcal{O}(\log |T|)$ for MINAVGTCO [13], GPA guarantees a constant approximation factor of $(1 - e^{-1})$ for PTCOA.

Lemma 1: The approximation ratio of Alg. 1 is at least $(1 - e^{-1}) \approx 0.632$.

To prove Lemma 1, we dive deep into the PTCOA problem. First, we cast PTCOA into the form of *Maximizing Submodular Set Functions under Cardinality Constraints* [26]. Then, we rely on an existing theorem for the analysis of GPA.

By definition of the set function μ in Eq. (11), $\forall E \subseteq K$

$$\mu(E) = \mu(K) \Leftrightarrow E \text{ forms a TCO for } (V, T, I) \tag{15}$$

According to Eq. (3) and (4), function μ is

(a) *nondecreasing*, because

$$\mu(E) \leq \mu(F), \forall E \subseteq F \subseteq K \tag{16}$$

(b) *submodular*. If $E \subseteq F \subseteq K$, then

$$\mu(E+e) - \mu(E) \geq \mu(F+e) - \mu(F), \forall e \in K\ , \tag{17}$$

because E may induce additional TCCs for e as compared to its superset F. In other words, adding edges from $(F - E)$ can only reduce the contribution of e towards TCO (see Fig. 2).

With Eq. (15), (16), and (17), we can formulate Problem 3 as an instance of the *Maximizing Submodular Set Functions under Cardinality Constraints* (MSSCC) problem:

Problem 4: MSSCC(K, m, μ): Given a finite set K, a positive constant m, and an integral-valued nondecreasing submodular set function $\mu : 2^K \to \mathbb{N}$, find a subset $E \subseteq K$ that maximize $\mu(E)$ while $|E| \leq m$, i.e.,

$$\max_{E \subseteq K} \{\mu(E) : |E| \leq m\} \tag{18}$$

Further, Alg. 1 for PTCOA follows the same greedy approach as Alg. 2 for MSSCC.

Theorem 1 provides an approximation guarantee of Alg. 2 for MSSCC, which has guided us to derive Lemma 1.

Theorem 1: [26] Given an instance of MSSCC, Alg. 2 always produces an approximation solution whose value is at least $(1 - \frac{1}{e})$ times the optimum.

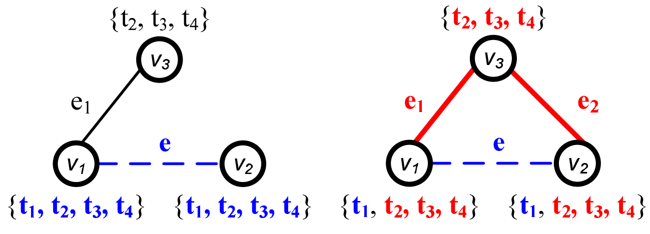
Proof of Lemma 1: According to Theorem 1, the approximation ratio of Alg. 1 is at least $(1 - e^{-1})$. ■

This constant approximation ratio of Alg. 1 in Lemma 1 is tight for PTCOA, which we put in Lemma 2.

Lemma 2: Problem 3 of PTCOA is NP-complete and cannot be approximated in polynomial time within a ratio of $(1 - e^{-1} + \epsilon)$ for any $\epsilon > 0$, unless P = NP.

Alg. 1 Greedy algorithm for PTCOA

```
GPA(V, T, I)
Input: $(V, T, I, m)$
Output: $E$
$E \leftarrow \emptyset$
while $|E| \leq m$ and $E$ does not form TCO do
    $e \leftarrow \arg\max_{e \in \overline{E}} \mu'(e|E)$
    $E \leftarrow E + e$
return $E$
```



(a) $E = \{e_1\}$, $\mu'(e|E) = 4$ (b) $F = \{e_1, e_2\}$, $\mu'(e|F) = 1$

Figure 2: Function μ is submodular: if $E \subseteq F$, then $\mu'(e|E) \geq \mu'(e|F)$, i.e., the contribution of e decreases, as the current edge set progressively extends from E to F.

Lemma 2 follows directly from Theorem 2.

Theorem 2: [18] Problem 4 of MSSCC is NP-complete and cannot be approximated in polynomial time within a ratio of $(1 - e^{-1} + \epsilon)$ for any $\epsilon > 0$, unless P = NP.

Lemma 1 and 2 jointly show that no polynomial-time algorithm can outperform Alg. 1. Moreover, Alg. 1 is efficient.

Lemma 3: The running time of Alg. 1 is $\mathcal{O}(|V|^2|T|)$.

Proof: The time complexity of GPA is bounded by that of GM, which is $\mathcal{O}(|V|^2|T|)$ [13]. ■

VI. PTCOM - Maximizing Partial TCO under Maximum node degree constraint

While tackling PTCOA is the first step towards building pub/sub overlays under node degree constraints, GPA inherits the drawbacks from GM and may produce a partial TCO with unbalanced node degrees (see [27], [28] and empirical evaluation in §IX). We therefore formalize the second problem of approaching a TCO with a uniform degree budget at each node locally, namely Maximizing Partial TCO with the Maximum Node Degree Constraint, PTCOM for short.

Problem 5: PTCOM(V, T, I, d): given (V, T, I) and an integer d, find $E \subseteq K$ that maximize $\mu(E)$ while the maximum node degree of graph $G = (V, E)$, denoted by $\Delta(V, E)$, is no more than d, i.e., $\Delta(V, E) \leq d$.

$$\max_{E \subseteq K} \{\mu(E) : \Delta(V, E) \leq d\} \tag{19}$$

Theorem 3: PTCOM is NP-complete.

Proof: This result follows directly from two facts: (1) The decision versions of Problem 2 and 5 are equivalent; and (2) Problem 2 of MINMAXTCO is NP-complete. ■

Just like PTCOA and MINAVGTCO, Problem 5 of PTCOM and Problem 2 of MINMAXTCO constitute a pair of *maximum packing* and *minimum covering* problems.

Alg. 2 [26] Greedy algorithm for MSSCC

```
Input: $(K, m, \mu)$
Output: $E$
$E \leftarrow \emptyset$
while $|E| \leq m$ and $\mu(E) \neq \mu(K)$ do
    $e \leftarrow \arg\max_{e \in \overline{E}} (\mu(E + e) - \mu(E))$
    $E \leftarrow E + e$
return $E$
```

VII. GPM: Greedy approximation for PTCOM

Alg. 3 specifies GPM, greedy algorithm for PTCOM. GPM operates in the same greedy framework as GPA: Alg. 3 initializes $E = \emptyset$, iteratively adds to E edge by edge, and terminates by either reaching the limit of the node degree constraint in Lines 5-6 or attaining a complete TCO in Line 2. At each iteration, GPM adopts the same edge selection rule as MinMaxODA [27]: Lines 3-4 select the edge with the highest contribution from the candidate edge set C, i.e., all potential edges that would minimally increase the maximum node degree of the current overlay.

MinMaxODA is a special case of GPM in which a complete TCO is returned. As opposed to MinMaxODA's logarithmic approximation ratio $\mathcal{O}(\log(|V||T|))$ for MINMAXTCO [27], the approximation ratio of GPM for PTCOM is bounded by a constant, $(1 - e^{-\frac{1}{6}})$.

Lemma 4: The approximation ratio of Alg. 3 is at least $(1 - e^{-\frac{1}{6}}) \approx 0.154$.

At a high level, the proof of Lemma 4 combines both the analysis for MinMaxODA [27] and the techniques for maximizing submodular set functions under constraints [26].

Let us take a closer look at the execution of GPM for a given PTCOM instance (V, T, I, d). Denote by E the output edge set of GPM and by E^* an optimal solution.

As Alg. 3 presents, GPM progresses phase by phase, and each phase adds a *matching*, i.e., a set of edges without common nodes. Initially in Phase 0, $E = \emptyset$, $\Delta(V, E) = 0$, and all nodes have an equal degree zero. In Phase 1, GPM adds an edge set M_1 that increases the degree of every node to 1; M_1 constitutes a *maximal matching*, and the next edge that GPM adds will inevitably upgrade the maximum node degree by 1, i.e., $\Delta(V, M_1 + e) - \Delta(V, M_1) = 1, \forall e \in \overline{M_1}$, and this marks the beginning of Phase 2. GPM repeats the same procedure in Phase $1, 2, \ldots,$ until reaching the local budget at each node (at the end of Phase d) or returning a whole TCO.

Denote by M_i the i-th matching that Alg. 3 adds in Phase i, $0 \leq i \leq d$, where $M_0 = \emptyset$. Let $E_i = \bigcup_{j=0}^{i} M_j, 0 \leq i \leq d$, and by definition, Alg. 3's output edge set $E = E_d$. Then we have the following lemma from [27], which guarantees that each phase of GPM produces a high-quality matching.

Lemma 5: [27] The contribution of M_{i+1} with regard to E_i is at least $1/3$ of that for any matching, $i \geq 0$:

$$\mu'(M_{i+1}|E_i) \geq \frac{1}{3}\mu'(M|E_i),\ M \text{ is a matching, } i \geq 0 \tag{20}$$

Alg. 3 Greedy algorithm for PTCOM

```
GPM$(V, T, I)$
Input: $(V, T, I, d)$
Output: $E$
1: $E \leftarrow \emptyset$
2: while $E$ does not form TCO for $(V, T, I)$ do
3:     $C \leftarrow \arg\min_{e \in \overline{E}} \Delta(V, E+e)$ // Candidate edge set
4:     $e \leftarrow \arg\max_{e \in C} \mu'(e|E)$
5:     if $\Delta(V, E+e) > d$ then
6:         break from while-loop Lines 2-8
7:     else
8:         $E \leftarrow E + e$
9: return $E$
```

Lemma 5 has taken us through an attempt of comparing E with the optimal edge set E^* phase by phase. To pave this path, we need to split E^* into a number of matchings, which directs us to a well-known result from graph theory.

Lemma 6: [17], [27] Given a graph $G = (V, F)$ with the maximum node degree $\Delta(V, F)$, we can always divide the edge set F into $(\Delta(V, F) + 1)$ matchings.

Please find proofs of Lemma 5 and 6 in [27], which we also rewrite in Appendix. Now let us review Lemma 4 and complete the proof for the approximation ratio of GPM.

Proof of Lemma 4: Based on Lemma 6, we can partition E^* into $(d+1)$ disjoint matchings, $M_j^*, 1 \leq j \leq (d+1)$. Without loss of generality, we define $M_0^* = \emptyset$ and $E_i^* = \bigcup_{j=0}^{i} M_j^*, 0 \leq i \leq (d+1)$, then $E_0^* = \emptyset$ and $E_{d+1}^* = E^*$.

We have the following sequence of inequalities for all $i \geq 0$,

$$\mu(E^*) \leq \mu(E^* + E_i) \tag{21}$$

$$= \mu(E_i) + \sum_{j=1}^{d+1} \mu'(M_j^* | E_i \cup E_{j-1}^*) \tag{22}$$

$$\leq \mu(E_i) + \sum_{j=1}^{d+1} \mu'(M_j^* | E_i) \tag{23}$$

$$\leq \mu(E_i) + \sum_{j=1}^{d+1} 3\mu'(M_{i+1} | E_i) \tag{24}$$

$$= \mu(E_i) + 3(d+1)\left(\mu(E_{i+1}) - \mu(E_i)\right) \tag{25}$$

Eq. (21) comes from monotonicity of μ. Eq. (22) is a telescoping sum – recall discrete derivatives of submodular functions in Eq. (5) and (12). Eq. (23) follows from the submodular property of μ. Eq. (24) holds because of Lemma 5. Hence,

$$\mu(E^*) - \mu(E_i) \leq 3(d+1) \cdot \left(\mu(E_{i+1}) - \mu(E_i)\right) \tag{26}$$

Let $h_i = \mu(E^*) - \mu(E_i)$, and then

$$h_i \leq 3(d+1)(h_i - h_{i+1}) \tag{27}$$

$$\Rightarrow h_{i+1} \leq \left(1 - \frac{1}{3(d+1)}\right) h_i \tag{28}$$

$$\Rightarrow h_i \leq \left(1 - \frac{1}{3(d+1)}\right)^i h_0 \leq e^{-\frac{i}{3(d+1)}} \cdot h_0 \tag{29}$$

$$\Rightarrow \mu(E^*) - \mu(E_i) \leq e^{-\frac{i}{3(d+1)}} \cdot \left(\mu(E^*) - \mu(\emptyset)\right) \tag{30}$$

$$\Rightarrow \mu(E_i) \geq \left(1 - e^{-\frac{i}{3(d+1)}}\right) \cdot \mu(E^*) \tag{31}$$

The second "$\leq$" of Eq. (29) comes from the inequality

$$1 - x \leq e^{-x}, \forall x \in \mathbb{R} \tag{32}$$

In particular, for $i = d$ in Eq. (31),

$$\mu(E) = \mu(E_d) \geq \left(1 - e^{-\frac{d}{3(d+1)}}\right) \cdot \mu(E^*) \tag{33}$$

$$\geq \left(1 - e^{-\frac{1}{6}}\right) \cdot \mu(E^*) \tag{34}$$

Eq. (34) holds because $\frac{d}{d+1} \geq \frac{1}{2}, \forall d \in \mathbb{N} = \{1, 2, \ldots\}$. ■

Lemma 7: The running time of Alg. 3 is $\mathcal{O}(|V|^2|T|)$.

Proof: The time complexity of GPM is bounded by that of MinMaxODA [27], which is $\mathcal{O}(|V|^2|T|)$ [12]. ■

VIII. PARTIAL VERSUS COMPLETE

Our proposed algorithms achieve constant approximation ratios for partial TCO problems of PTCOA and PTCOM, while existing greedy algorithms for complete TCO problems attain logarithmic approximation ratios. This gap stems from different problem definitions between partial and complete TCOs rather than superiority of algorithmic strategies. Partial TCO problems optimize the number of TCCs, and complete TCO problems target at node degrees. Since previous greedy algorithms are special cases of our proposed algorithms, the worst-case performances of GPA and GPM are no better than those of the GM and MinMaxODA.

IX. EVALUATION

A. Experiment setup

We simulate GPA, GPM, and other algorithms in Java. First, we choose GM [13] and MinMaxODA [27] for comparison, which produce the TCOs with the lowest *average* and *maximum* node degrees among all known polynomial algorithms, respectively. Second, we place SpiderCast [14] in the chart, because it is highly efficient in constructing TCOs in a decentralized peer-to-peer manner and has been adopted in practice [35]. The original SpiderCast does not restrict node degrees, so we develop a refined version, SpiderCastM, which extends the protocol by imposing the *maximum* node degree constraint. More specifically, each SpiderCastM node has access to the global knowledge of all subscriptions and independently runs the neighbor selection heuristics until the node degree reaches the given maximum. We do not revise SpiderCast with regard to a constant *average* node degree, because it requires cooperations among peers and violates the decentralized spirit of SpiderCast. Third, we implement two random heuristics, RandomA and RandomM, which construct partial TCOs arbitrarily while respecting the *average* and *maximum* node degree constraints, respectively.

We mainly look at the value of $\mu(E)$, which is the optimization objective of Problem 3 and 5. In particular, we normalize $\mu(E)$ and define the *TCO support ratio* for an edge set E:

$$TcoSuppR(E) = \frac{\mu(E)}{\mu(K)} = \frac{\mu(\emptyset) - \mu(E)}{\mu(\emptyset) - \mu(K)} \qquad (35)$$

$TcoSuppR$ serves as an indicator of the overlay quality: the higher $TcoSuppR(E)$ is, the closer E approximates a TCO. Particularly, (1) $TcoSuppR(E) \in [0,1]$, $TcoSuppR(\emptyset) = 0$, $TcoSuppR(K) = 1$, and $TcoSuppR(E) = 1$ iff E forms a TCO; and (2) $TcoSuppR(E)$ is non-decreasing as E expands, i.e., $TcoSuppR(E) \leq TcoSuppR(E{+}e), \forall e \in \overline{E}$. We say that E forms an 80% TCO, if $TcoSuppR(E) = 0.8$.

We denote by E_{Alg} the overlay edge set that Alg produces, where Alg can be any algorithm. When there is no ambiguity, we often simplify $TcoSuppR(E_{\mathsf{Alg}})$ by $TcoSuppR_{\mathsf{Alg}}$.

B. Experiment workloads

We synthetically generate pub/sub workloads with three types of topic popularities: exponential, Zipfian, and uniform. We also extract data from real-world social networks, namely Facebook and Twitter.

(1) *Synthetic workloads*: Our inputs have the following ranges: $|V| \in [1\,000, 10\,000]$ and $|T| \in [1000, 10\,000]$. We denote by $T(v)$ the topic set to which node v subscribes and by $|T(v)|$ the *subscription size* of node v, i.e., $|T(v)| = |\{t \in T | I(v,t) = 1\}|$. In most cases, we set $|T(v)|_{\min} = 20$ and $|T(v)|_{\max} = 600$, where $|T(v)|_{\min}$ and $|T(v)|_{\max}$ denote the minimum and maximum subscription size over all nodes. We associate each topic $t \in T$ with a probability $q(t)$: each node subscribes to t with the probability $q(t)$. We draw $q(t)$ from either an exponential, a Zipfian (with $\alpha = 2.0$), or a uniform distribution, which we call Expo, Zipf, or Unif, respectively. These distributions are representative of actual workloads in today's industrial pub/sub systems [14]. Stock market monitoring engines use Expo for the study of stock popularity in the New York Stock Exchange [34]. Zipf faithfully describes the feed popularity distribution in RSS feeds [21].

(2) *Facebook dataset*: We use a Facebook dataset [36] with over 3 million distinct user profiles and 28.3 million social relations. In Facebook, when a user, say Alice, performs an activity (e.g., updating her status, sharing photos, or commenting on a post), all Alice's friends receive a notification. Therefore, we model each user as a topic, and all her friends as the respective subscribers. Likewise, the friend set of Alice forms her subscription set. Facebook relations are bidirectional: friends in Alice's social graph subscribe to notifications about Alice and vice versa.

(3) *Twitter dataset*: We also take a public Twitter dataset [20], containing 41.7 million distinct user profiles and 1.47 billion social followee/follower relations. Similarly to Facebook, we model users as topics and subscribers. However, relations in Twitter are unidirectional, i.e., Alice following Bob does not imply that Bob follows back.

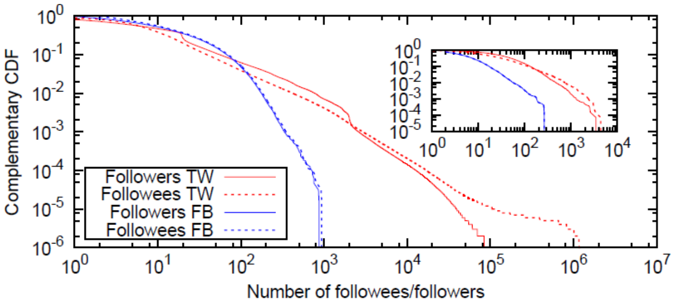


Figure 3: Complementary cumulative distribution function (CCDF) of followee/follower counts. Outer plot: Facebook (3M users) and Twitter (41.7M users). Inner plot: FB 10K and TW 10K.

We extract the workloads from the original Facebook and Twitter social graphs with a sampling methodology inspired from [30], [32]. We start with a few users as seeds and traverse the social graph via breadth first search until reaching the targeted number of nodes, and our sample includes all edges among the visited nodes. The sizes of our samples are 1K and 10K, i.e., $|V| \approx$ 1K, $|T| \approx$ 1K or $|V| \approx$ 10K, $|T| \approx$ 10K. We denote our sample instances by FB 1K, FB 10K, TW 1K, and TW 10K, respectively. Fig. 3 illustrates that our extracted samples retain the properties for the original data sets.

C. Partial TCOs for online social networks

We evaluate $TcoSuppR$ for various partial TCO design algorithms for real-world online social networks.

To guarantee fairness, we ensure that all partial TCOs output by GPA, GPM, SpiderCastM, RandomA, and RandomM have roughly the same number of edges. We first run MinMaxODA [27] and obtain a complete TCO; then we take $E_{\mathsf{MinMaxODA}}$ and a *budget ratio* $\beta \in (0,1]$ to set the average and maximum node degree constraints:

$$m = \beta \cdot |E_{\mathsf{MinMaxODA}}| \qquad d = \beta \cdot \Delta(V, E_{\mathsf{MinMaxODA}}) \qquad (36)$$

where m and d are defined in Problem 3 and 5 and fed as input parameters for different algorithms.

First, Fig. 4 and 5 show a general and consistent trend of GPA $>$ GPM $>$ SpiderCastM $>$ RandomA, RandomM through all cases. GPA, GPM, and SpiderCastM outperform the naive RandomA and RandomM by wide margins, and these gaps become even more remarkable when the instances scale up. For example, $(TcoSuppR_{\mathsf{GPM}} - TcoSuppR_{\mathsf{RandomM}})$ with $\beta = 0.2$ enlarges from 58.5% to 91.4% when the Facebook instances grow from 1K and 10K. This demonstrates the significance of partial TCO design for pub/sub systems.

Second, GPA, GPM and SpiderCast empirically present the *Pareto principle* for the edge contributions towards TCO under both Facebook and Twitter: over 80% of the TCO is attributable to less than 20% of all edges in a complete TCO. This phenomena is more profound for larger instances; as shown in Table I, $TcoSuppR_{\mathsf{GPM}}$ improves from 95.1% to 97.4% as our Facebook samples grow from 1K to 10K.

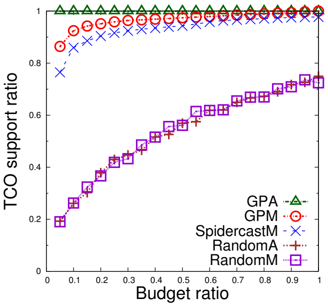


Figure 4: FB 1K

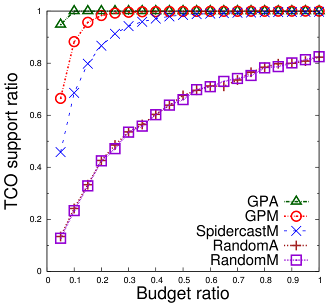


Figure 5: TW 1K

The root cause for this 80-20 rule rendered in the output overlays lies in the fact that these social network workloads are characterized by a power law distribution (a.k.a. a Pareto distribution): most subscriptions come from a small set of popular topics, and a bulk of topics attract only a few subscribers and are not well correlated. Since the popular topic set is of small cardinality and vital significance, a small number of well selected edges can support a highly connected partial TCO; meanwhile, the "heavy tail" of unpopular topics costs a massive number of additional edges to fully achieve a TCO. Under FB 1K, MinMaxODA attains a complete TCO with the maximum node degree 231, while GPM achieves a 94.3% TCO with a budget ratio $\beta = 15\%$, i.e., $d = 231 \cdot 15\% = 34$. These results provide a strategic insight for pub/sub overlay design: knowing that roughly 80% of the edges in a complete TCO are of minor contributions, it is more cost-effective to focus our efforts on optimizing the 20% that is critical.

Third, GPA yields a slightly higher $TcoSuppR$ than GPM, because the average degree constraint admits more flexibility than the maximum degree constraint. However, this marginal improvement comes at a price: E_{GPA} is unevenly distributed across all nodes and effectively creates a few hotspot nodes whose degrees are often unacceptably high; moreover, this inborn weakness of GPA is amplified by the Pareto principle observed in the workloads. For instance, at $\beta = 0.2$, the average and maximum node degree of GPA is 2.74 and 373. Note that GPA may hit a complete TCO even with a low β. This is not surprising, because the *actual* budget ratio of GPA with regard to GM is higher than β, i.e., $\frac{|E_{\mathsf{MinMaxODA}}|}{|E_{\mathsf{GM}}|} \cdot \beta > \beta$, given $E_{\mathsf{MinMaxODA}}$ generally contains more edges than E_{GM}.

Fourth, as a simple peer-to-peer protocol, SpiderCastM yields high TCO support ratios that are close to those of GPA and GPM: $(TcoSuppR_{\mathsf{GPM}} - TcoSuppR_{\mathsf{SpiderCastM}})$ is 3.4% under FB 1K and 4.8% under TW 1K, on average respectively. The impressive performance of SpiderCastM strongly encourages the design of more advanced decentralized protocols for partial TCOs, which may take a local knowledge or promote coordination. This sits on top of our future work.

The difference between RandomA and RandomM is negligible, therefore we only plot and report RandomM in the rest of our evaluation.

Table I: TCO support ratio with $\beta = 0.2$ under FB and TW

$TcoSuppR$	FB1K	FB10K	TW1K	TW10K
GPA	1	1	1	1
GPM	0.951	0.974	0.983	0.998
SpiderCastM	0.904	0.915	0.867	0.986
RandomM	0.367	0.06	0.425	0.669

D. Impact of budget ratio β

We study the impact of budget ratio β on the output partial TCOs of different overlay design algorithms. We generate instances under three pub/sub topic popularities with $|V| = 1000$, $|T| = 1000$, $|T(v)|_{\min} = 20$, and $|T(v)|_{\max} = 600$.

First, Fig. 9 presents the same performance rank as Fig. 4 and 5: GPA $>$ GPM $>$ SpiderCastM $>$ RandomM. Second, GPA, GPM, and SpiderCastM still exhibit the Pareto principle in many cases, which is more evident under more skewed workloads, such as Expo and Zipf. Even under Unif, GPA, GPM, and SpiderCastM manage to fulfill most of the TCO with only a small amount of edges and hence yield much better TCO support than RandomM. This further proves the advantages of greedy strategies over naive randomness in the construction of pub/sub partial TCOs.

E. Impact of $|V|$

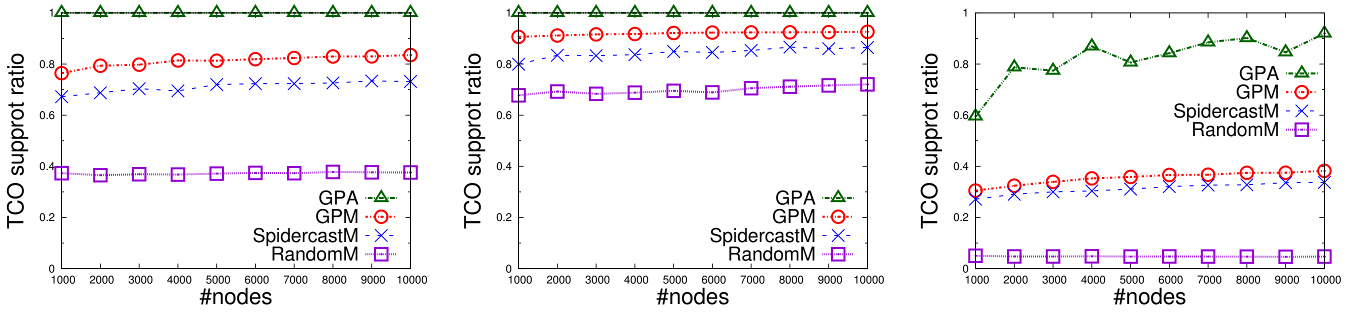
Fig. 6 depicts the comparison among GPA, GPM, SpiderCastM and RandomM as $|V|$ increases from 1000 to $10,000$, where we fix $|T| = 1000$, $|T(v)|_{\min} = 20$, and $|T(v)|_{\max} = 600$. Besides, we set the maximum node degree budget to be $d = 4$, which is around 20% of the maximum node degree in the TCO produced by MinMaxODA; accordingly, the global budget in the total number of edges is $m = \frac{d \times |V|}{2} = 2|V|$.

First, GPA, GPM, and SpiderCastM show steady performance acceleration over RandomM in TCO support. For example, $(TcoSuppR_{\mathsf{GPM}} - TcoSuppR_{\mathsf{RandomM}})$ is on average 43.9%, 22.7%, and 30.1% under Expo, Zipf, and Unif, respectively. This benefit becomes increasingly important when we raise the number of nodes: $TcoSuppR$ for all GPA, GPM, and SpiderCastM increases at a noticeable rate, while $TcoSuppR_{\mathsf{RandomM}}$ remains constantly low with little improvement. This is expected, because (1) all GPA, GPM, and SpiderCastM strive to exploit the subscription correlation for constructing partial TCOs; and (2) larger node sets lead to more candidate edges and thus higher opportunities for these algorithms to find better edges with more contributions.

Second, the TCO support ratios drop as the pub/sub workloads are losing the innate skewness from Expo and Zipf to Unif. At $|V| = 10,000$, $TcoSuppR_{\mathsf{GPM}}$ is 83.5%, 92%, 38.3% under Expo, Zipf, and Unif, respectively. Besides, the Pareto principle holds for GPA and GPM under Expo and Zipf and get lost under Unif. We can explain this by the characteristics in the input instances: the Pareto principle is embedded in Expo and Zipf, but not in Unif.

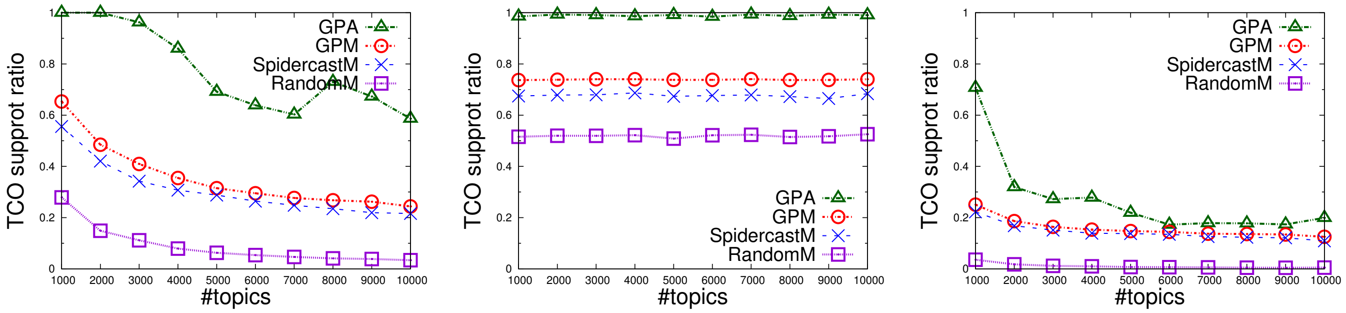
F. Impact of $|T|$

Fig. 7 shows how GPA, GPM, SpiderCastM, and RandomM behave when the topic set varies in size. We increase $|T|$ from 1000 and $10,000$ and fix $|V| = 2000$,



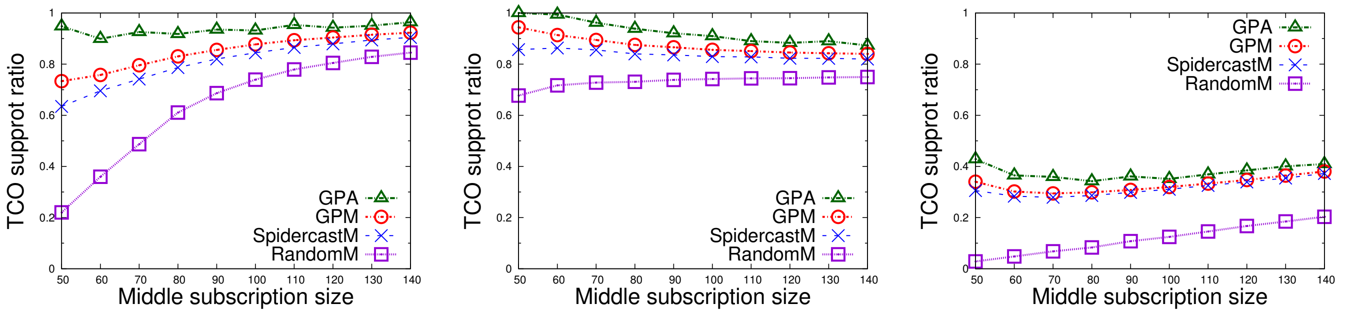
(a) Expo (b) Zipf (c) Unif

Figure 6: Partial TCO design algorithms wrt. $|V|$ where $|T| = 1000, |T(v)|_{\min} = 20, |T(v)|_{\max} = 600, d = 4, m = 2|V|$



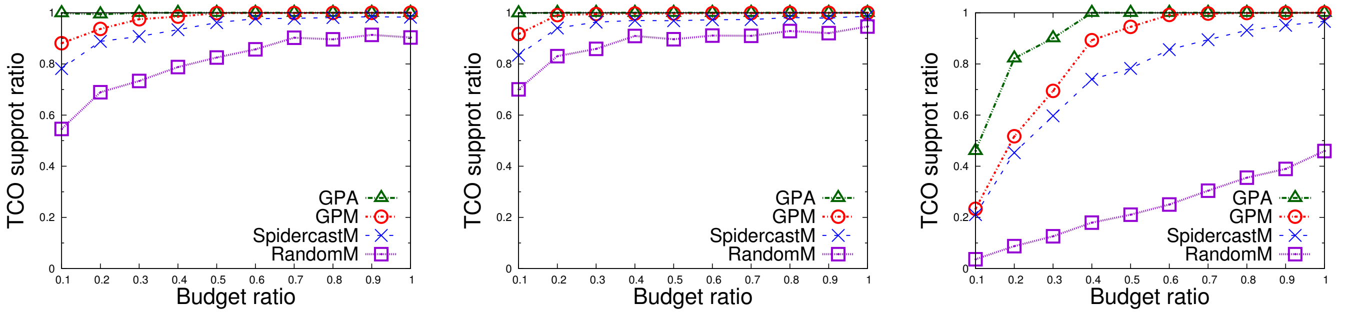
(a) Expo (b) Zipf (c) Unif

Figure 7: Partial TCO design algorithms wrt. $|T|$ where $|V| = 2000, |T(v)|_{\min} = 20, |T(v)|_{\max} = 600, d = 3, m = 3000$



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

Figure 8: Partial TCO design algorithms wrt. subscription size where $|V| = 2000, |T| = 1000, d = 4, m = 4000$



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

Figure 9: Partial TCO design algorithms wrt. β where $|T| = 1000, |T| = 1000, |T(v)|_{\min} = 20, |T(v)|_{\max} = 600$

$|T(v)|_{\min} = 20$, and $|T(v)|_{\max} = 600$. Besides, $d = 3$ and $m = \frac{d \times |V|}{2} = 3000$, where $d \approx 20\% \cdot \Delta(V, E_{\textsf{MinMaxODA}})$.

First, the TCO support decreases with the size of the topic set for all algorithms. For example, $TcoSuppR_{\textsf{GPM}}$ falls from 65.3% to 24.4% as $|T|$ increases from 1000 to 10,000. The reason is that increasing the number of topics reduces correlation among the nodes.

Second, both GPA and GPM show the Pareto principle under Expo and Zipf except Unif, because Unif is directly opposite to the Pareto distribution. Besides, algorithms are less sensitive to the number of topics under Zipf, for instance, $TcoSuppR_{\textsf{GPM}}$ only reduces by 0.39% when $|T|$ increases from 1000 to 10,000. In spite of expanding the topic set, the deep-rooted skewness of Zipf allows the algorithms to maintain the TCO support ratios to some extent.

Third, GPA, GPM and SpiderCastM output much higher TCO support than RandomM, since RandomM does not leverage correlation: $(TcoSuppR_{\textsf{GPM}} - TcoSuppR_{\textsf{RandomM}})$ is 26.7%, 22.0%, and 14.7% under Expo, Zipf, and Unif, respectively on average.

G. Impact of subscription size

Fig. 8 depicts how the subscription size influences the performances of the algorithms. We set $|V| = 2000$, $|T| = 1000$, and the subscription range $[|T(v)|_{\min}, |T(v)|_{\max}]$ to be $[10, 90], [20, 100], \dots, [100, 180]$. We define the *middle subscription size* as $|T(v)|_{\text{mid}} = \frac{|T(v)|_{\min} + |T(v)|_{\max}}{2}$, then $|T(v)|_{\text{mid}} \in [50, 140]$. Besides, we take $d = 4$ and accordingly $m = \frac{d \times |V|}{2} = 4000$, where $d \approx 20\% \times \Delta(V, E_{\textsf{MinMaxODA}})$.

First, raising subscription size may either increase or decrease $TcoSuppR$. As $|T(v)|_{\text{mid}}$ grows from 50 to 140, $TcoSuppR$ increases under Expo and Unif but decreases under Zipf. On the one hand, a larger subscription size enhances correlation across the nodes. Upon higher correlation, nodes share more common interests, an edge addition is expected to bring a higher contribution towards the TCO, and thus $\mu(E)$ is bigger. On the other hand, more subscriptions also imply a larger $\mu(K)$, i.e., more TCCs have to be reduced to attain a complete TCO. Therefore, the curve of $TcoSuppR(E) = \frac{\mu(E)}{\mu(K)}$ with regard to the subscription size depends on the relative weights of both factors under each workload.

Second, the gaps between different algorithms shrink as each node increases its subscription size. While enlarging the subscription size elevates the correlation among nodes, it also diminishes variance between edges. On the one side, edge contributions increase; on the other side, edges vary less in their contributions. As a result, it matters less which algorithm we employ for partial TCO construction. If $|T(v)| \rightarrow |T|, \forall v \in V$, then all algorithms will converge to the same $TcoSuppR$. In this extreme case, all nodes subscribe to all topics, all potential edges are always of equal contributions towards TCO, and thus our different edge selection strategies turn out to be identical.

X. Conclusion

We propose PTCOA and PTCOM for the design of pub/sub partial TCOs under node degree constraints. We develop two centralized algorithms, GPA and GPM, that approximate optimal solutions with constant bounds for these NP-complete problems. Extensive empirical evaluation shows that our designed algorithms possess good practicality and cost-effectiveness under real-world pub/sub workloads. The Pareto principle prevails in many practical input instances and the output overlays, which provides us a solid guidance for effectively building partial TCOs by separating the *vital few* from the *trivial many*. We compare our centralized algorithms with the fully decentralized SpiderCastM, demonstrating that decentralized protocols are of great potential impact in both research and practice for partial TCO design, and that a theoretically sound centralized algorithm is a valuable instrument to distributed algorithm designers.

References

[1] BitTorrent. http://www.bittorrent.com/.

[2] Hadoop. http://hadoop.apache.org/.

[3] IBM Watson IoT Platform. http://internetofthings.ibmcloud.com/.

[4] Message Queue Telemetry Transport. http://mqtt.org/.

[5] 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*.

[6] B. H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Commun. ACM*, 13(7):422–426, 1970.

[7] V. Bortnikov, G. Chockler, A. Roytman, and M. Spreitzer. Bulletin board: a scalable and robust eventually consistent shared memory over a peer-to-peer overlay. *Operating Systems Review*, 44(2):64–70, 2010.

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

[9] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber. Bigtable: A distributed storage system for structured data. In *OSDI '06*.

[10] C. Chen and Y. Tock. Design of routing protocols and overlay topologies for topic-based publish/subscribe on small-world networks. In *Middleware Industry '15*.

[11] C. Chen, Y. Tock, H.-A. Jacobsen, and R. Vitenberg. Weighted overlay design for topic-based publish/subscribe on geo-distributed data centers. In *ICDCS*, 2015.

[12] C. Chen, R. Vitenberg, and H.-A. Jacobsen. A generalized algorithm for publish/subscribe overlay design and its fast implementation. In *DISC'12*.

[13] 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*.

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

[15] 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.

[16] J. C. Corbett, J. Dean, M. Epstein, A. Fikes, C. Frost, J. J. Furman, S. Ghemawat, A. Gubarev, C. Heiser, P. Hochschild, W. Hsieh, S. Kanthak, E. Kogan, H. Li, A. Lloyd, S. Melnik, D. Mwaura, D. Nagle, S. Quinlan, R. Rao, L. Rolig, Y. Saito, M. Szymaniak, C. Taylor, R. Wang, and D. Woodford. Spanner: Google's globally-distributed database. In *OSDI'12*.

[17] R. Diestel. *Graph Theory*. Springer, 2006.

[18] U. Feige. A threshold of ln n for approximating set cover. *J. ACM*, 1998.

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

[20] H. Kwak, C. Lee, H. Park, and S. Moon. What is Twitter, a social network or a news media? In *WWW'10*.

[21] 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*.

[22] D. Malkhi, M. Naor, and D. Ratajczak. Viceroy: A scalable and dynamic emulation of the butterfly. In *PODC'02*.

[23] G. S. Manku, M. Bawa, and P. Raghavan. Symphony: Distributed hashing in a small world. In *USITS*, 2003.

[24] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. Openflow: Enabling innovation in campus networks. *SIGCOMM Comput. Commun. Rev.*, 2008.

[25] R. Melamed and I. Keidar. Araneola: A scalable reliable multicast system for dynamic environments. *J. Parallel Distrib. Comput.*, 2008.

[26] G. Nemhauser, L. Wolsey, and M. Fisher. An analysis of approximations for maximizing submodular set functions - I. *Mathematical Programming*, 14(1):265–294, 1978.

[27] M. Onus and A. W. Richa. Minimum maximum degree publish-subscribe overlay network design. In *Infocom'09*.

[28] M. Onus and A. W. Richa. Parameterized maximum and average degree approximation in topic-based publish-subscribe overlay network design. In *ICDCS'10*.

[29] J. A. Patel, E. Rivière, I. Gupta, and A.-M. Kermarrec. Rappel: Exploiting interest and network locality to improve fairness in publish-subscribe systems. *Computer Networks*, 2009.

[30] F. Rahimian, T. Le Nguyen Huu, and S. Girdzijauskas. Locality-awareness in a peer-to-peer publish/subscribe network. In *DAIS'12*.

[31] J. Reumann. GooPS: Pub/Sub at Google. Lecture at CANOE Summer School, Oslo, Norway, August, 2009.

[32] 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*.

[33] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: a scalable peer-to-peer lookup service for internet applications. In *SIGCOMM'01*.

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

[35] G. Urdaneta, G. Pierre, and M. V. Steen. Towards a fully decentralized and collaborative hosting infrastructure for Wikipedia. In *WikiSym'08*.

[36] C. Wilson, B. Boe, A. Sala, K. P. Puttaswamy, and B. Y. Zhao. User interactions in social networks and their implications. In *EuroSys'09*.

[37] L. A. Wolsey. An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica*, 1982.

APPENDIX

The approximation ratio of Alg. 3, GPM, relies on Lemma 5 and 6, which Onus and Richa proved in [27]. We rewrite their proofs here for the completeness and readability of this work.

Proof of Lemma 5: Given an PTCOM instance (V, T, I, d), suppose $G = (V, E_i)$ is the current overlay. Let M^* be the matching that achieves the highest contribution with respect to E_i, which we denote by c^*, i.e.,

$$c^* = \mu'(M^*|E_i) \tag{37}$$

Meanwhile, our algorithm GPM finds the matching $M_{i+1} = \{e_1, e_2, \ldots, e_k\}$ in Phase $(i+1)$, where e_l is the l-th edge added in this phase, i.e., GPM adds e_a before e_b, if $1 \le a < b \le k$. We denote by c the contribution of M_{i+1} with respect to E_i, i.e.,

$$c = \mu'(M_{i+1}|E_i) \tag{38}$$

Let $F_0 = E_i$ and $F_l = F_{l-1} \cup e_l$ for $1 \le l \le k$. Let y_l be the edge contribution of e_l with regard to F_{l-1}, i.e.,

$$y_l = \mu'(e_l|F_{l-1}), \forall 1 \le l \le k \tag{39}$$

Then,

$$c = \sum_{1 \le l \le k} y_l \tag{40}$$

$$y_a \le y_b, \forall 1 \le a < b \le k \tag{41}$$

Let X_l be the set of edges in M^* that are incident to u_l or v_l, $1 \le l \le k$, but not incident to $u_{l'}$ or $v_{l'}$, $1 \le l' \le l$. Let $P_0 = M^*$ and $P_l = P_{l-1} - X_l$ for $1 \le l \le k$. Since M_{i+1} is a maximal matching, $P_k = \emptyset$. Let x_l represent the contribution of X_l with regard to F_{l-1}, i.e.,

$$x_l = \mu'(X_l|F_{l-1}), \forall 1 \le l \le k \tag{42}$$

Let c_l^* be the contribution of P_l with regard to F_l, i.e.,

$$c_l^* = \mu'(P_l|F_l), 1 \le l \le k \tag{43}$$

$$c_0^* = c^*, c_k^* = 0 \tag{44}$$

Further, $\forall 1 \le l \le k-1$,

$$\mu'(P_l|F_l) = \mu'(X_l|F_l) + \mu'(P_{l+1}|F_l \cup X_l) \tag{45}$$

$$\le \mu'(X_l|F_l) + \mu'(P_{l+1}|F_l) \tag{46}$$

$$\le \mu'(X_l|F_l) + \mu'(P_{l+1} \cup e_{l+1}|F_l) \tag{47}$$

$$= \mu'(X_l|F_l) + \mu'(e_{l+1}|F_l) + \mu'(P_{l+1}|F_{l+1}) \tag{48}$$

$$\Longrightarrow c_l^* \le c_{l+1}^* + x_{l+1} + y_{l+1} \tag{49}$$

Eq. (45) is a telescoping sum that stems from submodularity and $P_l = P_{l+1} + X_{l+1}$. Eq. (46) also comes from submodularity. Eq. (47) follows the fact that adding an edge always leads to a non-negative contribution. Eq. (48) holds because $F_{l+1} = F_l \cup e_{l+1}$. Eq. (49) simply rewrites Eq. (48).

Summing up Eq. (44) and (49) over $0 \le l \le k-1$, we have

$$\sum_{1 \le l \le k} x_l + \sum_{1 \le l \le k} y_l \ge c^* \tag{50}$$

Now we compare the contribution of e_l against the contribution of X_l with regard to F_{l-1}. There are two cases:

(a) If X_l has 2 edges, then GPM choose neither of them and choose e_l at the l-th step, $1 \le l \le k$. Since GPM greedily selects the edges at each iteration, the contribution of e_l with regard to F_{l-1} is at least as much as each edge in X_l. Hence,

$$y_l \ge \frac{x_l}{2}, \text{ if } |X_l| = 2, \forall 1 \le l \le k \tag{51}$$

(b) If X_l contains 0 or 1 edges, with the same argument,

$$y_l \ge x_l, \text{ if } |X_l| \le 1, \forall 1 \le l \le k \tag{52}$$

Combining Eq. (51) and (52), we have

$$y_l \ge \frac{x_l}{2}, \forall 1 \le l \le k$$

$$\Rightarrow \sum_{1 \le l \le k} y_l \ge \frac{1}{2} \sum_{1 \le l \le k} y_l \tag{53}$$

With Eq. (50) and (53), we have

$$3 \sum_{1 \le l \le k} y_l \ge c^* \tag{54}$$

$$\Rightarrow c \ge c^*/3 \tag{55}$$

■

Proof of Lemma 6: Given $G = (V, F)$, Vizing' Edge Coloring Theorem [17] ensures that we can color the edge set F with $(\Delta(V, F) + 1)$ colors such that the colors of any adjacent edges are different. Since each coloring is a matching, we can always divide the edge set F into $(\Delta(V, F) + 1)$ matchings. ■