Brief Announcement: 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 pub/sub communication. We propose the MinAvg-kTCO problem parameterized by k: 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 MinAvg-2TCO, we present GM2, the first polynomial time algorithm with an approximation ratio. With regards to MinAvg-kTCO, where $k \geq 2$, we propose a simple and efficient heuristic algorithm, namely HararyPT, that aligns nodes across different sub-overlays.

We experimentally demonstrate the scalability of GM2 and HararyPT under representative pub/sub workloads.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*network topology*; G.2.2 [**Discrete Mathematics**]: Graph Theory—*network problems*

General Terms

Algorithms, Theory, Experimentation

Keywords

Overlay networks, reliability, publish/subscribe

1. INTRODUCTION

Publish/Subscribe (pub/sub) systems constitute an attractive choice as the communication paradigm and messaging substrate for building large-scale distributed systems. 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.

A distributed topic-based pub/sub system is often organized as an application-level overlay of brokers (e.g., simply referred to as nodes) connected in a federated or in a peer-to-peer manner. The overlay infrastructure directly impacts the pub/sub system's performance and scalability, e.g., the message routing cost. Constructing a high-quality broker overlay is a fundamental problem that has received attention both in industry and academia [4, 1].

Gregory Chockler *et al.* define a *topic-connected overlay* (TCO), as an overlay, where all pub/sub nodes interested in the same topic are organized in a connected dissemination sub-overlay [4]. 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. A TCO also results in more efficient routing protocols, a simpler matching engine, and smaller forwarding tables.

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 [4, 1]. 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 the 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 §3). 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. In practice, the set of active machines in a data center shows non-negligible variations over time.

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 [4, 1]. On the other hand, a few pub/sub systems (e.g., [3]) have

*A full version of this paper is available in [2].

PODC'13, July 22–24, 2013, Montréal, Québec, Canada.

explored the problem of dynamically maintaining the *TCO*. Basically, these approaches constantly make incremental adjustments to 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. 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. 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, we reduce the diameters of the overlay, as we improve its connectivity (see §6). With lower diameters, message delays are likely to be diminished because fewer hops are needed for message delivery.

Nevertheless, these merits of *kTCO* come at a price – additional links are required. 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 [4]). 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, wireless networks, and survivable network designs.

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

1. We propose the MinAvg-kTCO problem of devising *kTCO* with the minimum number of links (see Problem 1 in §3). We formally prove the hardness of MinAvg-kTCO in §3.
2. We design two algorithms for MinAvg-kTCO. First, with regards to MinAvg-*2*TCO, we present GM*2*, the first polynomial time approximation algorithm in §4. Second, with regards to MinAvg-kTCO, where $k \geq 2$, we propose a simple and efficient heuristic algorithm, namely HararyPT, that aligns nodes across different sub-overlays (see §5).
3. We validate GM*2* and HararyPT with comprehensive experiments under characteristic pub/sub workloads in §6. GM*2* outputs a *2TCO*, whose average node degree is around 1.5 times that of the *1TCO* produced by the state-of-the-art algorithm. GM*2* also improves the topic diameters by 50%.

2. BACKGROUND

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 $TCO(V, T, Int, E)$.

3. THE MINAVG-KTCO PROBLEM

The definition of a k-connected graph 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 there are no subscribers 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.

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

4. GM*2* ALGORITHM TO BUILD *2*TCO

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

GM*2*(V, T, Int)
Input: V, T, Int
Output: $2TCO(V, T, Int, E_{GM2})$
1: $E_{GM2} \leftarrow \emptyset$, $E_{pot} \leftarrow V \times V$
2: **while** $TPSO(V, T, Int, E_{GM2})$ is not *2TCO* **do**
3: **for all** $e = (v, w) \in E_{pot}$ **do**
4: $estimate(e, E_{GM2}) \leftarrow \big|\{t \in T | Int(v, t) \wedge Int(w, t) \wedge$ no *TC-block* in $G^{(t)}$ contains both v and $w\}\big|$
5: $e \leftarrow$ find e s.t. $estimate(e, E_{GM2})$ is max among E_{pot}
6: $E_{GM2} \leftarrow E_{GM2} \cup \{e\}$, $E_{pot} \leftarrow E_{pot} - \{e\}$
7: return $2TCO(V, T, Int, E_{GM2})$

For the MinAvg-*2*TCO problem, we devise Greedy Merge for the *2TCO* algorithm, GM*2* for short.

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-connected block*, *TC-block* for short.

As specified in Alg. 1, GM*2* starts with $TPSO(V,T,Int,E)$ where $E = \emptyset$. 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 $t \in T$.

We denote by P_i the set of edges added to the overlay after the i-th iteration of GM*2*. Line 4 of Alg. 1 defines the *estimate* of e's contribution on topic t: $estimate^{(t)}(e(v,w), P_i) =$

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

The overall *edge estimate* is defined as

$$estimate(e, P_i) = \sum\nolimits_{t \in T} estimate^{(t)}(e, P_i) \ .$$

LEMMA 1. *Alg. 1 takes time $O(|V|^4|T|)$ to output a 2TCO.*

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

5. HARARYPT TO BUILD kTCO

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

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

HararyPT$(I(V,T,Int),k)$
Input: $I(V,T,Int),k$
Output: $kTCO(V,T,Int,E_{\mathsf{HPT}})$
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: $E_{\mathsf{HPT}} \leftarrow \bigcup_{t \in T} E^{(t)}$
5: return $kTCO(V,T,Int,E_{\mathsf{HPT}})$

HararyPT stems from graph theory about 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 $\mathbb{V}^{(t)}$.

In order to promote edge sharing across different sub-overlays, Alg. 2 first obtains a node sequence for all the nodes in Line 1. Then Alg. 2 adopts the same linear ordering for all Harary constructions across all topics (Lines 2-3). By sharing the determined node sequence, these Harary graphs are likely to share a lot of edges, especially when the workloads are highly correlated. As a consequence, the output *kTCO* tends to have a low node degree.

6. EVALUATION

We implemented GM*2*, 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 [4]. 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 *2TCO*).

We set $|V| \in [100, 1\,000]$, $|T| = 200$, and each node has a fixed subscription size of 30. 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 $p(t)$. The value of $p(t)$ is distributed according to either an exponential, a Zipfian, or a uniform distribution, which we call Expo, Zipf, or Unif, for short. These distributions are representative of actual workloads used in industrial pub/sub systems today [3].

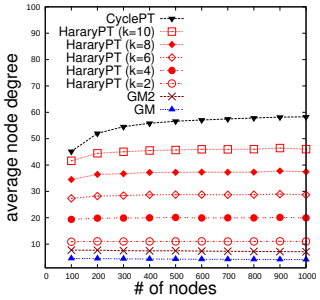

Figure 1: Node degree – Expo

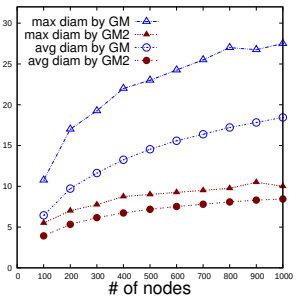

Figure 2: Topic diameter – Unif

Fig. 1 compares the average node degrees in the output overlays produced by different algorithms under Expo. For a specific algorithm $\mathcal{A}$, we denote by $\overline{d}_{\mathcal{A}}$ the average node degree produced by $\mathcal{A}$. We focus on GM*2* in Fig. 1. First, $\overline{d}_{\mathsf{GM}2}$ and $\overline{d}_{\mathsf{GM}}$ are quite close: $\overline{d}_{\mathsf{GM}2}$ is smaller than $1.65 \cdot \overline{d}_{\mathsf{GM}}$, on average. Second, $\overline{d}_{\mathsf{CyclePT}}$ is about 5 times higher than $\overline{d}_{\mathsf{GM}2}$ and tends to increase with the number of nodes, while $\overline{d}_{\mathsf{GM}2}$ 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 GM*2* and GM to find neighbors with more interest overlap, thus reducing overall number of neighbors needed.

Fig. 1 also shows that 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.

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 $G^{(t)} = (V^{(t)}, E^{(t)})$. We denote the maximum and average topic diameter across all topics as *Diam* and $\overline{diam}$, respectively. Fig. 2 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 gaps of $(Diam_{\mathsf{GM}} - Diam_{\mathsf{GM}2})$ and $(\overline{diam}_{\mathsf{GM}} - \overline{diam}_{\mathsf{GM}2})$ grow as the input instances scale up.

7. REFERENCES

[1] C. Chen, R. Vitenberg, and H.-A. Jacobsen. A generalized algorithm for publish/subscribe overlay design and its fast implementation. In *DISC*, 2012.

[2] C. Chen, R. Vitenberg, and H.-A. Jacobsen. Constructing fault-tolerant overlay networks for topic-based pub/sub. Technical report, U. of Toronto & U. of Oslo, 2013. http://msrg.org/papers/TR-kTCO.

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

[4] G. Chockler, R. Melamed, Y. Tock, and R. Vitenberg. Constructing scalable overlays for pub-sub with many topics: Problems, algorithms, and evaluation. In *PODC*, 2007.