Optimizing Event Distribution in Publish/Subscribe Systems in the Presence of Policy-Constraints and Composite Events[*]

Weifeng Chen[1], Zihui Ge[2], Jim Kurose[1], Don Towsley[1]

[1]Department of Computer Sciences
University of Massachusetts Amherst, MA 01003
{chenwf, kurose, towsley}@cs.umass.edu

[2]AT&T Labs-Research
Florham Park, NJ 07932
gezihui@research.att.com

Abstract

In the publish/subscribe paradigm, information is disseminated from publishers to subscribers that are interested in receiving the information. In practice, information dissemination is often restricted by policy constraints due to concerns such as security or confidentiality agreement. Meanwhile, to avoid overwhelming subscribers by the vast amount of primitive information, primitive pieces of information can be combined at so-called brokers in the network, a process called composition. Information composition provides subscribers the desirable ability to express interests in an efficiently selective way.

In this paper, we formulate the Min-Cost event distribution problem in pub/sub systems with policy constraints and information composition. Our goal is to minimize the total cost of event transmission while satisfying policy constraints and enabling information composition. This optimization problem is shown to be **NP***-complete. Our simulation study shows that our heuristics work efficiently, especially in a policy-constrained system. We also find that by increasing the number of broker nodes in a pub/sub system, we are able to reduce the total cost of event delivery.*

1. Introduction

In a content-based publish/subscribe (pub/sub) system [5, 19], publishers advertise information throughout the network. Subscribers express their interests in receiving particular information through a subscription process that establishes communication channels between publishers and interested subscribers. Following the subscription process, only information wanted by subscribers is disseminated. Distributing information from a publisher to a subscriber always incurs a *transmission cost*. Given a fixed amount of information to transmit from a publisher to a set of subscribers, there may exist multiple routes, each of which may have a different transmission cost. Minimizing the transmission cost in this case is related to the Minimum Steiner Tree problem, a problem that has been well studied [21]. We concern ourselves in this paper with minimizing transmission costs in a pub/sub system with two new considerations being taken – *policy constraints* and *information composition*.

Policy often plays an important role in information dissemination. Information flows on the Internet are restricted by BGP policies among different transit domains [15]. Policy constraints are also applied in information flow security models to prevent unauthorized flow of sensitive information [14]. For example, in the Bell-LaPadula model [4], information can only be allowed to flow from a principal with a lower security level to one with a higher level. The goal of the Transnational Digital Government (TDG) project [10] is to build a pub/sub system among the Organization of American States (OAS), Belize and the Dominican Republic (DR). In this project, sensitive bilateral information between Belize and DR is not allowed to transit through a node controlled by OAS.

A unit of information in a content-based pub/sub system is referred to as an *event* [5]. *Primitive events* are predefined atomic events that are typically generated directly by publishers. Hotel room prices, airline ticket prices and a weather forecast for a particular day are all primitive events published by different Web services. *Composite events* are formed by composing a set of primitive events and/or other composite events through operations such as disjunction, conjunction, sequence, iteration, and negation [7]. Composite events provide subscribers the ability to express interests in a flexible and sophisticated way and avoid being overwhelmed by a large number of primitive events [17]. Consider the following example. Alice lives in Illinois and is planning a Christmas vacation. She has multiple options, and her decision depends on flight prices, hotel availability and weather conditions. Alice may subscribe to each of these three primitive events but then would receive all hotel, flight and weather information, almost all of which will not be of interest to her. Instead, she prefers that these three

[*]This research has been supported in part by the NSF under grant awards EIA-0131886 and EIA-0080119. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

primitive events be merged somewhere in the pub/sub system so that she only receives information about a composite event in which flight's price is less than four hundred dollars, three-night hotel room is available, and the weather is perfect. In this example, the three primitive events are called the *component events* of the resultant composite event.

In this paper, we formulate the Min-Cost (minimum-cost) event distribution problem in a pub/sub system with policy constraints and event composition. Our goal is to find the event transmission solution with minimum transmission cost that satisfies policy constraints and enables event composition. This Min-Cost problem is shown to be an **NP**-complete problem through the reduction from the Minimum Steiner Tree problem. A greedy heuristic is then proposed, whose performance is evaluated via simulation. Our study shows that a greedy heuristic performs quite well, with all approximate solutions falling within 1.08 of the optimal solutions in the cases considered. A further enhanced algorithm based on the greedy heuristic improves the performance to fall within 1.012 of the optimal solutions. As more policy constraints are applied, the approximate solutions become even closer to the optimal solutions. We also find that, by increasing the number of brokers in a pub/sub system, we are able to further reduce the total cost of event delivery.

The remainder of this paper is organized as follows. In Section 2, we describe the system models and formulate the Min-Cost problem. Section 3 analyzes the complexity of the Min-Cost problem, and then presents heuristic algorithms to solve it. The performance of the heuristic is evaluated in Section 4. Section 5 discusses several additional issues and presents related work. Finally, we conclude the paper in Section 6.

2 System Model and Problem Formulation

We first describe our model of the pub/sub system in the presence of policy constraints and composite events. Then we formally define the Min-Cost event distribution problem.

2.1. Model and notation

A pub/sub system can be built at the network layer using active routers as brokers [22] or at the application-layer where both publishers, subscribers and brokers are end-hosts [11, 18]. In either case, we model the system as follows.

2.1.1 Network

We represent the distribution topology of the pub/sub system as a directed weighted graph $G = (V, L, C)$. Here V is the set of active functional components, i.e., $V = P \cup S \cup B$ where P, S and B are the set of publishers, subscribers and brokers respectively. $L \subseteq V \times V$ is the set of directed links (either physical or logical, depending on the way a pub/sub system is implemented), and $C : L \mapsto R^+$ is the weight function associated with L — $C(u, v)$ captures the cost of transmitting a unit rate event-flow over link $(u, v) \in L$. Note that, for a network-layer pub/sub system, the distribution topology G overlays the physical connectivity graph, while for an application-layer pub/sub system, G is close to a full mesh, since any two nodes should be able to connect to each other, except for restrictions such as NATs and firewalls.

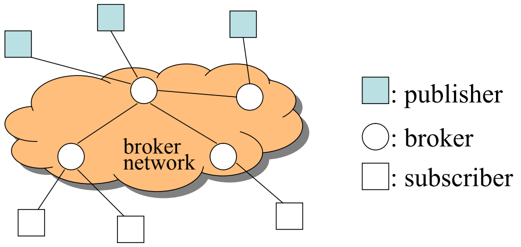


Figure 1. A pub/sub network infrastructure.

2.1.2 Events

We consider two types of events – primitive events and composite events in our framework. Primitive events refer to the raw events generated directly from publishers and composite events are the integrated events generated through logical operations from the component events, which can itself be a composite event or a primitive event.

Let E_p be the set of primitive events generated by the publishers. The *source* function $\mathcal{S} : P \times E_p \mapsto \{0, 1\}$ indicates the source of primitive events:

$$\mathcal{S}(p, e) = \begin{cases} 1, & \text{publisher } p \text{ generates primitive event } e \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

Based on a subscriber's interests, composite events are generated by brokers at nodes "inside" the network (see Figure 1). We assume that a broker is able to generate a composite event given that the broker receives all of the component events. Typically, composite events are generated by applying different logic operations (e.g., conjunction, negation) on their component events. We ignore these logic operations since they are event-specific. Instead, we represent the relationship between a composite event and its component events through a *composition matrix* $\mathcal{M}$. More precisely, let E_c be the set of composite events and $E = E_p \cup E_c$ be the set of all events. $\mathcal{M}_{E \times E}$ is defined as:

$$\mathcal{M}(d, e) = \begin{cases} 1, & \text{event } d \text{ is a component event of } e \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

For completeness, we define that each primitive event is a component event of itself while no composite event is a component event of itself. Consequently, $\mathcal{M}(e, e) = 1$ for all $e \in E_p$ and $\mathcal{M}(e, e) = 0$ for all $e \in E_c$.

Each event in the pub/sub system has an associated data rate. Let $\lambda(e)$ denote the *generation rate* for each $e \in E$. Generally, a composite event should have a rate that is no more than the aggregated rate of its component events, i.e., $\lambda(e) \leq \sum_{d:\mathcal{M}(d,e)=1} \lambda(d)$. This is because there is no additional information introduced during event composition.

With the event generation rate defined, the transmission cost for event e over link (u, v) is thus $C(u, v)\lambda(e)$.

2.1.3 Subscription interests

Subscribers' interests are represented by the *interest function* $\mathcal{I} : S \times E \mapsto \{0, 1\}$, where

$$\mathcal{I}(s, e) = \begin{cases} 1, & \text{subscriber } s \text{ is interested in event } e \\ 0, & \text{otherwise} \end{cases}$$

A pub/sub system must satisfy the no-false-exclusion requirement [1]. That is, a subscriber interested in receiving an event must receive that event.

2.1.4 Event distribution policies

Events are transmitted among functional entities (i.e., V in G). However, due to particular purposes (e.g., security), there are often restrictions on event distribution between two entities. For instance, in the previous example of the TDG project, a broker in Belize is not allowed to deliver sensitive bilateral information with DR to a broker owned by or located in OAS. To incorporate such restrictions, we introduce a matrix for event distribution policies. More specifically, an *event-distribution policy* $\mathcal{P} : L \times E \mapsto \{0, 1\}$ is defined as

$$\mathcal{P}((u, v), e) = \begin{cases} 1, & e \text{ is allowed to transmit from } u \text{ to } v \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

2.2 Problem formulation

Having defined the necessary notation, we now present a formal formulation of the *MIn-Cost event distribution problem in the presence of PolIcy-constraints and Composite Events* (MICPICE). Our objective is to find a transmission scheme such that the cost of delivering the requested events to the subscribers, while satisfying the policy constraints, is minimized. More formally, we have the following definition.

Definition 2.1: *Given $G = (V, L, C)$, $\mathcal{S}$, $\mathcal{M}$, $\mathcal{I}$, λ and $\mathcal{P}$, the MICPICE problem is to find a transmission assignment $\eta : L \times E \mapsto \{0, 1\}$, where*

$$\eta((u, v), e) = \begin{cases} 1, & e \text{ is delivered from } u \text{ to } v \\ 0, & \text{otherwise} \end{cases}$$

so as to minimize

$$\textstyle\sum_{e \in E} \sum_{(u,v) \in L} \lambda(e) C(u, v) \eta((u, v), e) \quad (4)$$

subject to

(a) feasibility: $\forall (u, v) \in L, \forall e \in E$,

$$\eta((u, v), e) \le \min\{X(u, e), 1\} \quad (5)$$

(b) no-false-exclusion: $\forall s \in S, \forall e \in E$,

$$\mathcal{I}(s, e) \le X(s, e) \quad (6)$$

(c) policy constraint: $\forall (u, v) \in L, \forall e \in E$,

$$\eta((u, v), e) \le \mathcal{P}((u, v), e) \quad (7)$$

(d) path constraint: $\forall u \in V, \forall e \in E$,

$$\begin{aligned} X(u, e) = & \max_{(v,u) \in L} \eta((v, u), e)[X(v, e) - 1] \\ & + N\mathcal{S}(u, e) + N \min_{d:\mathcal{M}(d,e)=1} X(u, d) \end{aligned} \quad (8)$$

Here $N = |V|$ is a large constant. $X(u, e)$ is a control variable indicating whether an event e is available at node u under the transmission assignment η — $X(u, e) \ge 1$ if node u either generates event e or receives event e from a neighboring node; otherwise, $X(u, e) < 1$.

Intuitively, condition (5) states the feasibility constraint – a node can transmit an event to its neighbor only when the event is available at the node. Condition (6) specifies that every subscriber must receive all events that it subscribes to. Condition (7) guarantees that the transmission of an event does not violate the policy constraints. Condition (8) guarantees that $X(u, e) \ge 1$ if and only if node u generates e or there exists a transmission path from a node that generates e to node u.

It may not be straightforward to see that equation (8) captures the condition described above. We now show that it indeed does. It is easy to see that when node u is the publisher of event e ($\mathcal{S}(u, e) = 1$), or when u receives all the component events of e ($\min_{d:\mathcal{M}(d,e)=1} X(u, d) \ge 1$), $X(u, e) \ge N \ge 1$. When u does not generate e, however there is a transmission path $(v_1, v_2, \cdots, v_i, u)$ such that node v_1 generates e ($X(v_1, e) \ge N$) and $\eta((v_1, v_2), e) = \eta((v_2, v_3), e) = \cdots = \eta((v_i, u), e) = 1$), from (8), we know $X(u, e) \ge X(v_i, e) - 1 \ge \cdots \ge X(v_1, e) - i \ge N - i \ge 1$. This is because $X(\cdot, e)$ decreases by at most 1 for each hop through which e is transmitted and the number of hops is bounded by the total number of nodes except u, i.e., $i \le N - 1$.

To establish that $X(u, e) \ge 1$ is a sufficient condition for event e being available at node u, we provide a proof by contradiction. Assume node u is the one with the largest $X(\cdot, e)$ value among all the nodes at which e is unavailable. Equation (8) becomes $X(u, e) = \max_{(v,u) \in L} \eta((v, u), e)[X(v, e) - 1]$. If $X(u, e) \ge 1$, there exists at least one node v such that $\eta((v, u), e) = 1$ and $X(v, e) = X(u, e) + 1$. If e is available at v, it should also be available at u (since $\eta((v, u), e) = 1$), conflicting with the assumption. On the other hand, if e is unavailable at v, we have $X(v, e) = X(u, e) + 1 > X(u, e)$, conflicting with u having the maximum value of $X(\cdot, e)$. Thus, we conclude that condition (8) guarantees that $X(u, e) \ge 1$ if and only if e is available at u under the transmission scheme η. In all, conditions (5)-(8) restrict η to be a valid transmission scheme.

Note that the quadratic term $\eta((v,u),e)[X(v,e)-1]$ in (8) can further be removed by the following transformation:

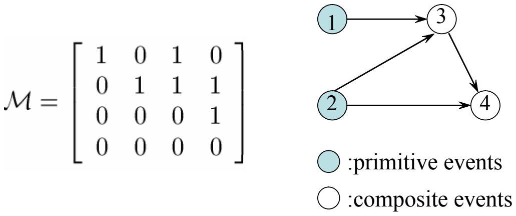
$$\eta((v,u),e)[X(v,e)-1] = \\ \max\{X(v,e)-1-N[1-\eta((u,v),e)],0\}$$

By doing so, Definition 2.1 becomes an integer linear programming formulation.

2.3 Feasibility of event distribution

Due to policy constraints enforced on event distribution, a MICPICE problem may not have a feasible solution. For a subscriber to receive a primitive event of interest, there should exist a path from the publisher generating the event to the subscriber under the policy constraint. In the case when a subscriber subscribes to a composite event, the policy constraint should allow a path from each publisher generating a component event to the subscriber. In addition to the existence of these paths, the MICPICE problem also requires a valid event composition matrix to be feasible. An event composition matrix $\mathcal{M}$ (defined in (2)) is *valid* if and only if the corresponding composition graph is acyclic and the in-degree of every composite-event node is greater than zero.

Definition 2.2 *A* composition graph *corresponding to a composition matrix $\mathcal{M}$ is a directed graph $G_{\mathcal{M}} = (V_{\mathcal{M}}, L_{\mathcal{M}})$, where the vertex set $V_{\mathcal{M}}$ consists of the events and a directed edge from node d to e exists if and only if $\mathcal{M}(d,e) = 1$ for $d \neq e$.*



$$\mathcal{M} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Figure 2. A composition matrix and its corresponding graph.

Figure 2 shows an example of a valid composition matrix and its corresponding graph. A valid composition matrix guarantees that there are no circular dependencies, e.g., there cannot exist event d and e such that $\mathcal{M}(d,e) = 1$ and $\mathcal{M}(e,d) = 1$.

A primitive event e' is called an *original event* of event e if there is a directed path from e' to e in the composition graph $G_{\mathcal{M}}$. For example, in Figure 2, events 1 and 2 are the original events of event 3. They are also the original events of event 4.

With these definitions, we have the following proposition.

Proposition 2.3 *The MICPICE problem have a feasible solution if and only if (1) composition matrix $\mathcal{M}$ is valid, and (2) if $\mathcal{I}(s,e) = 1$, there exists a path from the publisher generating e' to s for each original event e' of e under the policy constraint.*

Given an event composition matrix and a set of event distribution policies, the algorithm described in the next section is able to find a feasible event distribution scheme if exists, and properly terminates if it does not.

3 Solving the MICPICE Problem

In this section we first prove that the MICPICE problem is an **NP**-complete problem. We then present heuristic algorithms to approximately solve this problem.

3.1 Complexity analysis of the MICPICE problem

We begin with the Minimum Steiner Tree problem (MST), which is a well-known **NP**-hard problem. Given a connected graph with weights associated with edges and a subset of vertices, the MST problem is to find the minimum-weight subtree in the graph that includes all of the vertices in the subset [21].

Notice that a valid distribution of a primitive event from a publisher to its subscribers corresponds to a Steiner tree connecting the publisher and the set of subscribers. The published event is simply transmitted along the tree from the publisher to the subscribers. Based on this observation, we can prove that the MICPICE problem is **NP**-complete.

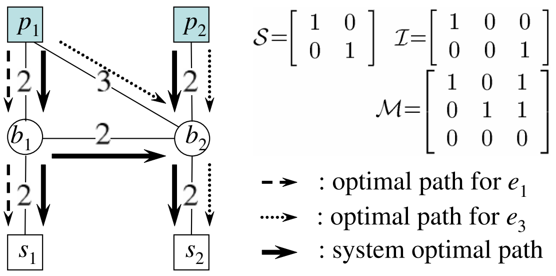
Theorem 3.1: *The MICPICE problem is* **NP**-*Complete.*

Proof (sketch): The MST problem is a special case of the MICPICE problem, when: (i) there exists a single primitive event; (ii) all links are allowed to transmit the event; and (iii) the union of the publisher generating the event and the subscribers subscribing to the event corresponds to the subset of vertices.

3.2 Optimal solution of a MICPICE problem

If there are no composite events in the MICPICE problem, i.e., $E_c = \emptyset$, then it can be decomposed into independent sub-problems, one for each primitive event. In other words, the optimal assignment η is the aggregate of the optimal transmission scheme of each individual primitive event, which is equivalent to finding the minimum Steiner tree connecting to the publisher and the set of interested subscribers in the reduced topology – some edges may be removed due to policy constraints.

However, once composite events are considered, the above property of independence disappears – the composition and transmission of composite events depends on the transmission of their component events. Thus a MICPICE problem can no longer be decomposed into independent subproblems. For example, consider a simple pub/sub system shown in Figure 3, where each link is annotated with its cost. The optimal solution for the system, shown in bold



p_1
p_2
b_1
b_2
s_1
s_2
2
3
$\mathcal{S}=\begin{bmatrix}1&0\\0&1\end{bmatrix}$
$\mathcal{I}=\begin{bmatrix}1&0&0\\0&0&1\end{bmatrix}$
$\mathcal{M}=\begin{bmatrix}1&0&1\\0&1&1\\0&0&0\end{bmatrix}$
: optimal path for e_1
: optimal path for e_3
: system optimal path

Figure 3. Transmission solutions of a pub/sub system.

lines, is not the aggregate of the optimal solution for e_1 and e_3 individually. As a consequence, existing approximate algorithms ([2, 3, 13]) for the minimum Steiner tree problem are not sufficient for the MICPICE problem. Instead, we develop a heuristic algorithm that takes into account the interaction between the transmission of composite events and their component events, as we show next.

3.3 MICPICE heuristics

We first present a greedy heuristic that is easy to understand, but without accounting for the dependency between the transmission of composite events and their component events. An enhancement on the greedy heuristic is then presented to include such consideration.

3.3.1 Greedy heuristic

The idea of the greedy heuristic is as follows. The algorithm looks at one event at a time (which we will refer to as a round) for dissemination. Consider an event e to be distributed. e is only allowed to be transmitted along a subset of links in the network $G = (V, L, C)$ due to its policy requirement. The *eligible route* of event e is a cut graph G_e of G by deleting edges (u, v) where $\mathcal{P}((u, v), e) = 0$. That is, G_e includes only the edges along which e is allowed to be transmitted. If e is a primitive event, e will be transmitted along the shortest paths (cost-wise) in G_e from its publisher. If e is a composite event, e may be generated by one or more broker nodes, each of which receives all component events of e. We refer to these nodes as the *source nodes* of e, denoted by M_e. The distribution of composite event e is rooted from one or more of the source nodes in M_e. Depending on the receiving node u, the greedy heuristic will select the source node and transmission path with the minimum overall cost: $\min_{v \in M_e} c_e(v) + d(v, u)\lambda(e)$, where $c_e(v)$ is the cost of receiving all component events of e at node v, and $d(v, u)$ is the shortest path length from v to u in G_e.

Recall that in each round the heuristic chooses one event and determines the distribution paths of the event to the brokers and subscribers. When e is chosen, the heuristic needs to know all possible source nodes of e — M_e, and their associated cost $c_e(\cdot)$. Thus, it is necessary to determine the transmission paths of all of the component events before considering the composite event. This is achieved by selecting the events in a *topological order*.

Definition 3.2 *A* topological order *of events is a permutation T of the vertices in the composition graph $G_{\mathcal{M}}$ such that a directed edge (d, e) implies that d appears before e in T.*

For example, (1,2,3,4) and (2,1,3,4) are the topological orders of the events whose composition graph is shown in Figure 2.

Recall that we require any event composition matrix $\mathcal{M}$ to be valid, which requires its corresponding composition graph $G_{\mathcal{M}}$ to be acyclic. Thus a topological order of events exists for any valid $\mathcal{M}$.

After obtaining the topological order of the events, we can apply a Dijkstra-like procedure to get the shortest paths for each event e on its eligible route G_e. Figure 4 presents the pseudo code of this procedure.

```
1. for each $e$ sorted in topological order
       //initialized link cost $d(u, v)$ in $G_e$
2.     if $\mathcal{P}((u, v), e) = 1$
          $d(u, v) = C(u, v)$;
       else
          $d(u, v) = \infty$;
       //initialize node cost $c_e(v)$ for Dijkstra
3.     for each node $v$
          if $e$ is primitive event
             $c_e(v) := 0$ for publisher $v$;
             $c_e(v) := \infty$ otherwise;
          else        // e is composite event
             $c_e(v) := \sum_{\mathcal{M}(d,e)=1} c_d(v)$;
          predecessor$(e, v) = v$;        // point to itself
       // run Dijkstra-like procedure
4.     changed = false;
       for each edge $(u, v)$
5.        if $c_e(u) + \lambda(e) * d(u, v) < c_e(v)$
             $c_e(v) = c_e(u) + \lambda(e) * d(u, v)$;
             predecessor$(e, v) = u$;
             changed = true;
6.     repeat 4 if changed
```

Figure 4. Pseudo code of the greedy heuristic.

After the pseudo code above determines the predecessor(e, v) for each event e and node v, the transmission solution η can be obtained by tracing the predecessors. For each event e, the tracing procedure starts from the set of subscribers who are interested in e. Figure 5 shows the pseudo code of this tracing procedure.

Proposition 3.3 *The greedy heuristic finds a feasible solution, if one exists, of the MICPICE problem.*

The proof is straightforward. For a particular event, the greedy heuristic is able to find all possible source nodes due to the topological order of events. Then Dijkstra's algorithm can be used to find a shortest path if one exists [8].

```
for each $\mathcal{I}(s, e) = 1$
    Trace-Predecessor($e$, $s$);

Trace-Predecessor($e$, $v$)
{
    $u$=predecessor($e$, $v$);
    while ($u! = v$)
        if $\eta((u, v), e) == 0$
            $\eta((u, v), e) = 1$;
            Trace-Predecessor($e$, $u$);
    if $e$ is a composite event
    // decompose e and recursively trace the predecessor of the component events
        for each $d$ s.t. $\mathcal{M}(d, e) = 1$
            Trace-Predecessor($d$, $u$);
}
```

Figure 5. Tracing predecessors to obtain η.

3.3.2 Enhancements on the greedy heuristic

The greedy heuristic described above does not account for the interaction between the transmission of composite events and their component events. For example, when applied to the pub/sub system in Figure 3, the heuristic returns the dotted routes as the transmission solution for e_3. This greedy heuristic can be enhanced to perform better.

The idea behind the enhancements is quite simple. Consider the example in Figure 3 again. Because of the no-false-exclusion requirement, e_1 must be delivered to s_1. Once e_1 is determined to be delivered along (p_1, b_1) and (b_1, s_1), nodes b_1 and s_1 receive e_1 and consequently can "generate" e_1. This observation is reflected in the algorithm by having the receiving cost of e_1 for b_1 and s_1 to be 0, i.e., $c_{e_1}(b_1) = c_{e_1}(s_1) = 0$, after $\eta((p_1, b_1), e_1)$ and $\eta((b_1, s_1), e_1)$ are set to 1. Assigning $c_{e_1}(b_1)$ and $c_{e_1}(s_1)$ to be 0 will contribute to the further process of delivering e_3. Specifically, the heuristic will assign predecessor(e_1, b_2) to be b_1 rather than p_1 since delivering e_1 from b_1 to b_2 is now a better choice.

The pseudo code of this enhancement algorithm is presented in [6] due to space limitation. We would like to address here the following clarification of the enhanced heuristic.

Order of choosing subscribers Since the enhanced algorithm considers the transmission paths of an event e to subscriber v based on previously determined transmission paths of other subscribers and events, different orders in which the algorithm considers the events and the subscribers can result in different overall cost. Consider the example in Figure 6. Assume that p_1 needs to transmit event e to both s_1 and s_2. Figure 6(a) shows the solution of considering e in the order of (s_1, s_2) whereas Figure 6(b) shows the solution in the order of (s_2, s_1). In the latter case, after e is set to transmit via b_1 to s_2, $c_e(b_1)$ becomes zero, which changes the predecessor of s_1 to b_1.

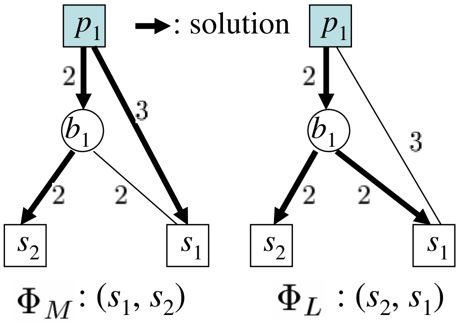


Figure 6. Different orders of choosing subscribers.

We consider three different orders in our study. In particular, we consider the order of minimum cost (denoted as Φ_M), the order of longest distance (Φ_L), and the random order (Φ_R). The random order Φ_R is simply that the heuristic randomly chooses subscribers. Using Φ_M, the heuristic chooses the subscriber that has a minimum receiving cost. The solution in Figure 6(a) is achieved using Φ_M since s_1 has a smaller receiving cost compared to s_2. When using Φ_L for event e, the heuristic chooses the subscriber who has the longest distance to M_e, the set of source nodes of e[1]. Recall that a source node of a primitive event is the corresponding publisher and a source node of a composite event is a node who have received all of the component events. In the case there exist no source nodes, the heuristic follows the order of Φ_M. The intuition behind Φ_L is that, once e is set to transmit along the path with the longest distance, we have the maximum increase of the size of M_e, i.e., all the nodes along the delivery path become the source nodes of e. As a consequence, other subscribers have a better chance to have a smaller distance to M_e. Actually, the solution in Figure 6 (b) is achieved using Φ_L. In particular, after event e is delivered to s_2, M_e becomes $\{p_1, b_1, s_2\}$ and s_1 has a smaller distance of 2 to M_e.

Let us consider the example in Figure 3 again. The enhanced heuristic updates predecessor(e_1, b_2) from p_1 to be b_1 after $\eta((p_1, b_1), e_1)$ is set to 1, returning the optimal solution of this example.

4 Evaluation

In this section, we use simulation to evaluate the performance of our algorithms.

4.1 Transmission costs

When a pub/sub system is implemented at the application layer, link cost $C(u, v)$ corresponds to the cost of a logical link (u, v). It is possible that two different logical links share some physical links. For example, in Figure 7, nodes b_1, b_2

[1] The distance from node u to the set of source nodes M_e is defined as $d(u, M_e) = \min_{v \in M_e} d(u, v)$, where $d(u, v)$ is the distance between nodes u and v.

and b_3 are application nodes whereas X is a physical router. Logical paths (b_1, b_2) and (b_1, b_3) share a common physical link (b_1, X). In this case, when an event e is delivered from b_1 to b_2 and to b_3 separately, the transmission cost is calculated as $\lambda(e)(C(b_1, b_2) + C(b_1, b_3))$ based on equation (4) in Definition 2.1. In this case the cost of link (b_1, X) is counted twice. This assumes that router X is not multicast capable. The simulation study presented below implements the pub/sub system at the application layer and follows this assumption. We will discuss the case when physical routers are multicast capable in Section 5.

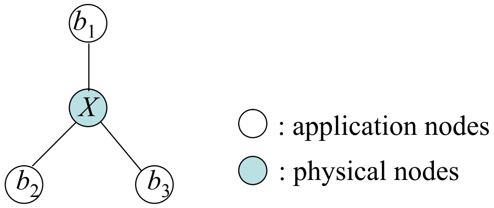


Figure 7. An application topology overlays a physical topology.

4.2 Simulation setting

We first use the Georgia Tech topology generator [20] to generate a physical network topology $G_P(V_P, L_P, C_P)$, with either a Flat model or a Transit-Stub model, as described below. All physical links have a link cost of one, i.e., $C_P(u, v) = 1$ for all $(u, v) \in L_P$. Given G_P, we randomly pick a set of publishers (P), subscribers (S) and brokers (B), and form an application-level network $G = (V, L, C)$ such that $(u, v) \in L$ if and only if the underlying shortest path from u to v does not go through any other application nodes. The cost $C(u, v)$ of a logical link $(u, v) \in L$ is the length of the shortest path between u and v in the underlying G_P. Other inputs to the MICPICE problem are generated taking the following into account:

- Each publisher generates a unique primitive event, i.e., $|E_p| = |P|$. We construct the composition graph as follows. Events are divided into different levels such that the first level consists of primitive events. There are no directed edges between events in the same level. All edges are from events in a lower level to those in a higher level. Recall that there is a directed edge from event d to e ($d \neq e$) if and only if $\mathcal{M}(d, e) = 1$. Each composite event requires at least one and at most three component events. Last, there are total three levels of events in the simulation we considered. Any composition graph constructed by this process is guaranteed to be valid.
- Each event is classified as either popular (with probability α) or unpopular (with probability $1 - \alpha$) [1]. A subscriber is interested in a popular event with probability p_{pop} and in an unpopular event with probability p_{unp}.
- Each event e is assigned a generation rate $0 < \lambda(e) < 1$. Since a composite event is composed of a set of component events, $\lambda(e) \leq \sum_{d:\mathcal{M}(d,e)=1} \lambda(d)$, i.e., the rate of a composite event is less than the sum of the rates of all of its component events.
- Policy constraints are generated as follows. Each directed edge $(u, v) \in L$ independently has a probability p_c that restricts the transmission of an event along that edge, i.e., $\Pr[\mathcal{P}((u, v), e) = 0] = p_c$ for $\forall e \in E$, $\forall (u, v) \in L$. As described earlier, $\mathcal{P}((u, v), e) = 0$ means that event e is not allowed to transmit along link (u, v).

4.3 Simulation results

4.3.1 Comparison of heuristic cost and optimal cost

We begin by comparing the transmission cost returned by the greedy heuristic to the optimal cost for small problem sizes. In the previous section, we described a straightforward greedy heuristic that does not account for the interaction between the transmission of composite events and their component events, followed by an enhancement that accounts for the interaction. We will refer to the former and the latter as the non-enhanced and enhanced heuristic, respectively.

A Flat random graph $G(V, L)$ is generated with $|V| = 12$ and $|L| = 18$. For such a small network, we have $G = G_P$, i.e., the application-level network is exactly the same as the physical network. Among those 12 nodes in G, we have three publishers ($|P| = 3$), three subscribers ($|S| = 3$) and six brokers ($|B| = 6$). Other parameters are set as $|E_p| = 3, |E_c| = 2, \alpha = 0.2, p_{pop} = 0.6, p_{unp} = 0.1$. We vary the probability p_c that a policy constrains event transmission along a link. For each value of p_c, 100 problem instances are generated. Both the non-enhanced heuristic and the enhanced heuristic using Φ_M are applied to each problem instance. Let c_{app} be the cost returned by the greedy heuristic and c_{opt} be the optimal cost[2]. We define the *Error Ratio* to be $r = \mathrm{E}[c_{app}]/\mathrm{E}[c_{opt}]$.

Figure 8 shows r as a function of p_c. From the figure, we observe that heuristic performs well – the average r returned by the non-enhanced heuristic is less than 1.08 in all cases, and the one returned by the enhanced heuristic is less than 1.012. Secondly, the enhanced heuristic indeed has a better performance compared to the non-enhanced one. Third, r decrease as p_c increases, i.e., our greedy heuristic performs better in a policy-constrained system. With a higher p_c, the number of links that can be used to transmit an event is reduced, which further reduces the set of feasible solutions. In this case, c_{app} is closer to c_{opt}. For example, when $p_c = 0.9$, we have $c_{app} = c_{opt}$. Note that when $p_c = 0$, there is no policy constraint on event distribution.

[2] The optimal cost is obtained by enumerating all feasible solutions, which has an exponential computational complexity.

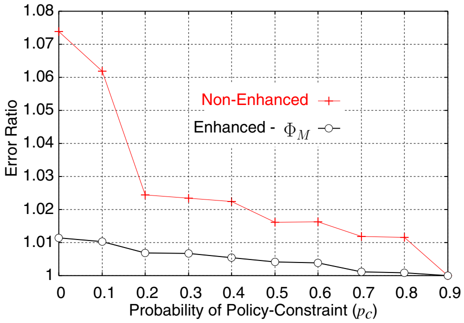


Figure 8. Comparison of approximate cost and optimal cost.

4.3.2 Investigating heuristic performance on larger systems

We next consider the effects of policy constraints and event composition on larger systems. All of the following study is conducted on two physical topologies – a Transit-Stub topology and a flat topology. Each topology contains 100 nodes. The Transit-Stub topology contains four transit nodes forming one transit domain, with the remaining stub nodes forming 12 stub domains. The flat topology is generated with parameters scale=10, edgemethod=3 and alpha=0.3 [20]. Due to the infeasibility of obtaining the optimal cost in larger systems, we only present the greedy cost in the following.

We first compare the performance of the enhanced heuristic using different orders of choosing subscribers. Recall that we consider three orders: the order of minimum cost (Φ_M), the order of longest distance (Φ_L), and the random order (Φ_R). The application network is formed by choosing $|P| = |S| = |E_p| = |E_c| = 10$ and $|B| = 20$ on each physical topology. No policy constraints are applied. 1000 instances of subscribers' interests, event composition and event rates are generated with $\alpha = 0.2, p_{pop} = 0.6, p_{unp} = 0.1$. For each problem instance, we apply the non-enhanced heuristic and the enhanced heuristic using Φ_M, Φ_L and Φ_R. Specifically, for random order Φ_R, we repeat the heuristic 100 times on a problem instance and obtain the average value (denoted as Φ_R-avg.) and the minimum value (denoted as Φ_R-min) for that particular problem instance. Eventually, the average value of the cost for the 1000 problem instances is calculated.

Figure 9 shows the simulation results. From these results we first notice that the greedy cost is greater in Transit-Stub (TS) topology than in Flat topology which reflects the fact that nodes have a higher connectivity in Flat topologies than in TS topologies. Secondly, the enhanced heuristic using Φ_L outperforms the others in the Flat topology whereas the one using Φ_M achieves the smallest cost in the TS topology. As discussed before, using Φ_L brings us the maximum increase of M_e, the set of source nodes of an event e. The average distance between any two nodes is smaller in the Flat topology than in the TS topology due to the higher node connectivity of the former. Thus when there are more source nodes in M_e, it is more likely that a subscriber will have a smaller distance to one of the source node in M_e, incurring a smaller receiving cost. As a result, Φ_L performs the best in the Flat topology. However, the node connectivity is lower in the TS topology and the heuristic using Φ_L achieves a cost a little greater than using Φ_M.

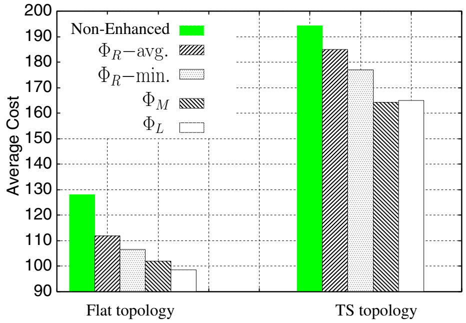


Figure 9. Comparison of heuristic cost.

In the rest of our simulation study, we will use Φ_L in the Flat topology and Φ_M in the TS topology, since they perform best in the corresponding scenario.

We next consider the effects of the number of composite events $|E_c|$. In this study, the application network is formed by choosing $|P| = |S| = 10$ and $|B| = 20$. No policy constraints are applied, i.e., $p_c = 1$. We vary the number of composite events $|E_c|$. For each value of $|E_c|$, 1000 instances of subscribers' interests, event composition and event rates are generated with $\alpha = 0.2, p_{pop} = 0.6, p_{unp} = 0.1$.

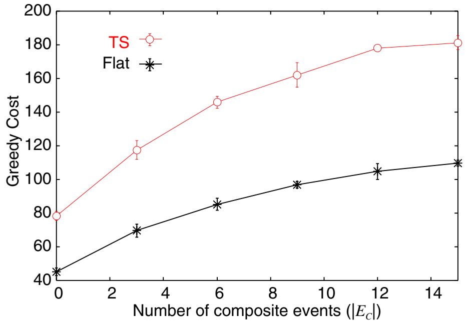


Figure 10. Approximation costs $|E_c|$.

Figure 10 shows the average cost returned by the heuristic as a function of $|E_c|$. As already shown in Figure 9, the greedy cost is greater in the TS network. An interesting finding is that the cost increases sublinearly as $|E_c|$ increases. As $|E_c|$ increases, the chance that a primitive event is used to generate composite events increases. The transmission of one primitive event can then be used to generate multiple composite events.

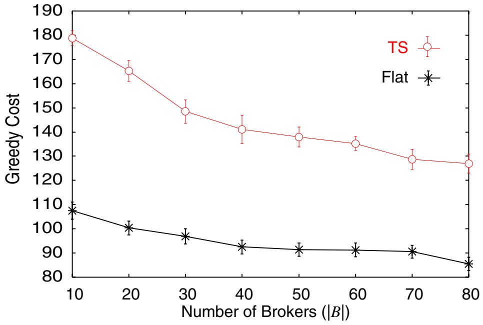


Figure 11. Greedy costs with $|B|$.

We further investigate the impact of the number of brokers $|B|$ on the greedy cost. Figure 11 plots the cost as a function of $|B|$ without policy constraints. This simulation is based on the application network formed on the two physical topologies as those used in Figure 10. We fix $|P| = |S| = |E_p| = |E_c| = 10$ and vary $|B|$. For each value of $|B|$, we construct 1000 instances of subscribers' interests, event composition and event rates using parameters $\alpha = 0.2, p_{pop} = 0.6, p_{unp} = 0.1$. This figure shows that the greedy cost monotonically decreases as $|B|$ increases in both topologies, since with more interior brokers available, events are able to take shorter paths. Note that when $|B| = 80$, the application network is exactly the same as the physical network. This result indicates that by adding more application-level brokers in a pub/sub system, we are able to achieve a smaller cost, especially in a TS topology.

5 Extension and Related Work

5.1 Implementation considerations

Our enhanced algorithm can be easily implemented on existing pub/sub architectures [5, 19, 11]. Using current pub/sub architectures, publishers are able to advertise events to publish; subscribers are able to specify their preference and consequently receive events of interest. Mechanisms proposed in [9, 17] can be used to detect composite events. Additional issues described below need to be considered for the greedy heuristic to be applied.

Topological orders of event distribution. Determining the topological ordering of the events is a sufficient condition for the greedy heuristic to find a feasible solution of a MICPICE problem if one exists. If the algorithm is centralized, it is easy to obtain the topological order when given a event composition matrix. Publishers and brokers consequently propagate cost information according to the order. In the absence of a centralized computation node, distributed algorithms (e.g. [16]) can be applied to obtain the order.

Dynamic subscription membership. Using Φ_M, when an event is delivered from its source nodes to a subset of subscribers, the greedy algorithm chooses a subscriber with the minimum receiving cost. Once a subscriber interested in event e leaves/joins the system, the current distribution solution of e may need to be modified, which may further change the solutions of other events. It may be expensive to update a distribution solution immediately when subscription membership changes. This problem also exists when the heuristic applies order Φ_L that chooses a subscriber having the longest distance to the set of source nodes. In this case, an alternative could be to update the event distribution solution periodically.

Distributed implementation. The main part of the greedy heuristic is the Dijkstra-like procedure that propagates and updates cost information. Since Dijkstra's algorithm can be implemented in a distributed way, the greedy heuristic can also be distributed.

5.2 Transmission costs with multicast-capable routers

In the simulation study above we assumed that physical routers are not multicast capable. Otherwise, the calculation of transmission costs will be different. Consider the example in Figure 7. If physical router X is multicast capable, the cost becomes $\lambda(e)[C(b_1, X) + C(X, b_2) + C(X, b_3)]$. It is clear that if multicast is supported by physical routers, the total transmission cost can be reduced. This subsection addresses the MICPICE problem when physical routers are multicast capable.

As before, let $G = (V, L)$ represent the application-layer pub/sub system, where $V = P \cup S \cup B$ is a set of application nodes and L is a set of logical links. The underlying application network G is a physical network represented by a directed weighted graph $G_P = (V_P, L_P, C_P)$, where $V_P \supseteq V$ (resp. L_P) is a set of physical nodes (resp. links) and each link $l \in L_P$ has a *physical link cost* $C_P(l)$. A routing table $\mathcal{T} : L \mapsto 2^{L_P}$ maps a logical link to a physical path, i.e., $\mathcal{T}(u, v)$ is the physical path from application node u to v. Based on this notation, a formal definition of the MICPICE problem in the presence of multicast-capable routers is presented in [6].

Definition 5.1: *Given $G = (V, L)$, $G_P = (V_P, L_P, C_P)$, $\mathcal{T}$, $\mathcal{S}$, $\mathcal{M}$, $\mathcal{I}$, λ and $\mathcal{P}$, the MICPICE problem is to find a transmission assignment $\eta : L \times E \mapsto \{0, 1\}$, where*

$$\eta((u,v),e) = \begin{cases} 1, & e \textit{ is delivered from } u \textit{ to } v \\ 0, & \textit{otherwise} \end{cases}$$

$$\textit{so as to minimize} \quad \sum_{e \in E} \lambda(e) \sum_{l \in \mathcal{K}_e} C_P(l) \qquad (9)$$
$$\text{where } \mathcal{K}_e = \bigcup_{(u,v):\eta((u,v),e)=1} \mathcal{T}((u,v))$$

subject to conditions (5)-(8) in Definition 2.1.

Informally, $\mathcal{K}_e$ is the set of physical links along which e is transmitted.

5.3 Measuring event-composition costs

Thus far, transmission costs are simply expressed as the product of event generating rates and link costs. This can be easily extended to include the event-composition costs, e.g., the overhead of composing composite events. This modification will be reflected in the greedy heuristic when predecessors are determined (particularly, line (5) in Figure 4)

5.4 Related work

There has been a large body of work in pub/sub systems. Recent research in pub/sub systems focuses on content-based systems, like SIENA [5] and Gryphon [19]. The advantage of content-based systems over channel-based alternatives is that subscribers have greater flexibility in specifying their requirements, instead of being limited to predefined channels.

In order to allow subscribers to receive events that satisfy complex patterns, composite events have been introduced in content-based pub/sub systems [7, 12, 17]. At the same time, various pub/sub architectures have also been proposed to detect composite events [9, 17]. We believe that these detection mechanisms can be directly applied in our work to optimize event distribution. However, composite events are not supported in Gryphon. Theoretically, SIENA supports composite events, but practically only the detection of event sequences has been implemented [9].

Bauer and Varma [3] proposed a distributed heuristic for multicast path setup based on shortest paths, referred to as *distributed SPH*. Our heuristic is quite similar to the distributed SPH. In particular, when delivering a primitive event using Φ_M, the enhanced heuristic performs exactly the same as the distributed SPH. When delivering a composite event e, our enhanced heuristic requires all of the source nodes of e be available, which is achieved by delivering events in a topological order.

6 Conclusions

Motivated by the practical concern for policy constraints on event distribution and the considerable demand and attention received for event composition, we have formulated and studied the MIn-Cost event distribution problem in the presence of PolIcy-constraints and Composite Events (MICPICE) in this paper. By reducing the Minimum Steiner Tree problem to MICPICE, we have proved the intractability of finding the optimal solution for MICPICE. Consequently, we proposed a greedy heuristic to approximately solve the problem. Our simulation study showed that the transmission cost returned by the greedy heuristic is within 1.08 of the optimal cost in the cases studied. An enhanced algorithm based on the greedy heuristic further improves the performance to fall within 1.012 of the optimal cost. Moreover, the approximation ratio becomes even smaller when the policy constraints are more restrictive. We also found that, by increasing the number of brokers, we are able to reduce the total transmission cost.

References

[1] M. Adler, Z. Ge, J. Kurose, D. Towsley, and S. Zabele. Channelization problem in large scale data dissemination. In *ICNP 2001*, pages 100–109, Riverside, CA, November 2001.

[2] F. Bauer and A. Varma. Degree-constrained multicasting in point-to-point networks. In *IEEE INFOCOM*, Boston, April 1995.

[3] F. Bauer and A. Varma. Distributed algorithms for multicast path setup in data networks. In *IEEE GLOBECOM*, Singapore, November 1995.

[4] D. Bell and L. LaPadula. Secure computer systems: Mathematical foundations and model. Technical Report M74-244, The MITRE Corporation, Bedford, MA, 1973.

[5] A. Carzaniga, D. S. Rosenblum, and A. L. Wolf. Design and evaluation of a wide-area event notification service. *ACM Transactions on Computer Systems*, 19(3):332–383, August 2001.

[6] W. Chen, Z. Ge, J. Kurose, and D. Towsley. Optimizing event distribution in publish/subscribe systems in the presence of policy-constraints and composite events. Technical Report 04-105, Dept. of Computer Science, University of Massachusetts, Amherst, 2004.

[7] C. Collet and T. Coupaye. Primitive and composite events in NAOS. In *Actes des 12e Journees Bases de Donneees Avancees*, pages 331–349, Cassis, France, September, 1996.

[8] T. Cormen, C. Leiserson, and R. Rivest. *Introduction to Algorithms*, chapter 26, pages 550–578. MIT Press, 1990.

[9] S. Courtenage. Specifying and detecting composite events in content-based publish/subscribe systems. In *Proceedings of 22nd International Conference on Distributed Computing Systems Workshops (ICDCSW '02)*, Vienna, Austria, July 02 - 05, 2002.

[10] J. Fortes. Transnational digital government research: Building regional partnerships. Highlighted case study presentation at the Digital Government conference, 2003.

[11] Z. Ge, P. Ji, J. Kurose, and D. Towsley. Matchmaker: Signaling for dynamic publish/subscribe applications. In *Proceedings of IEEE ICNP 2003*, Atlanta, GA, November 4-7, 2003.

[12] N. H. Gehani, H. V. Jagadish, and O. Shmueli. Composite event specification in active databases: Model implementation. In *Proceedings of the 18th International Conference on Very Large Databases*, Vancouver, British Columbia, Canada, 1992.

[13] C. Gropl, S. Hougardy, T. Nierhoff, and H. J. Promel. Approximation algorithms for the steiner tree problem in graphs. In *Steiner trees in industry, X. Cheng and D.Z. Du (eds.)*, pages 235–279, Kluwer 2001.

[14] S. Kent and R. Atkinson. Security architecture for the internet protocol. IETF, RFC 2401, `http://www.faqs.org/rfcs/rfc2401.html`.

[15] J. F. Kurose and K. W. Ross. *Computer Networking: A Top-Down Approach Featuring the Internet*, chapter 4, pages 271–376. Addison Wesley, 2000.

[16] J. Ma, K. Iwama, T. Takaoka, and Q. Gus. Efficient parallel and distributed topological sort algorithms. In *2nd AIZU International Symposium on Parallel Algorithms/Architecture Synthesis (pAs'97)*, Fukushima, Japan, March 1997.

[17] P. R. Pietzuch, B. Shand, and J. Bacon. A framework for event composition in distributed systems. In *Proceedings of the 4th International Conference on Middleware (MW'03)*, Rio de Janeiro, Brazil, June, 2003.

[18] A. Rowstron, A. Kermarrec, M. Castro, and P. Druschel. SCRIBE: The design of a large-scale event notification infrastructure. In *Third Networked Group Communication*, London, UK, November 2001.

[19] R. Strom, G. Banavar, T. Chandra, M. Kaplan, K. Miller, B. Mukherjee, D. Sturman, and M. Ward. Gryphon: An information flow based approach to message brokering. In *International Symposium on Software Reliability Engineering '98 Fast Abstract*, 1998.

[20] G. Tech. Modeling topology of large internetworks. `http://www.cc.gatech.edu/projects/gtitm/`.

[21] P. Winter. Steiner problem in networks: a survey. *ACM Networks*, 17(2):129–167, Summer 1987.

[22] S. Zabele, M.Dorsch, Z. Ge, P. Ji, M. Keaton, J. Kurose, J. Shapiro, and D. Towsley. SANDS: Specialized active networking for distributed simulation. In *DARPA Active Networks Conference and Exposition (DANCE)*, San Francisco, CA, May 2002.