Matchmaker: Signaling for Dynamic Publish/Subscribe Applications *

Zihui Ge, Ping Ji, Jim Kurose and Don Towsley
Computer Science Department,
University of Massachusetts at Amherst,
{gezihui,jiping,kurose,towsley}@cs.umass.edu

Abstract

The publish/subscribe (pub/sub) paradigm provides content-oriented data dissemination in which communication channels are established between content publishers and content subscribers based on a matching of subscribers interest in the published content provided – a process we refer to as "matchmaking". Once an interest match has been made, content forwarding state can be installed at intermediate nodes (e.g., active routers, application-level relay nodes) on the path between a content provider and an interested subscriber. In dynamic pub/sub applications, where published content and subscriber interest change frequently, the signaling overhead needed to perform matchmaking can be a significant overhead. We first formalize the matchmaking process as an optimization problem, with the goal of minimizing the amount of matchmaking signaling messages. We consider this problem for both shared and per-source multicast data (content) distribution topologies. We characterize the fundamental complexity of the problem, and then describe several efficient solution approaches. The insights gained through our analysis are then embodied in a novel Active Matchmaker Signaling Protocol (AMSP). AMSP dynamically adapts to applications' changing publication and subscription requests through a link-marking approach. We simulate AMSP and two existing broadcast-based approaches for conducting matchmaking, and find that AMSP significantly reduces signaling overhead.

1 Introduction

The publish/subscribe (*pub/sub*) paradigm provides content-oriented data dissemination in which communication channels are established between content publishers and content subscribers based on a matching of interests in the content provided by publishers and that requested by subscribers. We consider a distributed architecture in which participating publishers and subscribers send/receive information through a multicast data dissemination structure (such as native-IP-multicast or application-level-multicast), with "interior" network elements (such as active routers or application-level relay nodes) forwarding data only when one or more downstream subscriber(s) are interested in receiving that data. Since unwanted information is dropped at the earliest opportunity, such an architecture has proven to be useful and efficient for building distributed simulations [22] and large scale event notification systems [4, 5, 20].

In order for the interior network nodes to determine whether to drop or to forward data to a downstream node, content forwarding state (e.g. data filters [22, 16]) should be configured along the data forwarding paths between publishers and subscribers. This is achieved through a **signaling process** in which subscribers send signaling messages (*subscription requests*) specifying the attributes of the content they wish to receive, and, optionally, publishers generate signaling messages (*publication advertisement*) specifying the content attributes of the data they will be publishing. We refer to the process of performing matching among publication advertisements and subscription requests as "*matchmaking*" and the interior network nodes which receive, process and forward publication advertisements and subscription requests as "*matchmakers*".

Two approaches have been explored to date for signaling publication advertisements and subscription requests, and performing interest matching. In the first approach (which we will refer to as "broadcast subscription") [20, 4, 5, 15], subscription requests from content subscribers are broadcast through the distribution paths to all publishers. Published data are then forwarded towards interested subscriber(s) only when they match a previously received subscription request from a downstream subscriber. A disadvantage of such an approach is that all subscription requests are installed as content forwarding state, while ideally only the subscription requests that can be satisfied by one or more upstream publisher(s) should be used. As an enhancement, in the second approach (which we will refer to as "broadcast publication"), explored in [5, 14, 15], publication advertisements from content publishers are broadcast to all subscribers, and stored as matchmaking state in the matchmakers. Subscrip-

*This work is supported in part by DARPA subcontracts with Northrop Grumman Information Technology 2000-012 and 2000-109, National Science Foundation subcontract with the University of Florida UF-EIES-0205003-UMA, and National Science Foundation grant ANI-0085848.

Proceedings of the 11th IEEE International Conference on Network Protocols (ICNP'03)
1092-1648/03 $17.00 © 2003 **IEEE**

IEEE COMPUTER SOCIETY

tion requests for content are then forwarded upstream by the matchmakers only towards those publishers that provide content that matches downstream subscriber interest.

In dynamic pub/sub applications, such as distributed interactive simulation (DIS) [12, 18, 22], published content and subscriber interest can change frequently, i.e., subscription requests and publication advertisements are updated frequently. In such cases, the signaling overhead needed to perform matchmaking (with broadcast publication/subscription involved) can be a significant overhead, even in comparison to data flows (between interest-matched publishers and subscribers only). This creates the need for a more efficient approach toward matchmaking.

In this paper, we investigate a third approach towards matchmaking in dynamic pub/sub applications. Rather than broadcast all subscriptions or all publications, we consider an approach in which publications and subscriptions are *selectively* exchanged among matchmakers. We will see that a matchmaker's decision as to whether to forward a subscription request can be answered easily (and often optimally) by exploiting a simple structural property of the pub/sub system. Informally, this property allows these publication advertisements and subscription requests to meet in the "middle" of the network, with matching being performed at the matchmaker nodes where they meet. Additional signaling messages carrying the information about matched publications and subscriptions are then forwarded upstream towards the publisher and downstream towards the subscriber, setting up data forwarding state on the appropriate paths. We note that while all three approaches result in the same flow of published *data* among the nodes, they differ significantly in their *signaling* overhead.

Our contributions are both theoretical and practical in nature. We formalize an optimization problem – the Min-Cost Matchmaker problem, whose objective is to minimize the overall signaling required to conduct matchmaking. We consider two different scenarios, one in which the underlying multicast dissemination topology is a single shared-tree and the other in which data are disseminated over per-source trees. We study the complexity of the Min-Cost Matchmaker problem and present off-line algorithms that solve the optimization problem in polynomial time. We observe that the Min-Cost Matchmaker solution incurs significantly less signaling cost than the broadcast publication or broadcast subscription approaches.

In order to put the min-cost matchmaker architecture into practice, we then consider how the off-line algorithms can be translated into practical protocols. We find that in the case of a shared tree, the solution structure immediately suggests a simple, optimally-efficient, distributed signaling protocol. In the case of per-source trees, we propose an online approximation algorithm that (unlike the off-line optimization algorithm) requires neither global knowledge of the multicast topology nor information about every publishers' and subscribers' publishing and subscribing activity. We develop a simple and fully distributed active matchmaker signaling protocol (AMSP) that realizes these algorithms. AMSP is able to *dynamically* adapt to the publication and subscription activities of the end systems, reducing the signaling overhead needed for matchmaking. We simulate AMSP and the two broadcast-based approaches for providing a pub/sub service in a distributed interactive simulation [12, 18, 22]. We find that AMSP can correctly make matches and set up data filters while incurring 75% less signaling overhead under various network and end-system-application scenarios.

The remainder of this paper is organized as follows. In Section 2, we provide an overview of the pub/sub architecture. In Section 3, we describe and formalize the Min-Cost Matchmaker problem in multicast-based pub/sub systems. We present polynomial time algorithms for the Min-Cost Matchmaker problem for both shared and per-source based multicast topologies and evaluate the benefits of adopting a min-cost matchmaker configuration in Section 4. In Section 5, we present our design of AMSP. We compare the performance of AMSP and existing broadcast signaling approaches via simulation in Section 6. Finally, in Section 7, we conclude our work and describe related future research.

2 A Pub/Sub Network Architecture

Figure 1 illustrates the network setting that we consider. End systems (gray squares in Figure 1) run applications that publish and/or request content. Within the network, ordinary routers (white circles) and active interior nodes (gray circles) sit along the multicast data dissemination paths between publishers and subscribers. Multicast distribution may be accomplished via IP multicast or overlay multicast [13, 19, 2]. Similarly, active interior nodes may be active routers in an active multicast network [22] or relay nodes in an application overlay network [20]. In either case we refer to these interior nodes as active routers, reflecting their functionality.

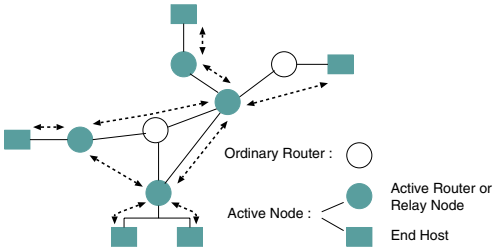

Figure 1. A pub/sub network infrastructure

Let us now consider the application-level data flow in more detail. Applications, sitting on the end hosts, generate data (in the case of publishers) and receive data (in the case of subscribers) over multicast data distribution trees, whose

Proceedings of the 11th IEEE International Conference on Network Protocols (ICNP'03)
1092-1648/03 $17.00 © 2003 **IEEE**
IEEE COMPUTER SOCIETY

links are indicated by solid lines in Figure 1. Although all applications participate in the same multicast group, not all subscribers are interested in all published data. Therefore, a pub/sub network uses a content-oriented data dissemination mechanism, such as content-based routing [20] or data filtering at the active nodes [16, 22], to ensure that data is delivered only to interested subscribers. Several studies [17, 5] have been devoted to designing efficient data structures and fast algorithms for data filtering and/or forwarding. In order to perform this filtering/forwarding, an active node will need information regarding publication advertisements and subscription requests; *it is the role of the signaling protocol to transport, store, and process these advertisements and requests*.

The signaling overlay component of the pub/sub infrastructure is indicated by dashed lines in Figure 1 and is logically separate from the data forwarding components. Indeed, we assume that the signaling connections between upstream/downstream active matchmakers in the data distribution path are unicast, reliable, and bi-directional, and may be routed differently from the multicast data packets. A protocol such as the Active Topology Discovery Protocol (ATDP) [21] can be used to construct this signaling overlay.

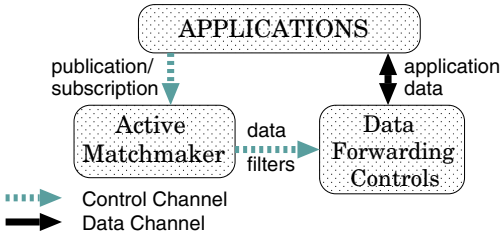

Figure 2. The relationship among functional components of a pub/sub system

Figure 2 summarizes the relationship between the data-publishing/subscribing applications, the data-filtering and forwarding controls, and the active matchmakers:

- The **application** generates publication advertisements and/or subscription requests and passes them to the active matchmaker components at the end system through the signaling interface. The application also sends and receives data (content) through the multicast data channel.
- The **active matchmaker** system is responsible for signaling publication advertisements and subscription requests, and detecting matches between them. Once a match is detected, a matchmaker propagates the publication/subscription matching information to all active nodes along the multicast data path between the matched publisher(s) and subscriber(s).
- Finally, the **data forwarding control** uses the publication/subscription matching information provided by the active matchmaker system to configure data filters and forwarding control to ensure that only data of interest are forwarded to downstream subscribers. The underlying IP- or application-level multicast infrastructure is used to disseminate application data. The underlying multicast topology can be either a single shared tree (e.g., CBT [1]) or per-source (publisher) trees (e.g., PIM-SM version 2 [8]).

2.1 Active matchmaker system with link-marking

Previous pub/sub systems [5, 15, 14] have assumed that either publications or subscriptions are broadcast everywhere. We will see shortly, however, that a matchmaker system that selectively distributes both publications and subscriptions can significantly reduce the overall number of signaling messages propagated throughout the system. To accomplish this, our active matchmaker system marks each directed link in the signaling overlay as either a "publish link" (p-link) or a "subscribe link" (s-link). By properly marking the links and allowing publications to be propagated only through p-links and subscriptions to be propagated only through s-links, publications and subscriptions can meet at a matchmaker node in the "middle" of the network, where matchmaking can be performed. Additional signaling messages carrying information about matched publications and subscriptions can then be forwarded upstream towards the publisher(s) and downstream towards the subscriber(s), setting up data forwarding controls on the path for the forthcoming data flows.

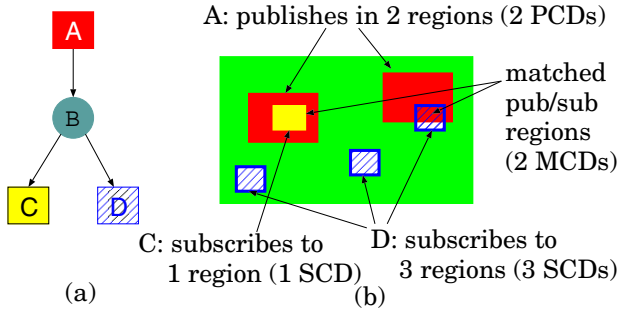

Figure 3. An example of content descriptors in a two-dimensional virtual world

Different link-marking configurations will result in different signaling costs. To illustrate these differences, let us consider a distributed interactive simulation (DIS) system simulating a 2-dimensional virtual world as a pub/sub application. A DIS entity will often only be interested in the events that occur in the area within a certain distance from its virtual location. In this case, the entity's subscription request would consist of a geometric description of the

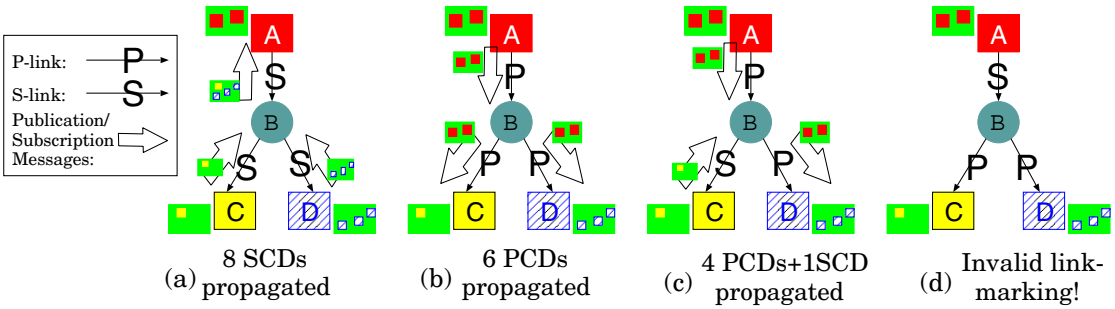

Figure 4. Examples of link-marking schemes

area in which it is interested. Similarly an entity will often only publish data regarding events or observations that occur within some distance of its virtual location. Here, the entity's publication advertisement would be the geometric description of the area for which it is publishing content. We note that location-sensitive applications in a mobile physical (as opposed to virtual) world have many of these same characteristics. Figure 3 illustrates a scenario in which one entity (A) is connected to entities C and D through active router B. A is publishing data associated with two different regions (shown by dark-shaded rectangles in Figure 3(b)); entities C and D wish to receive data published in the regions of interest shown. As a result of matchmaking, C will receive data from the intersection of its subscription request and A's left publication region, while D will receive data only from the intersection of its rightmost subscription request and A's right publication region.

Figure 3 is a specific example (a two-dimensional virtual world) of the more general notion of content matching. We will refer to the information that describes the published/subscribed content as a **content descriptor (CD)**. We will refer to the content descriptor contained in a publication message as a **PCD**, the content descriptor contained in a subscription messages as an **SCD**, and the content descriptor describing the overlapping region of a PCD and an SCD (that is used to set up the data forwarding controls) as a matched CD (**MCD**).

The number of CDs being propagated and stored throughout the system serves as a useful measure of signaling overhead. Note that in a dynamic pub/sub system, publishers and subscribers may change their interests over time (e.g., as a DIS simulation entity moves around in the virtual DIS world). In such cases, we are interested not only in the number of CDs that a publisher publishes or that a subscriber subscribes to, but also the *rate* at which they update their publications/subscriptions. *This signaling rate then determines the signaling cost.* We should mention that PCDs or SCDs from the same publisher or subscriber can be updated independently and asynchronously. Thus, merging multiple CD updates in one signaling message (to reduce signaling overhead) is not always feasible.

For ease of explanation, we will first assume that the message signaling rate is proportional to the number of distinct CDs. That is, we assume all CDs are updated at the same frequency; in Section 5, we will relax this assumption.

3 Min-Cost Matchmaker Problem

Having defined the notion of p-links and s-links and having defined signaling cost, one can naturally pose an optimization problem aimed at identifying the link-marking scheme that results in the least signaling overhead. We illustrate this optimization problem though an example in Figure 4.

Figure 4 compares several different link-marking schemes for the scenario described in Figure 3. The small rectangles within the larger rectangles on the side of nodes A, C and D represent the CDs that A publishes and C and D subscribe to. In Figure 4(a), all links are marked as s-links. This corresponds to the broadcast subscription scheme in which all subscribed CDs from subscribers C and D are propagated upstream toward the publisher. We observe that the broadcast subscription scheme requires 8 SCDs to be propagated across a link (the one CD from C is propagated across links CB and BA, while the three CDs from D are propagated across links DB and BA). When a broadcast publication scheme is used, as in Figure 4(b), the publications generated by A are sent to all subscribers, with 6 PCDs being transmitted overall. Figure 4(c) shows another link-marking scheme, where links AB and BD are marked as p-links and BC is marked as an s-link. The total signaling overhead in this case is reduced to 5 CDs (4 PCDS and 1 SCD). In fact, this link-marking configuration requires the least number of CDs to be signaled over all possible link-markings, making it the *optimal* link-marking that we are looking for. This example demonstrates the potential benefits of adopting a link-marking scheme (and the consequent PCD and SCD forwarding) other than the broadcast publication or broadcast subscription approach.

We note that not all link-markings are valid. We require that each publication be able to meet each subscription at some point within the signaling overlay, so that content matching can be performed and the MCD (if it exists) can be signaled along the path between the matched publisher and subscriber. Informally (we will formalize this require-

ment shortly), this means that any path from a publisher to a subscriber be a series of zero or more p-links followed by zero or more s-links. In the broadcast subscription approach, the publishing end hosts are the nodes at which the matches are made (A in Figure 4(a)); in the broadcast publication approach, the subscribing end hosts are the the nodes at which the matches are made (C and D in Figure 4(b)). For other link-markings, matches can be made at both end hosts and active routers. In the example of Figure 4(c), node B matches the PCDs from A and the SCD from C, and node D matches the PCDs from A and the SCDs from itself. Using any of these approaches, 2 MCDs will be detected (from Figure 3(b)), and a total of 4 signaling messages containing the MCDs will be propagated from the point at which a match is detected, so that the data forwarding controls can be properly configured (not shown in the figures). Figure 4(d) shows an invalid link-marking example.

3.1 Model and Notation

Let us now formalize the Min-Cost Matchmaker problem. We first consider the case in which multicast data dissemination uses source-specific multicast trees. As we shall see in Section 4.1, the shared multicast tree structure can be described as a more constrained version of this problem formalization.

A publish/subscribe system consists of:

- A network $G = (V, E)$ where V is the set of end hosts and active routers and E is the set of directed links.
- A set, $P \subseteq V$, of publishing nodes that provide information.
- A function, $f : P \rightarrow \mathbb{R}^+$, that maps each publishing node to the number of PCDs that it publishes, i.e., $\forall x \in P, f(x) =$ the number of PCDs that x publishes.
- A set, $S \subseteq V$, of subscriber nodes that request information.
- A function, $g : S \rightarrow \mathbb{R}^+$, that maps each subscriber node to the number of SCDs in which it is interested.
- A set $\mathcal{T}$, of multicast trees, where $T_x \in \mathcal{T}$ is the set of directed edges comprising the multicast tree for publisher x, i.e., $T_x \subseteq E : \forall e \in T_x$, e is an edge within the source-based multicast tree rooted at x that reaches all subscriber nodes, where $x \in P$.
- Partial order relations that represent the multicast tree for each publisher. $\Re_x \subseteq T_x \times T_x : \forall (t, q) \in \Re_x$, t is an upstream link of q in the multicast tree of x, where $x \in P$. Thus $t \Re_x q$ means that t is upstream of q in the multicast tree for publisher x. Note that $\Re_x$ is a transitive relation, i.e., $\forall r, q, t \in T_x, (r \Re_x q \wedge q \Re_x t) \rightarrow r \Re_x t$.

Our goal is to find a link marking function that maps the directed links within any multicast tree to s-links, and p-links, which will be denoted by 0 and 1, respectively:

$$\mathcal{M} : T \longrightarrow \{0, 1\}$$

where $T = \bigcup_{x \in P} T_x$ represents the set of directed links that lie within at least one multicast tree for some source(s). For $t \in T$, $\mathcal{M}(t) = 0$ means that t is an s-link and that all subscription messages should be propagated through t, and $\mathcal{M}(t) = 1$ means that t is a p-link and that all publish messages should be propagated through t. This link marking function must satisfy the following **validity constraint:** along the multicast path from a given publisher to a given subscriber, publication messages should meet subscription messages at exactly one node, i.e., p-links connecting from the publisher should join s-links connecting to the subscriber at a switchover point. Notationally, for publisher x, this is expressed as:

$$\mathcal{M}(t) = 0 \Rightarrow ((\forall q \in E) t \Re_x q \Rightarrow \mathcal{M}(q) = 0) \qquad (1)$$

or equivalently,

$$\mathcal{M}(t) = 1 \Rightarrow ((\forall q \in E) q \Re_x t \Rightarrow \mathcal{M}(q) = 1)$$

We define the cost of a link-marking configuration as the number of PCDs and SCDs propagated through all the directed links within the multicast trees, i.e.,

$$Cost(\mathcal{M}) = \sum_{t \in T} (\sigma^t(\mathcal{M}) + \tau^t(\mathcal{M})) \qquad (2)$$

where for link $t = (m, n) \in T$, σ^t is the number of PCDs propagated through link t, and τ^t is the number of SCDs propagated through link t. A more precise definition is presented in [9].

Given the above definitions, the **Min-Cost Matchmaker** problem is concerned with finding a link-marking function $\mathcal{M}$ that **minimizes cost function (2)** while **satisfying validity constraint (1)**.

Although the Min-Cost Matchmaker problem has a very large solution space, we will see in the next section, there exist polynomial time algorithms that can solve this problem.

4 Algorithms for Min-Cost Matchmaker

Before presenting the algorithms to solve the Min-Cost Matchmaker problem, we first identify several important properties of the problem.

Property 1: *The solution to the Min-Cost Matchmaker problem has a cost that is lower than, or equal to, that for broadcast publication or broadcast subscription.*

We define $\mathcal{M}^P(\cdot)$: $\forall t \in T, \mathcal{M}^P(t) = 1$;
$\mathcal{M}^S(\cdot)$: $\forall t \in T, \mathcal{M}^S(t) = 0$.

Here, $\mathcal{M}^P$ is the link marking that marks all the links on multicast trees as p-links, and $\mathcal{M}^S$ is the link marking that

marks all the links as s-links. They correspond to broadcast publication and broadcast subscription, respectively. Property 1 holds since both $\mathcal{M}^P$ and $\mathcal{M}^S$ are valid candidate solutions to the Min-Cost Matchmaker problem.

Property 2: *For any valid link marking scheme, the cost associated with a link is either the cost of that link when all links are marked as p-links or the cost of that link when all links are marked as s-links.*

Property 2 is more precisely captured in the following Lemma.

Lemma 1: *If $\mathcal{M}$ is valid,* $\forall t \in T$,

$$\sigma^t(\mathcal{M}) + \tau^t(\mathcal{M}) = \begin{cases} \sigma^t(\mathcal{M}^P) & \text{if} \mathcal{M}(t) = 1 \\ \tau^t(\mathcal{M}^S) & \text{if} \mathcal{M}(t) = 0 \end{cases}$$

The proof of Lemma 1 is presented in [9].

We will find Property 2 very useful. Even though it does not reduce the overall size of the solution space, it makes evaluating the cost associated with each valid solution straightforward. Also, this property provides us with insight about finding efficient algorithms for Min-Cost Matchmaker problem as we will see in Sections 4.1 and 4.2.

4.1 Min-Cost Matchmaking in a shared multicast tree

In this subsection, we focus on a special case of the Min-Cost Matchmaker problem where the underlying data dissemination structure is a shared multicast tree. Core-based Tree (CBT) [1] and PIM-Sparse Mode [6] are two examples of shared-tree multicast routing protocols.

When a shared tree is used, for any given links t, q, if t is an upstream link of q on some multicast tree, then for any other multicast tree, t is either not within the tree or it is an upstream link of q, i.e.,

$$(\exists x \in P, t \Re_x q) \Rightarrow (\forall y \in P, t \notin T_y \vee t \Re_y q) \qquad (3)$$

The shared tree structure simplifies the Min-Cost Match -maker problem. We show that the greedy approach given in Figure 5 can find the solution efficiently. For each link t, this algorithm first computes the number of PCDs in the case that all links are p-links ($\sigma^t(\mathcal{M}^P)$) and the number of SCDs in the case that all links are s-links ($\tau^t(\mathcal{M}^S)$). After that, each link, $t \in T$, is marked as a p-link if $\sigma^t(\mathcal{M}^P) < \tau^t(\mathcal{M}^S)$; otherwise t is marked as an s-link.

Claim 1: The greedy approach shown in Figure 5 can find the solution of the Min-Cost Matchmaker problem for shared multicast tree.

The success of the greedy approach is due to a very nice monotonicity property of the shared multicast tree structure. We state this **Monotonicity property** below.

Property 3: For all $t, q \in T$, if $(\exists x, t \Re_x q)$ then $\sigma^t(\mathcal{M}^P) \leq \sigma^q(\mathcal{M}^P)$ and $\tau^t(\mathcal{M}^S) \geq \tau^q(\mathcal{M}^S)$.

The proof of Property 3 is presented in [9].

```
1.  for each t ∈ T:
2.        compute σ^t(M^P), τ^t(M^S)
3.  for each t ∈ T:
4.        if σ^t(M^P) < τ^t(M^S)
5.              M(t) ← 1;
6.        else M(t) ← 0;
7.  return M;
```

Figure 5. A greedy approach for a shared multicast tree

The correctness of Claim 1 can be established by verifying that the solution produced by the algorithm in Figure 5 is both valid and optimal (see [9] for a detailed proof).

4.2 Min-Cost Matchmaking in per-source trees

The greedy approach presented above gives the optimal link-marking solution for shared multicast trees. However, it does not guarantee a valid link-marking for per-source multicast distribution trees. This is because the Monotonicity property, which we were able to establish for shared multicast trees, may be violated when source-specific routes between two active nodes diverge. An example illustrating this phenomenon is presented in [9]. As the result, the Min-Cost Matchmaker problem with per-source multicast trees is much more complicated than that for a shared multicast tree. Fortunately, we still can develop a polynomial time algorithm that solves it. We describe here the intuition behind this algorithm; details can be found in [9].

Consider the matchmaker scenario in which all links are marked as p-links. We try to improve upon this marking scheme by changing some p-links to s-links. From Property 2, we know that remarking a p-link, t, to an s-link will reduce the total cost by η^t, where $\eta^t = \sigma^t(\mathcal{M}^P) - \tau^t(\mathcal{M}^S)$ (where a negative value of η^t indicates a cost increase). Ideally, we would like to change all links with a positive value of η^t to s-links. However, due to the validity constraint, if a link is changed to an s-link, all of its downstream links are constrained to be s-links as well. Therefore, what we are really interested in is to find a "profitable" subset (C) of links with a positive value of η value whose overall *gain* ($\sum_{t \in C} \eta^t$) is greater than the sum of the *loss* of their downstream links ($\sum_{t \in C_{\Re}} -\eta^t$, where $C_{\Re} = \{q | \ \ (\exists t \in C)(\exists x \in P) \ \ t \Re_x q\}$). To find a profitable subset, we can construct a bipartite graph where each node corresponds to a link in the matchmaker problem. We place nodes with positive η^t values on one side and nodes with negative η^t values on the other side. A directed edge pointing from a positive node to a negative node is added if the positive node is an upstream link of the negative node in the matchmaker problem. We then send flows from the positive side to the negative side through the bipartite graph. Finding a maximum flow will help us find the profitable subset C, in that the maximum flow through C

Proceedings of the 11th IEEE International Conference on Network Protocols (ICNP'03)
1092-1648/03 $17.00 © 2003 **IEEE**
IEEE COMPUTER SOCIETY

will be limited by its downstream nodes' total loss. Identification of saturated downstream negative nodes will provide us with the information we need to find the subset C. Due to space limitations, we do not present the detailed algorithm or its proof of correctness here; interested readers can find the algorithm in [9], where an example is also provided to help illustrate the algorithm.

As we have seen in the sketch above, the Min-Cost Matchmaker problem for per-source multicast trees can be reduced to a Max-flow problem, to which polynomial-time solutions have been proposed [11, 7]. Our experience with the simulated scenarios discussed later in the paper indicates that a generated Max-flow problem may contain many isolated subgraphs, which makes solving the max-flow problem not very computationally expensive.

4.3 Top-down marking (TDM) algorithm

The polynomial-time algorithm discussed in the previous section for finding the optimal link-marking configuration with per-source based multicast distribution trees requires global information about the underlying multicast topology and the costs of all PCDs and SCDs from all publishers and subscribers. Unlike the case of shared trees, however, it does not suggest a distributed implementation that might be used in practice. In practice, we seek an algorithm (like that for shared trees) in which each link can locally determine its marking based on the costs of CDs propagated from its upstream and downstream neighbors.

```
1.  for each t ∈ T:
2.        compute σ^t(M^P), τ^t(M^S)
3.  for each t ∈ T sorted according to ⋃_{x∈P} R_x: [1]
4.        if ∃q ∈ T, x ∈ P, qR_x t ∧ M(q) = 0
5.              M(t) ← 0;
6.        else if σ^t(M^P) < τ^t(M^S)
7.                    M(t) ← 1;
8.              else M(t) ← 0;
9.  return M;
```

Figure 6. A greedy approximation for per-source multicast distribution trees

Figure 6 presents a greedy approximation to the link-marking problem in per-source-based trees. It takes a *top-down* approach, by sequentially checking each link along paths from publishers to subscribers. Under the condition that the validity constraints must not be violated, (i.e., no downstream link of an s-link can be marked as a p-link), the algorithm marks the link greedily as an s-link or a p-link depending on the local cost. Of course, since we do not search the entire solution space, this top-down marking approach will not always yield optimal solutions. We will see shortly, however, that the obtained solutions compare quite well with the optimum solution over a range of parameters.

We note that the top-down marking scheme always finds the optimal link-marking configuration in the case of a shared multicast tree, since the top-down algorithm in Figure 6 is effectively identical to the greedy shared-tree algorithm in Figure 5. Therefore, a pub/sub system can always adopt this top-down link-marking algorithm, without regard to the form of the underlying multicast tree.

4.4 Evaluation

In this subsection, we use simulation to generate instances of the Matchmaker problem, and explore the relative signaling costs of the Min-Cost Matchmaker solution (both exact and approximate), and the subscription broadcast and publication broadcast approaches described earlier.

4.4.1 Problem settings

We assume a hierarchical network topology represented by a transit-stub structure, and use the Georgia Tech topology generator [10] to generate such network topologies. Each network contains 100 nodes, 4 of which are transit nodes, forming one transit domain, with the remaining stub nodes forming 12 stub domains. We assume that all transit nodes and stub nodes are active. For each problem, we randomly choose publishers and subscribers from the stub nodes. We do not exclude the possibility that a publisher and a subscriber are collocated on the same node. When shared multicast trees are considered, we choose one of the transit nodes as the "core" and use CBT as the underlying routing protocol. When per-source based trees are considered, we let the data dissemination tree of each publisher be the reverse shortest path tree from each of the subscribers to the publisher (as built by DVMRP).

For each publisher or subscriber, we randomly assign a number of CDs that it publishes, or to which it subscribes. The number of CDs is drawn from a *bounded Pareto* distribution given by

$$f(n) = \frac{\alpha k^{\alpha} n^{-\alpha-1}}{1-(k/p)^{\alpha}} \qquad k \le n \le p$$

with $k = 1$, $p = 100$ and $\alpha = 1.0$.

For each link-marking configuration, we determine the total number of CDs propagated throughout the system, as defined in the $Cost$ function in equation 2. We consider four different approaches: (i) broadcast subscription: all links are marked as s-links (All-S) (ii) broadcast publication: all links are marked as p-links (All-P) (iii) optimal configuration: here, the link-marking is obtained by solving the Min-Cost Matchmaker problem (MCM) using our algorithms from Section 4.1 and Section 4.2, for shared tree and per-source based trees respectively (iv) suboptimal configuration: here, the link-marking is obtained by the top-down marking approximation algorithm (TDM) in Section 4.3 for per-source multicast trees.

[1] Use $\sigma(\mathcal{M}^P) - \tau(\mathcal{M}^S)$ to break tie. See [9] for details.

IEEE COMPUTER SOCIETY

4.4.2 Evaluation results

We first fix the number of publishers and the number of subscribers at 20 ($|P| = |S| = 20$) and generate 1000 different problem samples. Figure 7 plots the cumulative distributions of the $Cost$ corresponding to All-P, All-S and MCM respectively for shared multicast trees. The median value of $Cost$ under MCM is approximately 500, whereas the medians for All-P and All-S are approximately 3000 each. Furthermore, both All-P and All-S have 90-th percentiles of the $Cost$ greater than 5500, whereas, the 90-th percentile of the $Cost$ for MCM is only around 800.

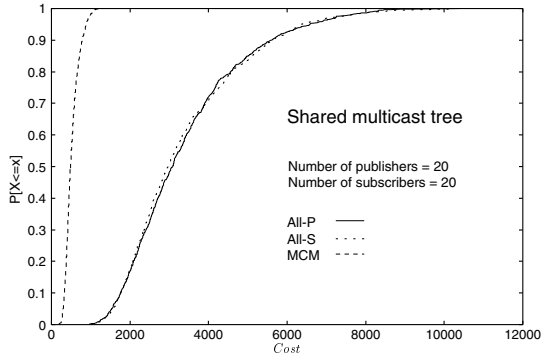

Figure 7. Cumulative distribution function of $Cost$ for shared multicast trees

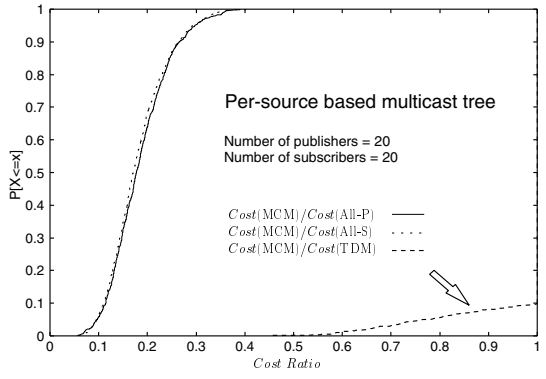

Figure 8. The cumulative distribution of $Cost$ ratios for per-source trees

Figure 8 plots the cumulative distributions of the $Cost$ ratio of MCM over that of All-P, All-S and TDM when per-source multicast trees are used. We find that in more than 50% of the samples, MCM provides solutions with $Cost$ less than 20% of that of either All-P or All-S, and in more than 95% of the samples, MCM can reduce the $Cost$ by at least two thirds in comparison to All-P or All-S. We also find that TDM finds solutions very close to optimal – in 90% of the problem samples, TDM finds the optimal solutions ($Cost$ ratio=1), while for the remaining 10% cases, TDM rarely gives a link-marking configuration that has more than twice the cost of the optimal MCM solution.

In summary, we find that MCM can significantly outperform the two broadcast approaches in terms of the $Cost$ associated with the resulting link-marking configuration. Furthermore, we have observed that the top-down marking algorithm finds very good approximate solutions, ones that are usually close or identical to the optimal solutions found via MCM.

5 AMSP Design

In the previous section, we saw that TDM link-marking often significantly reduces the signaling messages needed for matchmaking publications and subscriptions over using either broadcast publication or broadcast subscription. In this section, we will use the algorithms from section 4 as a guide in designing a fully distributed active matchmaker signaling protocol (AMSP) that dynamically adjusts the link-marking configuration to minimize the signaling cost for the active matchmaker system. Recall that an active matchmaker system should distribute publication advertisements and subscription requests, conduct interest matches among them, and trigger the installation of data filtering/forwarding controls once a match is detected.

While our emphasis in Section 4 was primarily theoretical, we now consider the three following practical aspects of AMSP:

- Determining the link marking configuration in AMSP in a **distributed** fashion, where each active node only determines the link markings for its attached links.
- Dealing with **dynamic** systems in which publishers and subscribers update their PCDs and SCDs whose number and update frequency may change over time. As we will see, AMSP estimates the rate of PCDs and SCDs transmitted and adjust the link-markings adaptively in order to accommodate such dynamics.
- AMSP **integrates** the link-marking scheme into the active matchmaker system. Our evaluation considers not just PCD and SCD propagation signaling costs, but all signaling costs, including signaling needed to obtain the optimal marking and signaling needed for the installation of data filtering/forwarding.

5.1 AMSP architecture

Our active matchmaker signal protocol (AMSP) consists of three components that run in parallel:

1. AMSP estimates CD update frequencies based on the publication advertisements and subscription requests signaled from applications and disseminates the estimated values. Using this information, AMSP com-

Proceedings of the 11th IEEE International Conference on Network Protocols (ICNP'03)
1092-1648/03 $17.00 © 2003 **IEEE**
IEEE COMPUTER SOCIETY

putes the estimated publish/subscribe costs for each local link.

2. Based on the estimated publication/subscription costs, AMSP applies the TDM link-marking algorithm in a distributed manner to assign a *p-link* or *s-link* attribute to each local link.
3. AMSP propagates publications and subscriptions according to the link attribute. The AMSP module at an active node where publication and subscription messages meet performs matching and signals the matched content descriptors (MCDs) to the nodes sitting on the path between the matched sender(s) and receiver(s). This permits content filtering/forwarding controls to be set up at these nodes.

In the following three subsections, we examine each of these AMSP components in detail.

5.2 Estimating and disseminating CD update frequencies

Let f_x denote the PCD update frequency (the number of PCDs updated per unit time) from publisher x, and let g_x denote the SCD update frequency from subscriber x. Since each publisher/subscriber may be publishing or subscribing to multiple CDs and with different CD update frequencies, f_x and g_x are determined by the overall CD update rates of all objects at x.

To adapt to an application's changing publishing/subscribing activity, f_x and g_x are periodically measured by the AMSP modules at each publisher and subscriber. To avoid synchronization between different publishers and subscribers, a small random value can be added to the measurement intervals. AMSP uses exponential smoothing to compute the average frequencies as follows,

$$f_x(t) = (1-\alpha) f_x(t-1) + \alpha \frac{\text{No. of PCD updates}}{\text{interval length}}$$

$$g_x(t) = (1-\alpha) g_x(t-1) + \alpha \frac{\text{No. of SCD updates}}{\text{interval length}}$$

where, α is the smoothing parameter.

If a new estimate differs significantly from the previously announced value, AMSP broadcasts the new value through the signaling overlay (towards all publishers for g_x and towards all subscribers for f_x). An AMSP signaling message of type COST_ANNOUNCE, that includes the originator's id x and the new value of f_x or g_x, is provided for this purpose.

To compute the publication/subscription costs at each link, a table of CD update frequencies, f_x and g_x is maintained at each active node by its AMSP module. When a COST_ANNOUNCE message is received, the local AMSP module updates its corresponding table entry with the new value of f_x or g_x contained in the message, and computes the **publication cost**, $C_i^{(p)}$, and the **subscription cost**, $C_i^{(s)}$, for each related local link i. Here

$$C_i^{(p)} = \sum_{x \in P_i} f_x \ , \quad C_i^{(s)} = \sum_{x \in S_i} g_x$$

where P_i is the set of publishers that disseminate data through link i, and S_i is the set of subscribers that receive data sent through link i. Therefore $C_i^{(p)}$ and $C_i^{(s)}$ are the estimates of the update frequencies of CDs that either are, or could be, propagated over link i.

A link's publication cost $C_i^{(p)}$ and subscription cost $C_i^{(s)}$ are needed by the link-marking algorithm as we will describe next.

5.3 Performing link-marking

Recall that in the signaling overlay, each link connecting two active nodes is bi-directional. We define the node sitting at the upstream end of the data flow as the *upstream node* of that link, and the node sitting at the downstream end of the data flow as the *downstream node*. The upstream-downstream relationship of the two nodes can be reversed for different multicast sources. To ensure that the link-marking scheme works, the link-marking attribute should be maintained identically at both ends of the link and initialized consistently (e.g., both as p-link).

We adapt the top-down link marking algorithm in Section 4.3 that determines the proper link-marking into a distributed computation in which the marking attribute for each link is independently determined only by a link's upstream AMSP module. For each node, we define a link i as an *upstream link* of link j if data received from i is disseminated over j within some multicast tree(s). For instance, in the example shown in Figure 3(a), link AB is an upstream link of link BC and link BD.

```
    for an outgoing link i:
1.    if (any upstream link of link i is s-link)
2.        label link i as an s-link;
3.    else if((C_i^(s) − C_i^(p)) > δ)
4.            label link i as p-link;
5.    else if ((C_i^(p) − C_i^(s)) > δ)
6.            label link i as s-link;
7.    else do not change previous marking;
```

Figure 9. Distributed top-down marking alg.

The distributed version of the TDM algorithm is presented in Figure 9. Here, link i is forced to be an s-link if any of its upstream links is marked as an s-link (this guarantees convergence of system to a valid link-marking configuration). Otherwise, the link-marking decision is made by comparing the publication cost and the subscription cost at this link. A hysteresis threshold, δ, is used to avoid frequent changes of the link-marking attribute. The link-marking al-

Proceedings of the 11th IEEE International Conference on Network Protocols (ICNP'03)
1092-1648/03 $17.00 © 2003 **IEEE**
IEEE COMPUTER SOCIETY

gorithm is executed every time the publication cost $C_i^{(p)}$ or the subscription cost $C_i^{(s)}$ associated with link i changes.

When a link-marking is changed by the upstream node at a link, CHANGE_MARKING and CHANGE_MARKING_REPLY signaling messages are exchanged between the two ends of the link to instantiate the change. Both the new link marking attribute and a list of associated CDs are included in these messages.

5.4 Propagating CDs

AMSP disseminates content descriptors based on the link-marking of each link. This is done through another type of signaling message, CD_UPDATE. A CD_UPDATE message that carries a PCD is forwarded only through p-links; one that carries an SCD is forwarded only through s-links.

In AMSP, the PCDs/SCDs are maintained in the AMSP module at the downstream/upstream node of a link respectively. Furthermore, when the upstream node of a link changes the marking attribute from s-link to p-link, the associated PCDs are sent to the downstream node (in the CHANGE_MARKING message). Similarly, when the downstream node of a link receives a CHANGE_MARKING message indicating that the link marking attribute has been changed to s-link, the associated SCDs are sent to the upstream node (in the CHANGE_MARKING_REPLY message).

Matches between PCDs and SCDs are detected by the AMSP modules at the "matchmaking active nodes" where p-links are connected to s-links. If a match is detected, the AMSP module triggers the installation of the match content descriptor (MCD) in the local data-forwarding plane. It also sends CD_UPDATE messages carrying MCD information to its upstream and downstream active nodes along the multicast data dissemination path between the source publisher and the requesting subscriber. These messages, in turn, trigger the MCD installation on their data forwarding plane.

The removal of a PCD/SCD/MCD can be achieved in two different ways. In a soft-state approach, publishers and subscribers periodically refresh their publication advertisements and subscription requests. A CD is removed if a refresh message is not received before a timeout. In a hard-state approach, publishers and subscribers assign a unique ID to each PCD and SCD, and are responsible for explicit or implicit withdrawal (by sending an update using the same ID) of a previously transmitted CD. The withdrawal of a PCD or a SCD may invalidate one or more MCD(s). The "matchmaking nodes" are responsible for removing those MCDs. The choice of which of these two approaches performs best remains an interesting research problem.

6 AMSP Simulation

To simulate a pub/sub system, we again generate network topologies, choose publishers and subscribers, and create shared tree or per-source trees multicast routing in the same manner as described in Section 4.4.

We are particularly interested here in scenarios in which the PCD and SCD change dynamically over time. To capture the dynamics of applications' publication and subscription activity, we simulate a distributed simulation (game) scenario on a two-dimensional gridded virtual world, as illustrated in Figure 10.

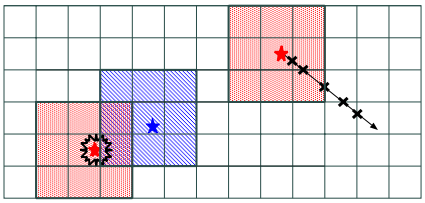

Figure 10. Application model

Each application (i.e., publisher and/or subscriber) runs multiple simulation entities (e.g., planes, tanks, and soldiers in a DIS scenario) in the distributed game. Each simulation entity generates a distinct PCD or SCD that indicates the area in which it will either be publishing data or in which it has a subscription interest. In our simulation, the content descriptors correspond to the 3×3 squares centered at the originator's location, as illustrated in Figure 10. As entities move, they generate new PCD or SCD updates. This happens when a simulation entity crosses a grid, as indicated by an '$\times$' in Figure 10. We use the random waypoint model [3] to drive the simulated motion of the simulation entities. We further introduce heterogeneity into the simulation by creating three types of entities: high-speed entities (e.g., an aircraft), medium-speed entities (e.g., a ground vehicle) and low-speed entities (e.g., a person on-foot).

The simulation entities are assigned to the publishers and subscribers as follows. For each publisher or subscriber x, an initial population of simulation entities, N_x, is assigned, where N_x is uniformly distributed between 50 and 150. After the simulation starts, new entities are added to each application according to a Poisson process with arrival rate $\lambda = 1/60$, (i.e., on average one new simulation entity is added to each application per minute). The lifetime of each entity is exponentially distributed with mean $60N_x$, (i.e., on average one simulation entity leaves the simulation per minute in each application).

In our simulated pub/sub infrastructure, active routers use the MCDs (the overlapping areas between PCDs and SCDs), determined by the distributed active matchmakers, to set up data forwarding controls so that the subscribers receive only content that matches their interests. We implement three signaling approaches for building the active matchmaker system – AMSP, the broadcast publication and the broadcast subscription approaches. In the next subsection, we present our simulation results.

Proceedings of the 11th IEEE International Conference on Network Protocols (ICNP'03)
1092-1648/03 $17.00 © 2003 **IEEE**
IEEE COMPUTER SOCIETY

6.1 Simulation results

The performance metric of interest is the total signaling message rate (messages per second). This corresponds to the number of per-hop signaling messages sent during the simulation divided by the length of the simulated time. These signal messages include the CD_UPDATE in both AMSP and the broadcast schemes, as well as the COST_ANNOUNCE, CHANGE_MARKING_REPLY and CHANGE_MARKING that are only used in AMSP.

We simulate the three signaling approaches discussed in this paper – broadcast publication (BP), broadcast subscription (BS), and AMSP. Note that although BP results in an All-P link marking and AMSP utilizes a TDM approach, the evaluation here differs from that in Section 4.4 in that the simulation scenario is dynamic and all signaling messages are included as cost. To provide a fair comparison, for each experiment, we use the same network and application settings, and the same initial value for the pseudo-random seed for each of these three signaling approaches.

In Figure 11, we plot the smoothed signaling message rate (with a 30-second window size) over time for BP, BS and AMSP, when there are 20 publishers and 20 subscribers, whose entities move around within a 1000×1000 virtual world. Per-source multicast trees are used. We observe a significant reduction in the total number of control messages propagated and processed using AMSP, as compared to BP and BS. With the exception of some fluctuations at the beginning of the simulation, both BP and BS require more than 3000 control messages (network wide) per second, while AMSP exhibits a control message rate of less than 800 per second, 3% of which are used to perform link-marking, with the remainder being used to disseminate PCDs, SCDs, and MCDs.

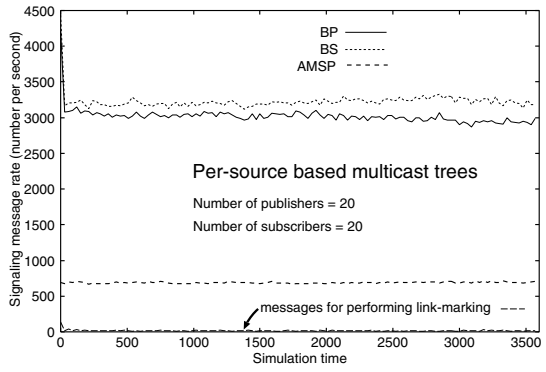

Figure 11. Running control message rate for different signaling approaches

We also investigate scenarios with an asymmetric number of publishers and subscribers. Figure 12 plots the average message rate for simulations in which there are 20 publishers and from 5 to 50 subscribers; here, per-source based multicast trees are used. Each point in the graph corresponds to the average of 10 independent experiments and we also plot the 90th percentile confidence intervals. We observe that when the number of publishers increases, the average signaling message rate also increases for all three approaches. When the number of publishers exceeds the number of subscribers, BS outperforms BP, otherwise BP outperforms BS. This behavior is likely due to the fact that publishers and subscribers have the same mobility model. Note, however, that the slope of increase associated with BS is lower than that with BP, suggesting that BP scales better with the number of subscribers. As expected, AMSP has signaling rate significantly below that of either BP or BS. We also conducted simulations with a fixed number of subscribers and varied the number of publishers. The results (not shown) were substantially the same as Figure 12, with the curves for the two broadcast approaches switched. In all cases, AMSP has both the lowest message signaling rate, and the smallest increase as the number of publishers/subscribers increases, demonstrating its ability to handle both a large number of publishers and a large number of subscribers.

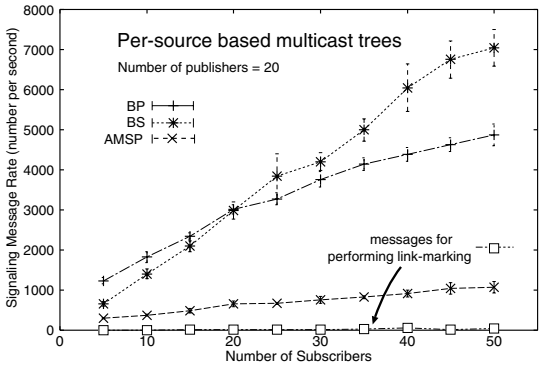

Figure 12. Signaling rate v.s. number of subscribers

In the simulations described thus far, we assume that the application divides the virtual space into 1000 by 1000 grids. Now we investigate the impact of grid size granularity. Since content descriptors are expressed in units of grid size, the grid size granularity plays a key role in determining the pub/sub system's capability of scoping content dissemination. In the extreme case that the whole virtual space is one grid unit, every publishers' content will match every subscribers' interest. Thus data will be flooded everywhere. Generally, the finer the gird, the more accurately subscribers can describe their interests, and the better the filtering granularity of the matchmaker system. However, countering this trend is the fact that, as the grid size becomes small, publishers and subscribers will frequently update their CDs as a

Proceedings of the 11th IEEE International Conference on Network Protocols (ICNP'03)
1092-1648/03 $17.00 © 2003 **IEEE**
IEEE COMPUTER SOCIETY

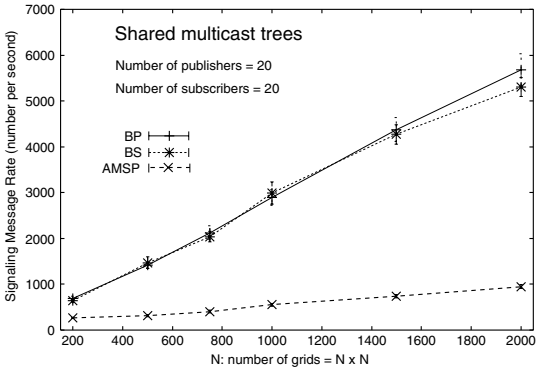

Figure 13. Signaling rate v.s. map granularity

result of their motion, and thus incur an increasing signaling rate. This phenomenon is shown in Figure 13. In Figure 13, we plot the signaling message rates when the virtual space is divided into different grid sizes, ranging from 200×200 to 2000×2000. While all approaches incur higher signaling cost as the number of grids increases, AMSP requires significantly less signaling overhead. In other words, for a given system signaling capacity, AMSP can support a much finer grid size than the broadcast approaches, and therefore can provide better content-oriented data forwarding.

7 Conclusion and Future Work

In this paper, we have presented a distributed pub/sub architecture in which end systems and "interior" network elements known as *matchmakers* sit between content publishers and subscribers. Matchmakers receive, process and forward publication advertisements and subscription requests, conduct interest matching among them, and use the matched publication and subscription to establish content-based data filtering/forwarding controls. We formalized an optimization problem, namely the Min-Cost Matchmaker problem, whose objective was to find the matchmaking configuration that minimized the signaling overhead. We studied the complexity and developed polynomial time algorithms for the Min-Cost matchmaker problem. Using these algorithms as a starting point, we then developed a simple and fully distributed active matchmaker signaling protocol (AMSP) that dynamically adapts to applications' publication and subscription activity and adjusts link-marking configurations so as to minimize the overall signaling overhead. We simulated AMSP as well as two existing approaches, broadcast publication and broadcast subscription, and found that AMSP significantly reduces the signaling message overhead in comparison to these two other approaches under various network and application configurations.

In our future work, we would like to implement and integrate the AMSP into an existing active-network-facilitated large-scale distributed simulation system [22].

References

[1] T. Ballardie, P. Francis, and J. Crowcroft. Core based trees (CBT), an architecture for scalable inter-domain multicast routing. In *ACM SIGCOMM*, 1993.

[2] S. Banerjee, B. Bhattacharjee, and C. Kommareddy. Scalable application layer multicast. In *Proceedings of ACM SIGCOMM*, Pittsburgh, PA, August 2002.

[3] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Mobile Computing and Networking*, 1998.

[4] L. F. Cabrera, M. B. Jones, and M. Theimer. Herald: Achieving a global event notification service. In *Proceedings of the Eighth Workshop on Hot Topics in Operating Systems (HotOS-VIII)*, Elmau, Germany, May 2001.

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

[6] S. Deering, D. Estrin, D.Farinacci, V.Jacobson, C.Liu, and L.Wei. The pim architecture for wide-area multicast routing. *ACM Transactions on Networks*, April 1996.

[7] J. Edmonds and R. M. Karp. Theoretical improvements in the algorithmic efficiency for network flow problems. *Journal of the ACM*, 19:248–264, 1972.

[8] B. Fenner, M. Handley, H. Holbrook, and I. Kouvelas. PIM-SM: Protocol specification (revised), March 2002. INTERNET DRAFT.

[9] Z. Ge, P. Ji, J. Kurose, and D. Towsley. Matchmaker: A distributed publish/subscribe architecture. Tech. Report, Umass Amherst, 2002.

[10] Georgia Tech. *Modeling Topology of Large Internetworks*. http://www.cc.gatech.edu/fac/Ellen.Zegura/graphs.html.

[11] A. V. Goldberg and R. E. Tarjan. A new approach to the maximum flow problem. In *Proceedings of Eighteenth Annual ACM Symposium on Theory of Computing*, pages 136–146, 1986.

[12] D. V. Hook, S. Rak, and J. Calvin. Approaches to relevance filtering. In *11th DIS Workshop*, September 1994.

[13] Y. hua Chu, S. G. Rao, and H. Zhang. A case for end system multicast. In *Proceedings of ACM SIGMETRICS*, Santa Clara,CA, June 2000.

[14] G. Muhl. Generic constraints for content-based publish/subscribe. In *Proceedings of the 6th International Conference on Cooperative Information Systems (COOPIS)*, 2001.

[15] G. Muhl, L. Fiege, F. C. Gartner, and A. Buchmann. Evaluating advanced routing algorithms for content-based publish/subscribe systems. In *MASCOTS*, 2002.

[16] M. Oliveira, J. Crowcroft, and C. Diot. Router level filtering for receiver interest delivery. In *Second International Workshop on Networked Group Communication (NGC)*, Palo Alto, California, Nov. 2000.

[17] J. Pereira, F. Fabret, F. Llirbat, and D. Shasha. Efficient matching for web-based publish/subscribe systems. In *Conference on Cooperative Information Systems*, pages 162–173, 2000.

[18] S. Rak and D. V. Hook. Evaluation of grid-based relevance filtering for multicast group assignment. In *14th DIS Workshop*, March 1996.

[19] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker. Application-level multicast using content-addressable networks. In *Proceedings of Networked Group Communication*, London, UK, November 2001.

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

[21] J. K. Shapiro, J. Kurose, D. Towsley, and S. Zabele. Topology discovery service for router-assisted multicast transport. In *Openarch*, 2002.

[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)*, May 2002.

Proceedings of the 11th IEEE International Conference on Network Protocols (ICNP'03)
1092-1648/03 $17.00 © 2003 **IEEE**

IEEE COMPUTER SOCIETY