CCD: Efficient Customized Content Dissemination in Distributed Publish/Subscribe

Hojjat Jafarpour, Bijit Hore, Sharad Mehrotra and Nalini Venkatasubramanian

Dept. of Computer Science, Univ. of California at Irvine
{hjafarpo,bhore,sharad,nalini}@ics.uci.edu

Abstract. In this paper, we propose a new content-based publish/subscribe (pub/sub) framework that enables a pub/sub system to accommodate richer content formats including multimedia publications with image and video content. The pub/sub system besides being responsible for matching and routing the published content, is also responsible for converting the content into the suitable (target) format for each subscriber. Content conversion is achieved through a set of content adaptation operators (e.g., image transcoder, document translator, etc.) at different nodes in the overlay network. We study algorithms for placement of such operators in the pub/sub broker overlay in order to minimize the communication and computation resource consumption. Our experimental results show that careful placement of these operators in pub/sub overlay network can lead to significant cost reduction.

Keywords: Publish/Subscribe, Operator placement, Content dissemination

1 Introduction

Publish/Subscribe (pub/sub) systems provide a selective dissemination scheme that delivers published content only to the receivers that have specified interest in it [1, 3, 5]. To provide scalability, pub/sub systems are implemented as a set of broker servers forming an overlay network. Clients connect to one of these brokers and publish or subscribe through that broker. When a broker receives a subscription from one of its clients, it acts on behalf of the client and forwards the subscription to others in the overlay network. Similarly, when a broker receives content from one of its clients, it forwards the content through the overlay network to the brokers that have clients with matching subscriptions. These brokers then deliver the content to the interested clients connected to them.

In this paper, we consider the problem of customized delivery in which clients, in addition to specifying their interest also specify the format in which they wish the data to be delivered. The broker network, in addition to matching and disseminating the data to clients also customizes the data to the formats requested by the clients. As the published content becomes richer in format, considering content customization within the pub/sub system can significantly reduce resource consumption. Such content customizations have become more attractive due to recent technological advances that has led to significant diversification of how users access information. Emerging mobile and personal devices, for instance, introduce specific requirements on the format in which content is delivered to the user. Consider a distributed video dissemination application over Twitter [2] where users can publish video content that must be delivered to their followers (subscribers). Followers may subscribe to such a channel using a variety of devices and prefer the content to be customized according to their needs. Additionally, device characteristics such as screen resolution, available network bandwidth etc., may also form the basis for required customization. Another example of such customized content dissemination system is dissemination of GIS maps annotated with situational information in responding to natural or man made disasters. In this case, receivers may require content to be customized according to their location or language.

Simply extending the existing pub/sub architectures by forcing the subscribers or publishers to customize content may result in significant inefficiencies and suboptimal use of available resources in the system. Therefore, there is a need for novel approaches for customized dissemination of content through efficient use of available resources in a distributed networked system. The key issue in customized content dissemination using distributed pub/sub framework is where in the broker network should the customization be performed for each published content? An immediate thought is to perform requested customizations at the sender broker prior to delivery. Such approach could result in significant network cost. Consider a simple broker network in Figure 1 where node *A* publishes a high resolution video in 'mpeg4' format and nodes *G, H* and *I* have subscribers that requested this content in 'avi', 'flv' and '3gp' formats, respectively. By performing customizations in the sender broker, *A*, the same content is transmitted in three different formats through <*A, B*> and <*B, D*> links which results in increased network cost. The alternate might be to defer customizations to the receiver brokers or broker *D*. Consider another case where *J, K* and *L* have subscribers with hand held devices that requested the video in '3gp' format. If the customizations are deferred to receiver brokers, conversion from 'mpeg4' to '3gp' is done three times, once in each receiving broker which results in higher consumption of computation resource in brokers. This also increases the communication cost by transmitting larger size video in 'mpeg4' format while it could be transmitted in '3gp' format that has smaller size.

The resulting communication and computation costs can be reduced by intelligently embedding customization operators in the pub/sub overlay network. For instance, the increased network cost in the first scenario could be prevented if the published video is sent to broker *D* in the original format and the customization operators are performed in this broker. Also by performing the conversion once at broker *A* or *C*, computation cost can be reduced significantly in the second scenario.

The above example shows merit of placement of operators in the network. In this paper, we explore this problem systematically and develop algorithms for efficient placement of operators. We model published content and required customization operators as a graph structure called *Content Adaptation Graph (CAG).* Then, we propose an optimal operator placement algorithm for small CAGs. The proposed algorithm performs the required operators in broker overlay such that the resulting communication and computation cost is minimized. For the larger CAGs, we show that the problem is NP-hard and propose a greedy heuristics-based iterative algorithm that significantly reduces customized dissemination cost compared to the cases where customizations are done either in the sender broker or in the subscriber brokers. Our extensive experiments show that the proposed algorithms considerably reduce bandwidth consumption and total customization cost in variety of scenarios.

The overall contributions of this paper are:

- We formally define the customized content dissemination, CCD, problem in a distributed pub/sub systems (Section 2). We also show that CCD with minimum cost is NP-hard when the number of requested formats is large.
- For small number of requested formats where enumeration of format sets is feasible, we propose an optimal operator placement algorithm in pub/sub broker network that minimizes the customization and dissemination cost (Section 4).
- For large number of requested formats we propose a greedy heuristics-based algorithm (Section 5).
- We present results of our extensive evaluation of the proposed techniques that show the considerable benefit of using them (Section 6).

We finally present related work in Section 7 followed by conclusions in Section 8.

2 Customized Content Dissemination

DHT-based Pub/Sub systems: Our CCD system architecture is based on a DHT-based pub/sub system [10, 11]. It consists of a set of content brokers that are connected through a structured overlay network. Each client connects to one of the brokers and communicates with the system through this broker. Often in DHT-based pub/sub, content space is partitioned among the brokers. Each broker maintains subscriptions for its partition of content space and is responsible for matching them with publications falling in its partition. In fact, each broker is the *Rendezvous Point (RP)* for the publication and subscriptions corresponding to its partition. When a broker receives a subscription from its client, it first finds the broker(s) responsible for partition(s) that the subscription falls in and forwards it to them. Similarly, when a broker receives a published content from its client, it finds the corresponding RP broker and forwards the content to the RP. The content is matched with the subscriptions at the RP and the list of brokers with matched subscriptions is created. Then the RP disseminates the content to all of these brokers through a dissemination tree constructed using the DHT-based routing scheme in the broker overlay network. Finally, every broker (with at least one client having a matching subscription) receive the content and transfer it to the respective clients. Since a broker acts as a proxy for all clients that connect to it, we can assume that it is the the subscriber or publisher and therefore simply concentrate on the broker overlay network. Various DHT-based routing techniques have been proposed in the literature [7, 8] that can be used for routing content from RP to the matching brokers. In this paper we use the *Tapestry* routing scheme [7], however, we can easily generalize our approach to other DHT-based routing schemes. In this paper we assume that given a set of subscribers (receivers), a broker can construct the dissemination tree as in Tapestry which then remains fixed for this particular instance of the dissemination event. For more details on dissemination tree construction we refer the interested reader to [9]. We choose the DHT-based pub/sub on Tapestry for a variety of reasons, two important ones being (i) In DHT-based pub/sub systems, for a given publication, a single broker (RP) has complete information about all brokers with matching subscriptions as well as formats in which content is to be delivered to them. (ii) Tapestry enables brokers to estimate the dissemination path for content, which is used to estimate the dissemination tree. Note that the estimated dissemination tree may not be same as the actual dissemination tree. An alternative for using the estimated dissemination tree is to discover the actual dissemination tree using a *tree discovery message* that is initiated at the RP and sent to all subscribing brokers. The leaf brokers in the dissemination tree then resend the message to the RP. Each message keeps information about the route from the RP to the leaf brokers which is then used by the RP to construct the exact dissemination tree for the given publication. In this paper we use tree discovery messages for constructing dissemination trees for publications. Figure 1 depicts a sample dissemination tree.

2.1 Content Adaptation Graph

We assume every client has a profile describing receiving-device characteristic (e.g., screen size and resolution) and connection characteristics (e.g., connection type and bandwidth). The client profile is registered at its broker and is used to determine the format(s) in which content needs to be delivered. Each subscription of the client along with its profile is forwarded to the corresponding RP which uses this information for optimal routing computation.

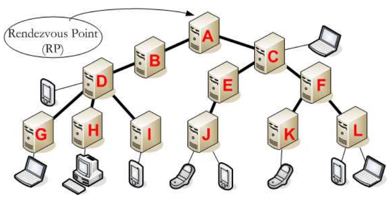

Fig. 1. Sample dissemination tree.

Similar to the conventional DHT-based pub/sub systems, the published content is forwarded to the corresponding RP. However, after detecting the brokers with matching subscriptions, the published content must be customized and disseminated according to the profiles of the matched subscriptions. For simplicity, let us assume that the computational resources at the brokers and transmission links between them (represented by edges in the dissemination tree) are identical, i.e., their characteristics such as bandwidth, delay, CPU speed etc. are same in every part of the tree[1]. Now, if the set of required formats is $F = \{F_0, ..., F_{m-1}\}$, for content $\mathbb{C}$ and format $F_i \in F$, we can associate a transmission-cost $\mathcal{T}_{F_i}(\mathbb{C})$ for each link. Let $O_{(i,j)}$ denote the operator that converts content format from F_i to F_j and its associated conversion cost by $\mathcal{C}_{O_{(i,j)}}(\mathbb{C})$. This represents the computation cost of performing this operator at any broker[2]. Note that it may not always be feasible to convert content from any given format F_i into another format F_j. For example, it might not be possible to convert a low resolution image into a higher resolution one. Alternatively, the system might not support particular conversions even if it were possible, e.g.,

[1] We have also considered the general case where brokers and links are not identical, however, due to space constraint we do not present it in this paper.

[2] In general we will assume these costs to represent the per-unit costs

converting video in 'avi' format into 'flv'. In such cases we assume $O_{(i,j)}$ to be undefined.

We use a directed, weighted graph structure to represent the required formats for a published content and the relationship between these formats. We will call this the *Content Adaptation Graph (CAG)*. The vertices in CAG correspond to the various content formats and the directed edges between two vertices represents the operator that converts content from the source format to the sink format directly. The associated *conversion cost* is represented by the weight of the edge. Similarly, a weight associated with each node of the CAG represents the per-unit *transmission cost* in that format. Figure 2 illustrates a CAG involving four formats of an 'mpeg4' video content with different frame sizes and bitrates. In this CAG we represent the transmission cost in Megabytes (MB) and the conversion cost in seconds.

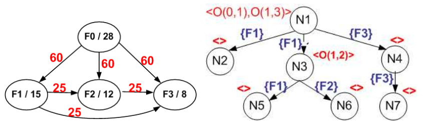

Fig. 2. A sample CAG and dissemination plan.

2.2 Cost-based Customized Dissemination

Consider the problem of customized dissemination of content $\mathbb{C}$ in format F_0 from RP to a set of brokers $R = \{R_1, .., R_r\}$ ($R \subseteq N$). Let F^{R_j} be the set of formats required at broker R_j. Let $\mathbb{T}$ denote the dissemination tree constructed according to the Tapestry framework where $N = \{N_1, .., N_n\}$ be the set of nodes and E be the set of edges in this tree. We denote the rendezvous node RP by N_1.

For a given dissemination tree $\mathbb{T}$, a *customized content dissemination plan* or *CCD plan* is an annotated tree $\mathbb{P}$ (with the same set of nodes and edges as $\mathbb{T}$) where each node and edge is annotated by the customization operators performed at the node and the formats in which the content is transmitted along the link respectively. Figure 2 shows a sample plan where the published content is delivered in format F_1 to brokers N_2 and N_5, in format F_2 to broker N_6 and in format F_3 to broker N_7. A subtree in the customization plan is called a *subplan*.

A customization plan provides the following information for each node, N_i, and link $< N_i, N_j >$ in the dissemination tree.

- O_{N_i}: the operators that are performed at N_i. E.g., in the plan depicted in Figure 2, $O_{N_1} = \{O_{(0,1)}, O_{(1,3)}\}$ that convert format F_0 to format F_1 and format F_1 to format F_3, respectively.
- $F_{in}^{N_i}$: the set of content formats that are received at N_i (from its parent). E.g., $F_{in}^{N_2} = \{F_1\}$ in Figure 2.
- $F_{out}^{N_i}$: the set of content formats that are required in N_i or are being sent by N_i to its children. E.g., $F_{out}^{N_3} = \{F_1, F_2\}$ in Figure 2.
- $F_{<N_i,N_j>}$: the set of formats that content is transmitted over $< N_i, N_j >$. E.g., $F_{<N_1,N_2>} = \{F_1\}$.

In every customization plan the content to be disseminated is available at the root node (RP) of the dissemination tree in its original published format. In a *valid* plan at every node N_i the input format set is identical to the set of formats that N_i receives from its parent. The input format for each operation $O_{(m,n)}$ performed at node is either forwarded by its parent or is generated at the node as a result of other operations. Likewise, the formats in which content is forwarded by N_i to its children are either received from its parent or generated in situ as a result of an executed operation. Finally, in a valid plan for every link $< N_i, N_j >$ the formats transmitted over it needs to pre-exist at its source i.e., $F_{<N_i,N_j>} \subseteq F_{out}^{N_i}$.

Cost Model: The conversion cost of a plan is the sum of costs of carrying out the operators specified for each of its nodes and transmission cost is the sum of costs of transmitting the content in the specified formats over all the links in the dissemination tree. Our model is similar to the one used in [18, 20] for in-network stream processing and cache replacement. We denote the conversion cost of a plan $\mathbb{P}$ by $\varphi_{\mathbb{P}}$ and the transmission cost by $\tau_{\mathbb{P}}$.

The *total cost* of the plan $\mathbb{P}$ for content $\mathbb{C}$ is denoted by $\Theta_{\mathbb{P}}(c)$, as a function of its conversion and transmission costs. In general one can use an additive formula such as:

$$\Theta_{\mathbb{P}}(\mathbb{C}) = \alpha\tau_{\mathbb{P}} + \beta\varphi_{\mathbb{P}} \text{ , where } \varphi_{\mathbb{P}} \text{ and } \tau_{\mathbb{P}} \text{ are normalized values, } \alpha, \beta \geq 0$$

The parameters α and β in the above cost function provide flexibility to customize the total cost function based on the system characteristics. For instance, if processing resources in a system are limited and expensive, the total cost function can reflect this by giving more weight to computing cost. Based on the above discussion, the computation cost of the plan depicted in Figure 2 is 110 and the communication cost of this plan is 73. Assuming $\alpha, \beta = 1$, the total cost of this plan will be 183. Therefore, the optimization problem can be stated as follows:

Customized Content Dissemination (CCD) Problem: Given a dissemination task find a valid customization plan with minimum total cost.

Theorem 1: CCD problem is NP-hard.

Proof: We show that the CCD problem is NP-hard when there is only one broker in the system. Clearly, if the problem is NP-hard for one broker, it remains NP-hard for $n(> 1)$ brokers too. We show that the NP-hard problem of computing the *"Minimum directed Steiner Tree"* can be reduced to an instance of the CCD problem. The minimum directed Steiner tree problem is the following: Given a directed graph $G = (V, E)$ with edge-weights, a set of terminals (vertices) $S \subseteq V$, and a root vertex r, find a minimum weight connected tree rooted at r, such that all vertices in S are included in the tree [12]. It is easy to see that any instance of the directed Steiner tree problem is equivalent to the degenerate CCD problem where there is only one broker in the network, the content adaptation graph CAG is set to be the same as G, the vertices's in S correspond to the set of formats (corresponding to a set of nodes in the CAG) in which content is required, and r is the original format of content. Since the CCD problem is NP-hard for the case of one broker, it remains NP-hard in the general case as well. $\square$

3 Multilayer Graph representation of CCD

An interesting observation is that CCD problem can be formulated as a minimum directed Steiner tree problem in a *multilayer graph* constructed from the given CAG and dissemination tree. In fact this observation was made in [13] for multicasting problem. A multilayer graph for CCD problem is constructed by combining the dissemination tree and the content adaptation graph (CAG) as follows:

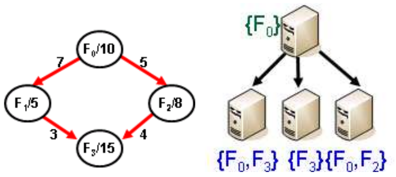

Fig. 3. A sample subtree and a CAG.

Generate m replicas of the dissemination tree, each representing a layer corresponding to a format in the CAG (m is the number of formats in the CAG). The restriction being that within each layer, data can be transmitted along the edges in the format corresponding to that layer only. We denote the multilayer graph by $\mathcal{G}_{\mathcal{ML}} = (\mathcal{V}, \mathcal{E})$ such that $\mathcal{V} = V_d \times V_c$ where V_c is the set of vertices in CAG and V_d is the set of nodes in the dissemination tree. Each vertex in $\mathcal{V}$ is therefore associated with exactly one pair of nodes - where the first member of the pair is a node in the dissemination tree and the other corresponds to a format in the CAG. For a vertex v in a multilayer graph the corresponding format in the CAG is referred by $v.format$ and the corresponding node in the dissemination tree by $v.node$. The edge set of $\mathcal{G}_{\mathcal{ML}}$ comprises the following two kinds of edges - edges that connect two nodes in the same layer (called *transmission edges*) and edges that connect nodes across layers (called *conversion edges*. There is a directed transmission edge in every layer corresponding to a link in the original dissemination tree. Similarly, there is a directed conversion edge joining the vertices corresponding to the same (physical) node across layers L_i and L_j if and only if there is an edge from format F_i to F_j in the CAG. The weight of a transmission edge in layer L_i is equal to the transmission cost of its corresponding format, i.e., F_i. Similarly, the weight of a conversion edge between two layers L_i and L_j is the same as the conversion cost from format F_i to F_j in the CAG. As discussed in Section 2.2, we will assume that the transmission cost and conversion cost are measured in the same unit and have been normalized, i.e., one unit of transmission cost is same as one unit of conversion cost.

Now, it is easy to see that any valid plan for the CCD problem can be represented as a tree in the corresponding multilayer graph. In fact, the minimum cost plan for a CCD problem corresponds to the minimum cost directed Steiner tree in $\mathcal{G}_{\mathcal{ML}}$. For each format F_k that is assigned to a link between N_i and N_j in the optimum plan, the transmission edge between corresponding nodes for N_i and N_j in the layer associated to F_k is also included in the minimum cost Steiner tree. For each operator $O_{(s,w)}$ assigned to node N_i in the optimum plan, the conversion edge between corresponding nodes for N_i between the layers associated with F_s and F_w is included in the minimum cost Steiner tree. Finally, one can see that cost of the optimal CCD plan is the same as the total weight of the edges in the minimum Steiner tree in the multilayer graph.

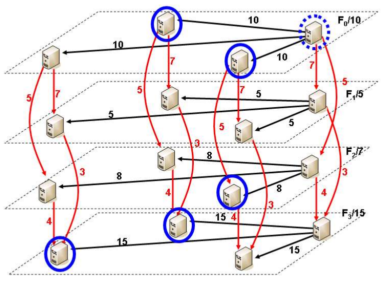

Fig. 4. Multi layer graph for Figure 3.

As an example consider the CAG and dissemination tree depicted in Figure 3. Figure 4 depicts the associated multilayer graph for the given CAG and dissemination tree. The source for the Steiner tree in the multilayer graph is the corresponding node for the dissemination tree's root in the layer associated with the initial format. The set of terminals for the Steiner tree consists of the corresponding nodes for subscriber brokers in their layers associated with their requested formats. Therefore, in the next two sections we develop two CCD algorithms that generate CCD plans with a small cost. In fact our first algorithm is designed to find the optimum CCD plan and can be used when the number of formats in CAG is small (less than 5). The second algorithm is meant for large CAGs (more than 5 nodes) and uses heuristics to generate low cost plans.

4 Optimal CCD Algorithm

In this section we describe an algorithm for finding minimum cost dissemination plan when the CAG contains small number of formats (less than 5). In many situations we may be able to categorize the devices into a small set of classes where determining an optimum dissemination plan is possible. For instance, in an image dissemination system , e.g., “PC with high speed connection”, “PC with dial-up connection”, “Mobile device with Wi-Fi connection” and “Mobile device with GSM connection”. An important advantage of this algorithm over the multilayer graph based approach is that it scales linearly with the dissemination tree size and can therefore be used for efficiently computing the optimal plan for large dissemination trees when the CAG is small.

Let us describe the main idea behind the optimal algorithm using an example. Consider a broker N_i that receives content in formats specified in the set $F_{in}^{N_i}$ from its parent (as shown in Figure 5). Let N_i have two children N_j and N_k. Let us assume that for every child node the minimum-cost dissemination plan for the subtree rooted at the node is known in advance for each possible input format set (recall, the sub-plan cost includes the transmission cost along the incoming edge at the node). Now, if the number of formats in the CAG is m, there are potentially 2^m distinct input sets for each child. Given the costs of these 2^m optimal sub-plans for each child of N_i (shown as arrays in the figure), let us see how to find the minimum cost plan for the subtree rooted at N_i parameterized on its input $F_{in}^{N_i}$. Take the simple case when $F_{in}^{N_i}$ is a singleton set $\{F_2\}$ from the CAG shown in Figure 2. To compute the minimum cost for this specific input, we generate all the formats that can be potentially generated from $\{F_2\}$ (based on the CAG) and note the corresponding conversion costs. For our example CAG, let the format sets generated from $\{F_2\}$ at N_i be denoted by $\mathbb{F}_i^* = \{\{F_2\}, \{F_3\}, \{F_2, F_3\}\}$. Of course, in the worst case $|\mathbb{F}_i^*| = 2^m$. Now, given the input $\{F_2\}$ at N_i, the best plan is the one that minimizes the sum of transmission cost of content in format F_2 to N_i (from its parent), the costs of the least expensive plans at N_j and N_k when inputs at N_j and N_k are restricted to be an element of $\mathbb{F}_i^*$ and the corresponding conversion cost at N_i to generate the union of the two input sets for N_j and N_k from $\{F_2\}$. Observe that irrespective of what formats are sent to N_j and N_k, their union has to be an element of $\mathbb{F}_i^*$. We use this observation to efficiently compute the best sub-plan for input $\{F_2\}$ at N_i as follows: For each $f_i^* \in \mathbb{F}_i^*$, determine input sets $f_j^* \subseteq f_i^*$ for N_j and $f_k^* \subseteq f_i^*$ for N_k independently such that the sub-plan cost at N_j and N_k are minimized. Add to the sum of these two costs, the cost of conversion from $\{F_2\}$ to f_i^* (i.e., the Minimum directed Steiner tree cost denoted as $\mathbb{S}(\{F_2\} \rightsquigarrow f_i^*)$). When there are k children this operation can be completed in $O(k.2^m)$ time if m is small and the array at each node is sorted in increasing order of sub-plan costs. We simply need to determine the minimum total cost over f_i^* (i.e., best element in $\mathbb{F}_i^*$). Since $|\mathbb{F}_i^*|$ is at most 2^m, we can determine the best plan for any given input at N_i ($\{F_2\}$ in this case) in $O(k.2^{2m})$. Further, since there are 2^m distinct inputs possible, we can fill the array at N_i in $O(k.2^{3m})$ time in the worst case.

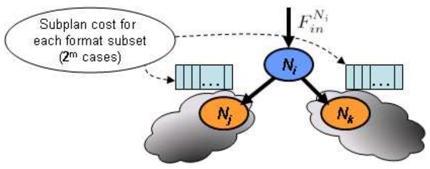

Fig. 5. Optimal CCD for a node.

Given the optimal substructure characteristic of this problem, we can give a dynamic programming based algorithm that computes the minimum cost plan for the CCD problem. Algorithm 1 shows the steps required for one broker N_i in the dissemination tree for a specified input format set $F_{in}^{N_i}$. The algorithm needs the input format set along with the dissemination subtree rooted at N_i and the list of arrays consisting of the best plans at its children nodes ($ChildSubPlans_{N_i}[]$). As mentioned, the algorithm assumes that the minimum cost plan for all input format sets are available for every child of N_i. We then initialize the empty plan $\mathbb{P}$ with infinite cost (Lines 4-5). Now, for each possible output format set at N_i the algorithm first finds the conversion cost using a directed Steiner tree algorithm [12] in line (8). Note that the minimum conversion cost can be computed efficiently since the CAG is assumed to be small. Then it computes the least expensive plan as illustrated in the example above (Lines 9-13). If the newly computed plan had smaller cost than the previous plans, the algorithm updates the minimum plan and its cost (Lines 14-17). Finally, the computed minimum plan's cost is updated by the transmission cost of the input format set and is returned as the minimum cost plan.

To find the minimum cost plan for a given dissemination tree, we call the *OptimalCCD* algorithm in the RP broker with $F_{in}^{RP} = \{F_0\}$. After running the algorithm, the minimum cost plan is available and the system uses it to detect which operators must be executed in each broker and which content formats must be transmitted over each link. Each node in the dissemination tree receives the content formats along with the portion of plan corresponding to the subtree rooted at that node. It then investigates the received plan and performs the

operators that are assigned to it in the plan and forwards the content in the formats indicated in the plan to each of its children along with their corresponding parts of the dissemination plan.
Theorem 2: The complexity of the optimal CCD algorithm is $O(nk_{avg}2^{3m}\mathbb{S})$, where n is the number of nodes in the dissemination tree, m is the number of formats in the CAG, k_{avg} is the average number of children a node has and $\mathbb{S}$ is the complexity of computing the minimum cost directed Steiner tree in the CAG .
Proof: The algorithm is recursively called for each node and there are n nodes in all. Now, if we denote the average number of children of a node in the dissemination tree by k_{avg} and the maximum cost paid for an instance of the "Minimum Steiner Tree" problem at any node by $\mathbb{S}$ (which is assumed to be almost linear due to the small value of m), then from the analysis done in the example above (Figure 5), we can show that the worst case complexity of Algorithm 1 is $O(nk_{avg}2^{3m}\mathbb{S})$. □

Algorithm 1 OptimalCCD

```
1: INPUT: $F_{in}^{N_i}$ (Set of input formats), $\mathbb{T}_{N_i}$(Dissemination subtree rooted at $N_i$),
   $ChildSubPlans_{N_i}$()(List of best child subplans)
2: OUTPUT: $\mathbb{P}_i$: Least cost subplan at $N_i$ for input $F_{in}^{N_i}$;
3:
4: $\mathbb{P} \leftarrow$ Empty plan;
5: $\Theta_{\mathbb{P}} \leftarrow \infty$;
6: for all $F_{sub} \in PowerSet(F)$ do
7:    $\mathbb{P}_{temp} \leftarrow$ Operators performed in $N_i$;
8:    $\Theta_{\mathbb{P}_{temp}} \leftarrow \alpha\mathbb{S}(\mathbf{F_{in}^{N_i}} \rightsquigarrow \mathbf{F_{sub}})$; // $\mathbb{S}(F_{in}^{N_i} \rightsquigarrow F_{sub})$:the minimum cost of converting content
      from a set of available formats, $F_{in}$ into set of output formats, $F_{out}$
9:    for all $N_j \in$ Children($N_i$) do
10:       $\mathbb{P}_{N_j} \leftarrow$ MIN{$ChildSubplans_{N_i}(N_j)$} s. t. $F_{in}^{N_j} \subseteq F_{sub}$;
11:       Add $\mathbb{P}_{N_j}$ to $\mathbb{P}_{temp}$;
12:       $\Theta_{\mathbb{P}_{temp}} + =$ TotalCost($\mathbb{P}_{\mathbf{N_j}}$); // $TotalCost(\mathbb{P})$: the total cost of the plan
13:    end for
14:    if $\Theta_{\mathbb{P}} > \Theta_{\mathbb{P}_{temp}}$ then
15:       $\mathbb{P} \leftarrow \mathbb{P}_{temp}$;
16:       $\Theta_{\mathbb{P}} \leftarrow \Theta_{\mathbb{P}_{temp}}$;
17:    end if
18: end for
19: $\Theta_{\mathbb{P}} + = (1-\alpha)\sum_{F_k \in F_{in}^{N_i}} \mathcal{T}_{F_k}(\mathbb{C})$;
20: return $\mathbb{P}$;
```

During implementation, the optimal CCD algorithm can be sped up by reducing the number of format sets to be considered in the output set of a node. If we cannot derive a particular format set from the input format set, there is no need to compute those sub-plans. However, in the worst case where CAG is a fully connected directed graph the algorithm may need to consider all 2^m subsets of formats.

5 CCD Problem for Large CAGs

In this section we present an iterative algorithm for CCD problems with large CAGs. Given an initial CCD plan, the algorithm iteratively selects a node in the dissemination tree and refines the local plan at the node to reduce the cost of the solution. The refining process may include the following two actions: (i) changing the conversion operators at this node and its children; (ii) changing the set of formats in which content is transmitted to each one of its children. The modified plan always has a cost lower than the previous one and acts as an input for the next iteration. The iterative CCD algorithms is shown below (Algorithm 2). We show through extensive experimentation (in Section 6) that these heuristics work very well in practice. In fact, in Section 6 we show empirically that the final plan costs are within a small factor of the minimum possible cost by establishing a theoretical lower bound to the cost of a CCD plan.

Algorithm 2 Iterative CCD algorithm for large CAG

```
1: INPUT: $\mathbb{P}$: The initial plan, $\mathcal{K}$: Number of iterations ;
2: OUTPUT: $\mathbb{P}$: The refined plan;
3:
4: for all $j = 0$ to $\mathcal{K}$ do
5:     $N_i$ = SelectNode($\mathbb{P}$)
6:     RefinePlan( $\mathbb{P}$,$N_i$)
7: end for
8: return $\mathbb{P}$;
```

The algorithm starts with an initial plan, then greedily selects a node using the *SelectNode* function call and applies the *RefinePlan* procedure to generate a better plan. In general, one may use a variety of criteria for termination, such as the magnitude of change in cost over successive iterations, number of iterations, time bound etc. In this paper, we just iterate for a fixed $\mathcal{K}$ times which is provided by the user as an input parameter. Next, we present details about the initialization, node selection and plan refinement steps of our iterative algorithm.

Step 1: Initial Plan Selection We can initiate the above algorithm using any valid plan. In this paper, we seed the algorithm using either one of the three following strategies. We call the first plans the *All-in-root* plan and the second one *All-in-leaves* plan. Both of these algorithms avoid in-network placement of customization operators and perform all the required operators either at the dissemination tree's root or at the leaves. The *All-in-root* CCD algorithm generates all the required formats in the dissemination tree by performing the necessary operators in the root (RP). Then, the generated content in various formats are forwarded towards the leaves based on their requests. On the other hand, the *All-in-leaves* CCD algorithm forwards the published content to all leaves and all of the nodes with matching subscription convert the content into the formats requested by its clients from the original format. We refer to the third initial plan as the *Single-format* plan. In this plan content is transmitted over a link exactly in one format, the one with the smallest transmission cost.

Step 2: Node Selection for Plan Refinement We considered several strategies for node selection. The first strategy is to select the nodes of the tree randomly in every iteration. We will refer to this as the Random scheme. While the random scheme is the most obvious, a smarter approach would be to base the selection on some estimation of the potential cost-reduction one can achieve by *refining* a given scheme. We use a greedy heuristic that selects the next node (i.e., dissemination plan) based on the difference between the current cost of a sub-plan and the estimated *lower bound* to the minimum achievable cost for the sub-plan. The *SelectNode* function returns the node N_i^* from the set of all nodes N_i in the tree such that the *slack* in the total cost paid in the local region of N_i is maximized. The *slack = (total conversion cost paid in the local region of* N_i *- the lower bound of the total conversion cost in the local region of* N_i*) + (total transmission cost in the local region of* N_i *- lower bound to the total transmission cost in the local region of* N_i*).* We will refer to this as the SLACK scheme from here onwards. Since our cost model consists of content transmission and content conversion costs, to find a lower bound for a plan we need a lower bound for each of these components in the total cost. We describe how the lower bounds are computed next.

Transmission-cost lower bound: We define a lower bound for transmission cost for each link in the dissemination tree and define the lower bound for the tree as the sum of the lower bounds for each one of its links. Consider a link $< N_i, N_j >$ in the dissemination tree where N_j is a leaf node. The content formats transmitted over this link depend on the formats requested by the clients attached to N_j. Consider the case where content is requested only in F_k by the clients at N_j. Since the transmission costs are proportional to the "size" of the content format, the minimum transmission cost for the link is at least as much as the size of the smallest format in the CAG that we can convert into F_k. In other words, the minimum transmission cost along $< N_i, N_j >$ corresponds to the transmission cost for the format with the smallest size, say F_k^{min} such that there is a path from F_k^{min} to F_k in the CAG. In general, if the content is required in more than one format at a node, say $\{F_{k_1}^{min}, \ldots, F_{k_l}^{min}\}$ we can compute the corresponding smallest formats and take the transmission cost of the largest of these as the lower bound for the link. This lower bound applies to edges between internal nodes of the dissemination tree as well. The set of formats requested at any internal node N_t is simply taken to be the union of formats requested at any client of a node in the subtree rooted at N_t. Below, we describe how one can quickly determine such a format for any link in the dissemination tree.

We maintain a sorted array of all the formats in the CAG in ascending order of their transmission costs. This is a one time operation which takes $O(mlog(m))$ time at most. Then, for a given target format F_k we go down the array and select the smallest format such that there is a path from this format to F_k in the CAG (this could very well be F_k itself). The transmission cost of this format is chosen as the lower bound for F_k. When the content is required in multiple formats in the subtree rooted at the child node of the link, we determine the lower bound for each format separately and set the largest of these as the lower bound for the link. The lower bound to the transmission cost of the whole tree (subtree) is simply the sum of the lower bounds for every link in the tree(subtree). We will denote this by $\mathcal{T}_{low}(t)$ for a subtree t or simply by $\mathcal{T}_{low}$ for the whole tree.

Conversion cost lower bound: Computing the lower bound for the total conversion cost is straightforward. The minimum conversion cost that needs to be paid for a plan is the cost of converting the original format into all the requested formats in the tree at least once. This is simply the cost of minimum directed Steiner tree of the CAG where the set of terminals is the set of all requested formats. We will denote this global lower bound to the conversion cost by $\mathcal{C}_{low}$. Note, in contrast to the transmission cost which is a positive number for every link in the dissemination tree, the lower bound for conversion cost is zero for each node because there is always a valid plan in which no operation is performed at a given node. As a result the lower bound for conversion cost of any node is 0.

Algorithm 3 RefinePlan.

```
1: INPUT: $\mathbb{P}$: The initial plan, $N_i$: Selected node ;
2:
3: $\mathcal{G}_{\mathcal{ML}}(\mathcal{V}, \mathcal{E})$ = createMLGraph($N_i$);
4: $Source \leftarrow \phi$; //Set of source vertices;
5: $Terminal \leftarrow \phi$; // Set of terminal vertices;
6: for every $v \in \mathcal{V}$ do
7:     if $v.node = N_i$ AND $v.format \in F_{in}^{N_i}$ then
8:         $Source = Source \cup \{v\}$;
9:     end if
10:    if $N_j \in Children(N_i)$ AND $v.node = N_j$ AND $v.format \in F_{out}^{N_j}$ then
11:        $Terminal = Terminal \cup \{v\}$;
12:    end if
13: end for
14: SteinerTree = MinSteiner($\mathcal{G}_{\mathcal{ML}}, Source, Terminal$);
15: if SteinerTree.cost < SubPlanCost($\mathbb{P}$,$N_i$) then
16:     Update($\mathbb{P}$);
17: end if
```

Step 3: Plan Refinement using Multilayer Graph The *RefinePlan* procedure takes as input a valid plan and a node N_i and updates the plan to a new one with smaller cost by modifying conversion operations and transmissions in the local region of N_i. Algorithm 3 shows the steps of *RefinePlan* procedure. In line 3 it creates the multilayer graph corresponding to the local region of N_i. In other words, it creates the multilayer graph corresponding to the "stump" of the sub-plan underneath N_i involving N_i and its children only. Therefore, the refinement step focuses on the conversion operation performed at one of these nodes and the transmission formats along the links between N_i and its children in the current plan. Next, the source and terminal nodes for the minimum cost Steiner tree computation in the multilayer graph must be determined. Any vertex with N_i as its associated node and one of the input formats in $F_{in}^{N_i}$ is added to the set of source vertices for the Steiner tree. Similarly any vertex in the multilayer graph that corresponds to one of N_i's children and an output format of the child in the current plan is added to the set of terminals for the Steiner tree. Lines 6-13 show the steps of forming these source and terminal sets. Once these sets have been determined, we use an approximation algorithm for Steiner tree computation [12] as shown in line 14. Finally, if the total cost of the computed Steiner tree is strictly smaller than the cost before refinement the plan is updated to the reflect the new operations and transmissions in the dissemination tree as described below (lines 15-17).

The Update($\mathbb{P}$) process does the following: For each transmission edge in the Steiner tree, the format associated to the layer is added to the set of formats that are transmitted through the link between N_i and the corresponding child. Similarly, for each conversion edge in the Steiner tree the corresponding operator in the CAG is added to the list of operators that are performed at the associated node in the current plan. Note that the input format set for N_i and the output format sets for N_i's children remain unchanged after the call to *RefinePlan* procedure. Since we use approximate Steiner tree algorithm, the Steiner tree may result in the higher cost plan where in this case no action is taken. It is easy to see that the refined plan remains a valid plan after performing an update.

Note that since we construct a multilayer graph for a node and its children only, the size of the graph is significantly smaller than the multilayer graph for all of the dissemination tree. Assume the maximum number of children for a node in a network with 1000 brokers is 10 and there are 10 formats in the CAG and all formats are requested in every child. The multilayer graph in this case has 150 vertices. The complexity of the Steiner tree algorithm for $\mathcal{K}$ iteration is $O(160^2 \times 150^4 \times \mathcal{K})$ which is significantly less than $O(10000^2 \times 3500^4)$ which was the complexity of the example in Section 3.

6 Experimental Evaluation

6.1 System setup

To evaluate our algorithms we developed a message level, event-based simulator on top of Tapestry routing scheme. We implemented our algorithms and customization operators in Java. Since the focus of this paper is content dissemination among brokers, we performed our simulations only for the broker overlay. There are 1024 brokers in the overlay network. We use the *matching ratio* as our main parameter, which is the fraction of the brokers that have matching subscriptions for a published content. As argued in [4], studying the behavior of our algorithms over the range of matching ratios enables us to interpret the results for both Zipf and uniform distribution of publications and subscriptions over the content space. For instance, the behavior of the algorithms for Zipf distribution in which a small portion of the event space is very popular while the majority of the event space has only few subscribers can be shown by the behavior of the algorithm for very high and very low matching ratios. For each matching ratio, the reported results are averages taken for 100 runs. We also use tree discovery message to detect the dissemination tree and the node and link costs. We account for the computation cost of performing our algorithms and the communication overhead of tree discovery message. Based on our prototyping, the average execution time of the algorithms was about 100ms and we set the probe message size to be 0.1 KB. Publishers and subscribers in the broker overlay are selected randomly for each run. Similarly, the requested formats by a subscriber are sampled uniformly at random from the set of all formats. Each broker has subscriptions for at most $\frac{1}{4}$ of the available formats in the CAG. The default value for α and β is set to 1 in the cost function indicating that the normalized communication and computation cost units have equal weight.

6.2 Dissemination scenarios

For our experimental study we used variety of small and large CAGs, however, because of space limitation in this section we present our results for two CAGs representing two dissemination scenarios. The first CAG is a small one that is used to evaluate our optimal CCD algorithm while the second one is a large CAG that is used to evaluate the heuristic based CCD algorithm.

Annotated Map Dissemination: For the first scenario, we considered customized dissemination of annotated maps to subscribers in the context of emergency. For instance, in case of wild fire an annotated map depicting shelters for evacuees and open roadways in a specific geographic region might need to be disseminated to the local population.

The published content in this scenario is an annotated map along with brief text description about each annotated item. Our system provides content in four different formats. The original format of the annotated map is PDF (F0). Depending on their preference and device, receivers can request the content in JPG image format (F1), text format (F2) or voice format which is text to speech conversion of the first annotated item(F3). For PDF to JPG and Text customizations we used PDFBox package (http://www.pdfbox.org/) and for Text to Voice conversion we used FreeTTS package (http://freetts.sourceforge.net/). Figure 6 depicts the corresponding CAG where the costs were computed based on our extensive prototyping.

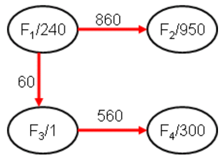

Fig. 6. Sample content and CAG for Annotated Map scenario.

Customized Video Dissemination: In the second scenario we consider dissemination of video content in variety of formats. In this scenario the CAG has 16 formats. The original content is in high quality 'mpeg4' format. The CAG contains four nodes in 'mpeg4' format that differ in frame size and bit rate. Also there are four nodes in CAG for each of 'avi', 'flv' and '3gp' formats. Similarly, each of these nodes represent specific frame size and bit rate for the video content. We also measure the content adaptation costs in the CAG based on extensive prototyping of possible transcoding between the available formats in the CAG. The costs of nodes in this CAG are in the range of [0,30]. For video transcoding we used *FFmpeg* which is a complete, cross-platform solution to record, convert and stream audio and video and includes libavcodec - a leading audio/video codec library[3]. The edge costs in this CAG are in the range of [0,60]. Because of very complex representation of this CAG (16 vertices and 210 edges) we only represent the CAG with 24 edges out of 210 edges in Figure 8.

6.3 Experiments

Based on the described system setup and the CAGs we present set of experiments that aim to evaluate the following:

- The effect of using optimal and heuristic CCD algorithms in reduction of content dissemination cost.
- The quality of the heuristic CCD algorithm results.
- The effect of different parameters.
- The effect of the relationship between communication and computation costs on the algorithm.

We use the small CAG from the annotated map scenario in the first two experiments to evaluate the benefit of using CCD algorithms and quality of the heuristic CCD algorithm compared to the optimal one. In the rest of experiments

[3] For information on FFmpeg please refer to "http://www.ffmpeg.org/".

we use the large CAG of the video dissemination scenario to evaluate different factors that are involved in the heuristic CCD algorithm.

Effect of CCD algorithms on cost: In this experiment we evaluate the effect of using the proposed CCD algorithms in reducing the dissemination cost. We compare our CCD algorithms with two alternative approaches, *All-In-Leaevs* (AIL) and *All-In-Root* (AIR). Figure 6.3 represents the percentage of savings in the dissemination cost in our CCD algorithms compared to the AIL and AIR approaches for different α and β ratios. The first graph depicts the results for the optimal CCD algorithm and the small CAG and the second one shows the results for the heuristic CCD algorithm and the large CAG. As it can be seen in both cases using CCD algorithms result in reduction of dissemination cost, however, the amount of saving may significantly vary for AIL and AIR approaches as α and β change. The amount of cost reduction depends on several factors including the communication and computation costs in the CAG, the number of different requested formats in brokers and the relationship between communication and computation costs in the system. An interesting fact shown in the graphs is that the CCD algorithms result in much higher savings as compared to the AIL approach when $\frac{\alpha}{\beta} = 0.1$. In contrast, when $\frac{\alpha}{\beta} = 10$ the AIR approach performs much worse than the CCD algorithms. The reason is when $\frac{\alpha}{\beta} = 0.1$ computation cost unit is much higher than communication cost unit and since AIL performs operators in leaves, an operator may be performed several times which results in higher total cost. In such cases as expected the difference between CCD plans and AIR is not very significant because the computation cost is minimized in AIR. On the other hand, when $\frac{\alpha}{\beta} = 10$ the generated plans by CCD algorithms are closer to AIL because communication cost is higher whereas AIR results in higher communication cost because of redundant transmission of the same content in different formats over some links. In general, these results show that regardless of CAG and requested formats in brokers, using our CCD algorithms always results in reduction of dissemination cost compared to at least one of the AIL or AIR approaches.

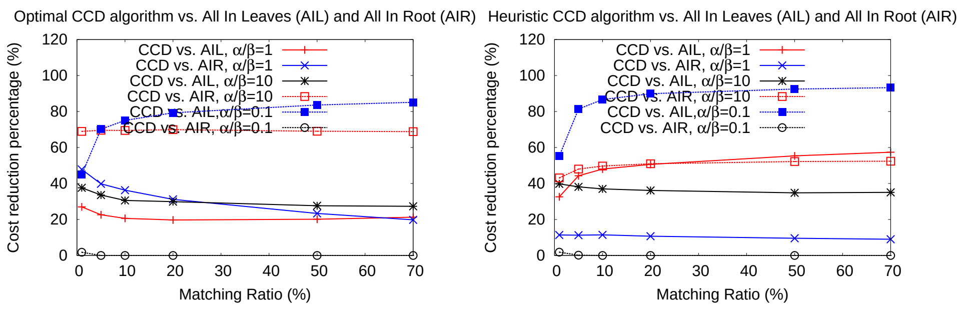

Fig. 7. Cost reduction percentage in Optimal and Heuristic CCD algorithms compared to AIL and AIR for different α and β values.

Quality of CCD heuristic: In this experiment we evaluate the effectiveness of the heuristic CCD algorithm in finding a dissemination plan. We compare the cost of the plan resulting from the heuristic CCD algorithm with the cost of the optimal dissemination plan that has the minimum cost. Since finding the minimum cost plan when the CAG is large is NP-hard we use our small CAG in this experiment. The minimum cost plan in this experiment is computed using our optimal CCD algorithm. Figure 9 depicts the percentage of cost difference between the minimum cost plan and the plan resulting from the heuristic CCD algorithm for 1000 iterations. The cost difference after a few iterations sharply falls to around 1% for all matching ratios. This shows that the proposed heuristic CCD produces dissemination plans significantly close to the minimum dissemination plans. Also this plan is achieved with very small number of iterations in the heuristic CCD algorithm.

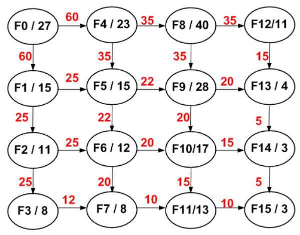

Fig. 8. Video dissemination CAG with subset of edges.

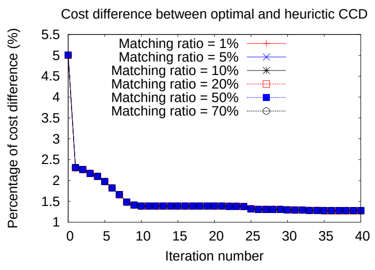

Fig. 9. Goodness of the heuristic CCD algorithm compared to the optimal algorithm.

In the previous experiments we showed that the CCD algorithms reduce the dissemination cost and the heuristic CCD algorithm results in close to optimal dissemination plans. In the rest of the experiments in this section we present the effect of different parameters on the effectiveness of the heuristic CCD algorithm.

Initial plan selection: In this experiment we compare three different dissemination plans, All In Root (AIR), All In Leaves (AIL) and Single format (SF). An important factor that affects the final plan cost is the relationship between communication and computation costs in the system. If the communication resources in a system are more expensive than computation resources, the initial plan that is used for the heuristic CCD algorithm may be different than when the computation resources are costlier than the communication resources. Figure 10 plots the costs of three initial dissemination plans for different matching rations in three different scenarios. As it is seen when the computation resources have more importance in the system ($\frac{\alpha}{\beta} = 0.1$), the AIR initial plan has smallest cost for all matching ratios. This is clear because of AIR plans have minimum computation cost. On the other hand, if the communication resources are more expensive, AIR plan results in more consumption of communication resources and therefore results in larger dissemination cost. Therefore, AIR s the worst initial plan when $\frac{\alpha}{\beta} = 10$. As it is seen in this case SF is a better initial plan to consider.

Note that these results are for specific CAG and subscription distribution among brokers. We have similar results for different CAGs and subscription distributions where single format or All In Root may result in better initial plan. Therefore, we conclude that to find a better initial plan, the heuristic

CCD algorithm computes all possible initial CCD plans and selects the one with the smallest cost as the initial plan for refining the plan using iterations.

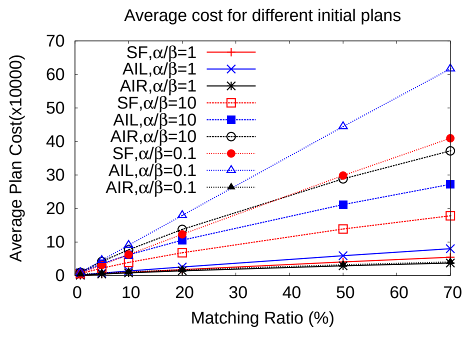

Fig. 10. Initial plan comparison for different α and β values.

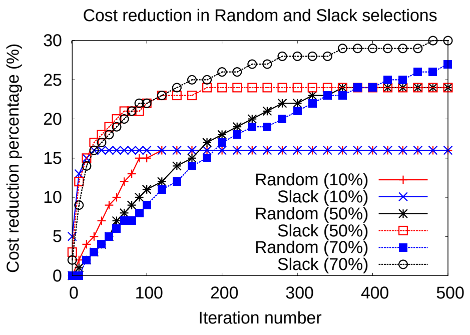

Fig. 11. Next node selection effect on cost reduction rate

Next step selection: In this experiment we evaluate the random and slack based selection techniques. Figure 11 depicts the percentage of cost improvement compared to the cost of initial plan for 500 iterations and three matching ratios, 10%, 50% and 70%. As it is seen for all matching ratios the rate in which the slack based techniques refines the dissemination plan to lower cost plan is significantly faster than the random technique. For instance, in 70% matching ratio the slack based technique results in 25% reduction in cost after around 150 iterations while it takes more than 500 iteration for the random technique to achieve the same percentage in cost reduction. Therefore, if we limit the number of iterations that the heuristic CCD algorithm performs for refining the plan, the slack based technique is superior to the random one. Another fact that is shown in the figure is that regardless of the next step selection technique, both random and slack based heuristic CCD algorithms converge to the same final dissemination plan after sufficient number of iterations. This means if there are enough resources available for a large number of iterations, both techniques achieve the same final refined dissemination plan.

7 Related Work

Most of the existing pub/sub systems have concentrated on providing efficient dissemination service for simple publication formats such as numerical or text content [1, 3, 5]. Shah *et al.* studied filter placement in content-based pub/sub network [15]. The objective of this approach is to minimize the total network bandwidth utilization resulting from dissemination of published content. However, their system does not consider the overhead resulting from filter operations in the cost function and only consider single filtering operation type. The content format also is not customized and published content is delivered in the same format to all receivers.

Diao *et al.* proposed ONYX, a customized XML dissemination framework that provides scalability and expressiveness [16]. ONYX provides incremental message transformation by using early projection and early restructuring of content. However, since content transformation operations are XML filtering and restructuring operations, ONYX does not consider overhead of transformation and only aims to minimize content transmission overhead.

The Echo pub/sub system is a high performance event delivery middleware designed for grid environments with large scale event rates [19]. While Echo provides event filtering and transformation service in pub/sub system, there are significant differences between Echo and our proposed CCD approach. Unlike CCD which is proposed for content-based pub/sub systems, Echo is a channel-based pub/sub system. Event types define C-style structures made up of atomic data types. For event filtering and transformation Echo extends event channels via derivation. However, all the required computation for filtering and transforming events are performed in the same source node for the original event channel.

Some multimedia content dissemination systems expand the multicasting concept by providing content customization services for group members. In [14], Lambrecht, *et al.* formally defined the multimedia content transcoding problem in a multicast system and provided heuristic algorithms for transcoding content into the format that is requested by each receivers. A similar system has been proposed in [13] where the multicast tree is mapped into a multilevel graph and an approximate Steiner tree algorithm to find efficient content transcoding in the network. However, unlike our proposed system, both of the systems assume that the multicast group is a fixed and predefined group. Also these systems only consider dissemination of multimedia content in the same file format which is a subset of the problem we consider here.

Content customization has been subject to extensive research in multimedia community. Nahrstedt *et al.* proposed *Hourglass* [17], a multimedia content customization and dissemination framework. Hourglass composes requested content formats from specified sources by efficiently placing composition services in the network and disseminates composed format to receivers in their requested formats. However, Hourglass assumes each adaptation service is performed only once in the system and also content dissemination is done using multiple dissemination trees: one for each content format. Both of these assumptions significantly simplify the customized content dissemination problem.

8 Conclusions and Future Work

We have introduced customized content dissemination system where content is only delivered to receivers that have requested it and in their desired format. We proposed operator placement algorithms on top of a DHT-based pub/sub framework in order to customize content format such that dissemination cost, which we defined as a linear function of customization (computing) and transmission (communication) costs, is minimized. We formally defined the problem and showed that it is NP-hard. We proposed two approaches to generate an efficient operator placement plan. Our first algorithm, the optimal CCD, finds the minimum cost CCD plan when the number of requested formats in the system is small. For the scenarios with large number of required formats we proposed an iterative heuristic algorithm that considerably reduces the CCD cost compared to performing customizations in the dissemination tree root or in the receiver brokers. We also showed the benefit of using our algorithms through extensive experiments. We have extended our proposed algorithms to take into account the heterogeneity of brokers and links along with the effect of concurrent publications in computing dissemination plans. However, due to the space limitation we did not present these extensions along with the corresponding experimental results in this paper.

In the heuristic CCD algorithm we used a multilayer graph for a subtree of depth one in the dissemination tree. As part of our future work we are investigating the trade-off in choosing subtrees with higher depth and complexity of the minimum directed Steiner tree computation. We are also working on a heuristic algorithm based on our Optimal CCD algorithm to generate a more effective initial plan for our heuristic CCD algorithm when the CAG is large. We are also investigating other cost models including dissemination time and the ways that the CCD algorithms can be adapted for such cost models.

References

1. S. Castelli, P. Costa, G. P. Picco,*HyperCBR: Large-Scale Content-Based Routing in a Multidimensional Space.* IEEE INFOCOM 2008.
2. www.twitter.com
3. I. Aekaterinidis, P. Triantafillou, *PastryStrings: A Comprehensive Content-Based Publish/Subscribe DHT Network.* IEEE ICDCS 2006.
4. F. Cao, J. Pal Singh, *MEDYM: Match-Early with Dynamic Multicast for Content-Based Publish-Subscribe Networks.*, ACM/Usenix/IFIP Middleware 2005.
5. G. Li, V. Muthusamy, H.A. Jacobsen, *Adaptive Content-Based Routing in General Overlay Topologies.* ACM/Usenix/IFIP Middleware 2008.
6. D. Tam, R. Azimi, H.A. Jacobsen, *Building Content-Based Publish/Subscribe Systems with Distributed Hash Tables.* International Workshop On Databases, Information Systems and Peer-to-Peer Computing 2003.
7. B. Y. Zhao, L. Huang, J. Stribling, S. C. Rhea, A. D. Joseph, and J. Kubiatowicz. *Tapestry: A Resilient Global-scale Overlay for Service Deployment.*, IEEE Journal on Selected Areas in Communications, Vol. 22(1),2004.
8. A. Rowstron and P. Druschel, *Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems.* ACM/Usenix/IFIP Middleware 2001.
9. S. Q. Zhuang, B. Y. Zhao, A. D. Joseph, R. H. Katz, J. Kubiatowicz, *Bayeux: An Architecture for Scalable and Fault-tolerant Wide-Area Data Dissemination*, Proceedings of ACM NOSSDAV 2001.
10. R. Baldoni, C. Marchetti, A. Virgillito, R. Vitenberg, *Content-Based Publish-Subscribe over Structured Overlay Networks.*, IEEE ICDCS 2005.
11. A. Gupta, O. Sahin, D. Agrawal, A. El Abbadi, *Meghdoot: Content-Based Publish/Subscribe over P2P Networks.* ACM/Usenix/IFIP Middleware 2004.
12. M. Charikar, C. Chekuri, T. Cheung, Z. Dai, A. Goel, S. Guha, and M. Li, *Approximation algorithms for directed steiner problems.*, ACM-SIAM symposium on Discrete algorithms, 1998.
13. Henig, A. and Raz, D., *Efficient management of transcoding and multicasting multimedia streams.*, 9th IFIP/IEEE International Symposium on Integrated Network Management, 2005.
14. T. Lambrecht, B. Duysburgh, T. Wauters, F. De TurckBart Dhoedt and P. Demeester., *Optimizing multimedia transcoding multicast trees.*,Computer Networks, Volume 50, Issue 1, 16 January 2006, Pages 29-45.
15. R. Shah, Z. Ramzan, R. Jain, R. Dendukuri and F. Anjum,*Efficient Dissemination of Personalized Information Using Content-Based Multicast.*, IEEE Trans. Mob. Comput. 3(4) 394-408,2004.
16. Y. Diao, S. Rizvi, and M. J. Franklin, *Towards an Internet-Scale XML Dissemination Service.*, VLDB Conference, August, 2004.
17. K. Nahrstedt, B. Yu, J. Liang, Yi Cui, *Hourglass Content and Service Composition Framework for Pervasive Environments*, Elsevier Pervasive and Mobile Computing, 2005.
18. C.-Y. Chang and M.-S. Chen, *On Exploring Aggregate Effect for Efficient Cache Replacement in Transcoding Proxies*, IEEE Trans. on Parallel and Dist. Sys., 14(7),2003.
19. G. Eisenhauer, K. Schwan, F. E. Bustamante,*Publish-Subscribe for High-Performance Computing.* IEEE Internet Computing 10(1): 40-47 (2006)
20. U. Srivastava, K. Munagala and J. Widom, *Operator placement for in-network stream query processing*, ACM PODS 2005.
21. Y. Zhu and M. Ammar, *Algorithms for assigning substrate network resources to virtual network components*, IEEE INFOCOM 2006.