CCD: A Distributed Publish/Subscribe Framework for Rich Content Formats

Hojjat Jafarpour, Bijit Hore, Sharad Mehrotra, and Nalini Venkatasubramanian

Abstract—In this paper, we propose a content-based publish/subscribe (pub/sub) framework that delivers matching content to subscribers in their desired format. Such a framework enables the pub/sub system to accommodate richer content formats including multimedia publications with image and video content. In our proposed framework, users (consumers) in addition to specifying their information needs (subscription queries), also specify their profile which includes the information about their receiving context which includes characteristics of the device used to receive the content (e.g., resolution of a PDA used by a consumer). The pub/sub system besides being responsible for matching and routing the published content, also becomes responsible for converting the content into the suitable format for each user. Content conversion is achieved through a set of content adaptation operators (e.g., image transcoder, document translator, etc.). We study algorithms for placement of such operators in heterogeneous pub/sub broker overlay in order to minimize the communication and computation resource consumption. Our experimental results show that careful placement of operators in pub/sub overlay network results in significant cost reduction.

Index Terms—Publish/subscribe, operator placement, customized 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], [5], [2]. 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 a produced 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 present our work on customized content dissemination (CCD) [3] and extend it to address the effect of heterogeneity and concurrent publications in the system. 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[1] where users can publish video content that must be delivered to their followers (subscribers). Followers may subscribe to such 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 Fig. 1 where node A publishes a high-resolution

- *H. Jafarpour is with the NEC Laboratories America, Inc., 10080 North Wolfe Road, Suite SW3-350, Cupertino, CA 95014-2515. E-mail: hojjat@sv.nec-labs.com.*
- *B. Hore, S. Mehrotra, and N. Venkatasubramanian are with the Department of Computer Science, Donald Bren School of Information and Computer Sciences, University of California, Irvine, Irvine, CA 92697-3435. E-mail: {bhore, sharad, nalini}@ics.uci.edu.*

Manuscript received 4 Jan. 2010; revised 11 May 2011; accepted 31 May 2011; published online 21 July 2011.
Recommended for acceptance by Y. Hu.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-2010-01-0013.
Digital Object Identifier no. 10.1109/TPDS.2011.212.

1. http://twitter.com/.

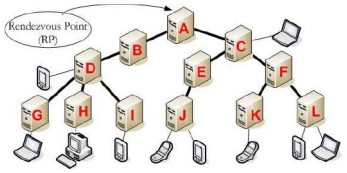

Fig. 1. Sample dissemination tree.

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 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. We extend the proposed algorithms to account for heterogeneity resulting from the broker network and concurrent publications taking place in the system. 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 problem in a distributed pub/sub systems (Section 2). We also show that optimal CCD problem is NP-hard.
- 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 3). We also extend our work in [3] by an alternative dissemination refinement algorithm.
- For cases where the number of requested formats is large, we propose a greedy heuristics-based algorithm (Section 4).
- We describe how to factor in the heterogeneity of network and brokers and account for concurrent publications (Section 5). This is a major extension to our work in [3].
- 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

In this section, after a brief introduction of the pub/sub framework that we use in our approach, we introduce our system model for customized content dissemination in pub/sub framework and present the CCD problem formally.

DHT-based pub/sub. Our customized content dissemination system architecture is based on a distributed DHT-based pub/sub. It consists of a set of stable nodes as content brokers that are connected through a structured overlay network. Each client connects to one of the brokers and communicates through which it communicates with the system. In DHT-based pub/sub, content space is partitioned among the set of brokers. Each broker maintains subscriptions for its partition of content space and is responsible for matching them against the publications belonging to the same partition. In fact, each broker is the *Rendezvous Point (RP)* for the publications and subscriptions in its partition. A broker forwards all subscriptions from his own clients to the brokers (RP) responsible for the corresponding content partitions. Similarly, when a broker receives a published content from its client, it forwards the content to the appropriate RP. The content is matched with the list of 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. Note that the RP may not be able to directly communicate with the matched brokers since these brokers may not be direct neighbors of RP in the DHT-based routing scheme. Finally, each of the brokers with matching subscription, after receiving the content deliver it to the appropriate subscribers (clients). Note that although clients can subscribe and unsubscribe dynamically, the effect of these actions will be reflected in publication routing only after the subscriptions and unsubscriptions reach their corresponding PR node. Since a broker acts as a proxy for all clients that connect to it, in this paper we consider the broker itself as the subscriber or publisher in lieu of its clients. Hence, we can simply concentrate on the broker overlay network. Various DHT-based routing techniques have been proposed in the

literature [7], [6] that can be used for routing content from RP to the matching brokers. In this paper, we use the *Tapestry* routing scheme [6]; 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 in Tapestry. For more details on dissemination tree construction, we refer the interested reader to [8]. DHT-based pub/sub on Tapestry suits our customized content dissemination system for two important reasons.[2] 1) In DHT-based pub/sub, 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. As we will see this knowledge is essential for our proposed system. 2) 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 in the RP broker and sent to all subscribing brokers. The leaf brokers in the dissemination tree then resend the message to the RP broker. Each tree discovery message keeps information about the route from the RP broker to the leaf broker in the tree and RP broker uses such information to construct the exact dissemination tree for the given publication. In this paper, we use tree discovery message for constructing dissemination tree for publications. Note that by using DHT-based pub/sub, all the properties of such systems including robust content routing in presence of failures and resilience to churn in the broker network will be available for our system [10], [9]. Fig. 1 depicts a sample dissemination tree.

2.1 Content Adaptation Graph

We assume every client has a profile describing the context for the subscribed content such as, device characteristic (e.g., screen size and resolution) and connection characteristics (e.g., connection type and bandwidth). The client profiles are registered at their brokers and is used to determine the format(s) in which content needs to be delivered. Each subscription is forwarded with its required profile to the appropriate RPs who then use this information for optimal routing computation.

When published content is forwarded to an RP, the content needs to be customized according to the profiles of the matching 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 (We will consider the general case where brokers and links are not identical in Section 5.). Now, if the set of all required formats for content $\mathbb{C}$ in format F_i is $F = \{F_1, \ldots, F_m\}$, we can associate a transmission-cost $\mathcal{T}_{F_i}(\mathbb{C})$ per link. $O_{(i,j)}$ represents the operator that converts content format from F_i to F_j. Associated with each $O_{(i,j)}$ is a conversion cost $\mathcal{C}_{O_{(i,j)}}(\mathbb{C})$ that represents the computation cost of performing this operator at any broker.[3] Note that it may not always be feasible to convert content from any given format F_i into another format F_j or the system may not support particular conversions. For example, it might be impossible to convert a low-resolution image to a higher resolution one; converting a video in "avi" format into "flv" might not be supported, etc. In such cases, we assume $O_{(i,j)}$ to be undefined.

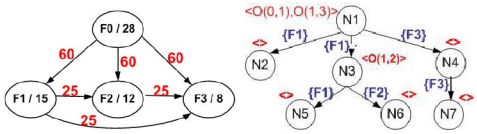

Fig. 2. A sample CAG and dissemination plan.

We use a directed, weighted graph to represent the various formats and their relationships. We will call this the *Content Adaptation Graph*. The vertices in CAG represent the content formats and the directed edges represent the operators that convert content from the source format to the sink format. The associated cost is denoted by the edge weight. Similarly, a weight associated with each node of the CAG represents the per-unit transmission cost for content in that format. Fig. 2 illustrates a CAG involving four formats of an "mpeg4" video content with different frame sizes and bit rates. Content size is denoted in Megabytes (MBs) and the conversion cost is measured in seconds taken to convert 1 MB of source data.

2.2 Cost-Based Customized Dissemination

Consider the problem of customized dissemination of content $\mathbb{C}$ in format F_0 from RP broker to a set of brokers $R = \{R_1, .., R_r\}$. 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. Note that $R \subseteq N$.

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 links, respectively. Fig. 2 shows a sample plan where content is delivered in format F_1 to brokers N_2 and N_5, in format F_2 to N_6 and in format F_3 to N_7. A subtree in the customization plan is called a *subplan*.

In every customization plan, we assume that the content to be disseminated is available at the root node (RP) of the dissemination tree in its original published format.

Cost model. We associate two cost measures with a customization plan: *Conversion cost* and *Transmission cost*. 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 [19], [17] for in-network stream processing and cache replacement. We

2. Note that our proposed system can be implemented on top of any distributed pub/sub setting where a dissemination tree is used to deliver published content to subscribers and the set of receivers and their requested content formats is available in the root of dissemination tree.

3. In general we will assume these costs to represent the per-unit costs.

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:

$\Theta_{\mathbb{P}}(\mathbb{C}) = \alpha\tau_{\mathbb{P}} + \beta\varphi_{\mathbb{P}}$, where $\varphi_{\mathbb{P}}$ and $\tau_{\mathbb{P}}$ 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 Fig. 2 (in the appendix, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPDS.2011.212) 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 is the following:

Customized Content Dissemination Problem. For a customized dissemination task find the valid customization plan with minimum total cost.

Theorem 1. *CCD problem is NP-hard.*

Proof. Given in Appendix, available in the online supplemental material. □

3 Optimal CCD Algorithm

An interesting observation is that CCD problem can be formulated as a minimum directed Steiner tree problem for a *multilayer graph* constructed from the given CAG and dissemination tree. In fact this observation was made in [12] for multicasting problem. A multilayer graph for CCD problem is constructed by combining the dissemination tree and the content adaptation graph into a single multilayered graph where all contents within a single layer are in the same format and weight of edges between different levels denote conversion costs. The CCD problem can be posed as a minimum weight directed Steiner tree problem on this graph. We present the details in appendix, available in the online supplemental material.

However, when the CAG has a small number of formats (less than 5), one can use a brute force algorithm that generates the optimal operator placement plan for the CCD problem. For instance, in an image dissemination system we may be able to categorize the devices into a small set of classes, e.g., "PC with high speed connection," "PC with dial-up connection," "Mobile device with Wi-Fi connection," and "Mobile device with GSM connection." In such cases, an optimum solution is possible. An important advantage of this algorithm compared to the general solution based on multilayer graph structure is that it finds the *minimum* cost CCD plan. It also has a linear complexity with respect to the dissemination tree size and can be used for efficiently computing the optimal plan for large dissemination trees when the CAG is small.

We provide a dynamic programming algorithm to find the optimal solution to the CCD problem and present a detailed example in appendix, available in the online supplemental material. The complexity of the proposed algorithm is given by the theorem below.

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. Given in the appendix, available in the online supplemental material. □

3.1 CCD Plan Refinement

As mentioned, the first step in the computing a CCD plan for a given publication is to discover the dissemination tree rooted at the corresponding RP broker. This can be done using a tree discovery message as described in Section 2. Another alternative is to use the estimated dissemination tree in the root. In the this approach, the rendezvous point for a publication generates an estimated dissemination tree based on its Tapestry routing table. It then computes a minimum cost CCD plan for the estimated tree using the optimal CCD algorithm. Based on this plan, RP submits to each of its child nodes N_j content in a set of formats (as determined by the computed CCD plan) along with a subplan rooted at N_j of the minimum cost plan determined by RP. A child node N_j, in turn, computes an estimated dissemination tree for the nodes that it is responsible to route the content to (note that such information is available in the subplan associated with N_j). If the dissemination subtree estimated by N_j diverges from the dissemination subtree used in the subplan, node N_j determines the benefit of refining the subplan versus following the less accurate plan received from the parent. A node can refine the subplan by calling the optimal CCD algorithm based on the input format in which it received content from its parent and the formats in which it is responsible to deliver content to the nodes it is responsible for. The steps that a broker in the dissemination tree must follow to refine the dissemination plan is represented in Algorithm 1.

Algorithm 1. CCD Plan refinement at broker N_i.

1: $B \leftarrow$ The benefit threshold for recomputing an inaccurate plan.
2: **INPUT:**
3: $F_{in}^{N_i} \leftarrow$ Set of formats in which the content is received.
4: $P_i \leftarrow$ The customization subplan received from the parent.
5:
6: $R \leftarrow P_i$**.getReceivers()** {Subscribers and their requested formats.}
7: $D_i \leftarrow$ **computeRefinedTree(**R**)**
8: **if** $D_i \neq P_i$.getEstimatedTree() **then**
9: **if Benefit**$(P_i, D_i) >$ **B then**
10: $P_i \leftarrow$ **Compute CCD Plan for(**$F_{in}^{N_i}$**,**R**) using optimal CCD algorithm**
11: **end if**
12: **end if**
13: **PerformOperators(**N_i, P_i**)**
14: **for all** $N_j \in N_i$.getChildrenList() **do**

15: $P_j \leftarrow P_i.getSubplan(N_j)$
16: $F_{in}^{N_j} \leftarrow P_i.getFormats(N_j)$
17: Forward P_j and $F_{in}^{N_j}$ to N_j.
18: **end for**

4 CCD Problem for Large CAGs

In this section, we present a brief overview of our heuristic algorithm that tackles the case of large CAGs. We present details of the algorithm in the appendix, available in the online supplemental material. The algorithm considers an initial CCD plan as the input. It then iteratively selects a node in the dissemination tree and refines the subplan corresponding to the selected node and its children to reduce the cost of the whole dissemination plan. The refining process may include the following two actions: 1) changing the conversion operators at this node and its children; 2) 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).

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: $N_i = \text{SelectNode}(\mathbb{P})$
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 incremental change in cost in successive iterations, number of iterations, time bounded, etc. In this paper, we just iterate for a fixed number of times, $\mathcal{K}$ which is provided by the user.

5 Heterogeneity and Concurrent Publications

Up till now we have implicitly assumed homogeneity of the overlay network. That is, all the nodes and links in the system have similar characteristics. Broker nodes and the communication networks that connect them are certainly not homogeneous in the real world. Different broker nodes may have widely different computational resources and connectivity between different brokers may differ significantly in terms of available bandwidth. Such heterogeneity may be an outcome of physical differences between brokers and communication networks in which they reside, or the result of the current load in the system as a result of concurrent publications. Heterogeneity may have considerable impact on customized dissemination. For instance, performing customization operators on nodes with limited computational resources may be deemed more expensive as compared to performing them in nodes with larger computational resources. Likewise, transmitting content over a high bandwidth link might be considered cheaper compared to transmission over links with limited bandwidth. To be able to adapt our customized dissemination approach to heterogeneous networks, we first need to estimate cost of performing operations on diverse nodes and for transmitting content over diverse networks. To account for resource heterogeneity, we adopt a model similart to the ones proposed in [16], [13] that assigns a cost for a resource based on the resource usage time. In particular, let the cost of using node N_i for one time unit be denoted by C_{N_i} which can be defined as $C_{N_i} = \gamma t_J$, where t_J is the execution time of a (benchmark) job J on N_i and γ is a constant. Note that C_{N_i} is higher for nodes with less-computational resources compared to that for nodes with larger computing power. Similarly, we define the cost of using a link $<N_i, N_j>$ (denoted by $C_{<N_i,N_j>}$) as $C_{<N_i,N_j>} = \epsilon t_T$ where ϵ is a constant and t_T is the transmission time for a fixed size content (say, 1 KB). Again, the cost of transmitting over a high-bandwidth link would be cheaper compared to transmitting over links that have low bandwidth. Given the above cost models for broker nodes and links, we can compute the cost of executing the customization operator $O_{(i,j)}$ at node N_i as $CC_{O_{(i,j)}} \times C_{N_i}$. Similarly, the cost of transmitting content in F_i over link $<N_i, N_j>$ can be computed as $TC_{F_i} \times C_{<N_i,N_j>}$.

The customized dissemination problem over heterogeneous networks can be restated as that of identifying the least cost plan based on the cost functions that account for network and node heterogeneity as discussed above. We note that we can easily extend our optimal and heuristic CCD algorithms to address customized dissemination in heterogeneous networks. In particular, the *tree discovery message* that introduced in Section 2, in addition to gathering information about the structure of the dissemination tree can be modified to also collect information about the node's computational resources and link bandwidth based on which dissemination plan that accounts for heterogeneity in the network can be determined. The details are straightforward and we omit them for brevity. We will explore the impact of considering network heterogeneity in the experimental section.

Based on the proposed solution for accounting for heterogeneity, we can also account for the effect of concurrent publications in computing dissemination plans. This is done by continuously updating the broker and link costs in the system based on their available resources. Each broker keeps estimates of its current computation load and its links communication load and updates its cost and its links' cost according to the resources allocated to the existing dissemination plans that are running in the system. Therefore, when a new content is published the system considers the effect of the currently running plans in computing the new plan by using the updated broker and link costs. Assuming the current cost of broker N_i is C_{N_i} and the maximum computation load that can be accepted at the broker is $\mathbb{C}_{N_i}^{max}$ and the current load on the broker is $\mathbb{C}_{N_i}^{current}$, we use the following formula to update the broker cost:

$$C_{N_i} = \begin{cases} \mathrm{C}_{N_i} \times \frac{\mathbb{C}_{N_i}^{max}}{\mathbb{C}_{N_i}^{max} - \mathbb{C}_{N_i}^{current}}, & \text{if } \mathrm{C}_{N_i}^{max} > \mathbb{C}_{N_i}^{current}, \\ \mathrm{C}_{N_i}^{Max}, & \text{if } \mathrm{C}_{N_i} > {}^{Max}_{N_i}. \end{cases}$$

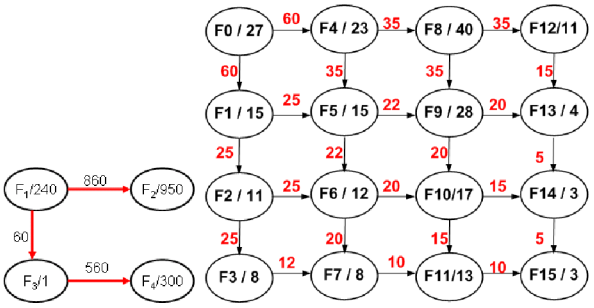

Fig. 3. Annotated map and video dissemination CAGs.

In the above formula $C_{N_i}^{Max}$ is the maximum cost that can be assigned to a broker. As long as the updated cost is less than the maximum cost we used the computed value to update the broker cost. Otherwise, the broker has consumed all the resources and we assign the maximum cost to it. The link cost is updated in a similar way.

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 1,024 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 presented results are average values 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 our algorithms was 100 ms and we assumed probe message size is 0.1 KB. Publishers and subscribers in the broker overlay are selected randomly. Similarly, the requested formats by a subscriber are selected from available format set using uniform distribution. Each broker has subscriptions for at most $\frac{1}{4}$ of available formats in the CAG. The default value for α and β is set to one in the cost function meaning that the communication and computation normalized cost units are the same.

6.2 Dissemination Scenarios

An important factor in customized dissemination is the constructed CAG for the published content. 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 our proposed heuristic CCD algorithm.

Annotated Map Dissemination. For the first scenario, we considered customized dissemination of annotated maps in emergency management context. In this scenario, annotated maps are disseminated among subscribers. For instance, in case of wild fire an annotated map depicting shelters for evacuees in a specific region is disseminated.

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.

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 effect of the concurrent publications on the heuristic CCD algorithm.

We use the small CAG from the annotated map scenario in the first experiment to evaluate the benefit of using optimal CCD algorithms. In the rest of experiments 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 in cost reduction. 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). Fig. 4 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 results 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 against AIL approach when $\frac{\alpha}{\beta} = 0.1$, while when $\frac{\alpha}{\beta} = 10$ the saving in cost compared to AIR is higher. 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 it is expected the difference between CCD plans and AIR is not very significant because the computation cost is minimum 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 and AIR results in

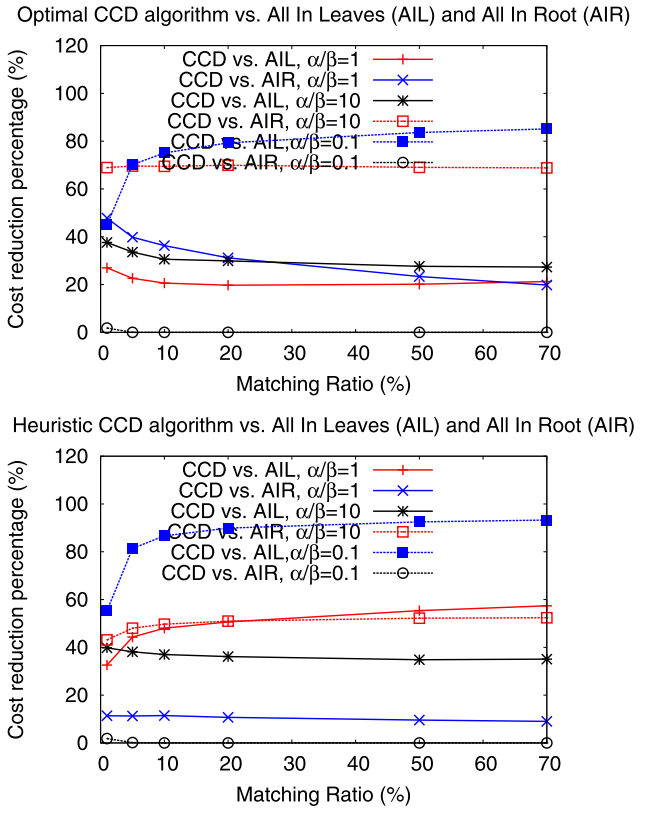

Fig. 4. Cost reduction percentage in Optimal and Heuristic CCD algorithms compared to AIL and AIR for different α and β values.

higher communication cost because of transmission of 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 AIL and AIR approaches.

Concurrent publications. In the last set of experiments, we plot the benefit of considering concurrent publications for computing dissemination plans in Fig. 5. The graph plots the percentage of cost reduction for dissemination plans when 100 publications are happening in a single time unit. The plotted values are the average benefit for every five publications. After the first five publications, considering available resources in brokers and links results in around 80 to 90 percent reduction in the dissemination plan cost when matching ratio is 50 and 70 percent. Similarly the graph shows significant reduction in plan costs for 5, 10, and 20 percent matching ratios. These results validate the proposed technique for heterogeneity and concurrent publications. Note that the significant benefit in plan cost for considering heterogeneity and concurrent publications is for fifth through 30th publications for 50 and 70 percent matching ratios while it occurs for publications 10 through 70 for matching ratio 20 percent and after 30th publications for 10 and 5 percent matching ratios. This is justifiable because in higher matching ratios larger number of brokers and links in the broker network are involved in the dissemination plan and therefore more resources are being consumed by the initial publications and the other publications need to take these resource consumptions into account. However, for smaller matching ratios smaller

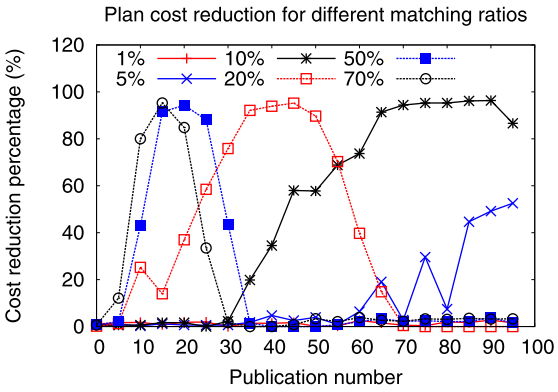

Fig. 5. Plan cost reduction percentage in presence of concurrent publications.

portion of brokers and links are affected by initial publications so the later publications will have more resources available in the broker overlay network.

Another important fact that is shown in Fig. 5 is that the benefit of considering heterogeneity and concurrent publications vanishes after certain number of publications. As shown in the graph, after the 30th publication benefit becomes almost zero for 70 percent matching ratio. The reason is that after having 30 publications with 70 percent matching ratio, the resources in all brokers are consumed and for the subsequent publications the brokers are saturated. For the smaller matching ratios resource saturation happens after having larger number of publications because each publication consumes lesser amount of communication and computation resources.

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], [2], [5]. Shah et al. studied filter placement in content-based pub/sub network [14]. 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 [15]. 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.

Some multimedia content dissemination systems expand the multicasting concept by providing content customization service for group members. In [13], Lambrecht et al. formally defined 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 [12], 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 multicast group is a fixed and predefined group.

Content customization has been subject to extensive research in multimedia community. Nahrstedt et al. proposed *Hourglass* [16], a multimedia content customization and dissemination framework. 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.

8 Conclusions and Future Work

We 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 DHT-based pub/sub framework 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 proposed approaches to use estimated dissemination tree and take into account broker overlay heterogeneity and concurrent publications effect.

In our heuristic CCD algorithm we used multilayer graph for a subtree of depth one in the dissemination tree [3]. 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.

References

[1] S. Castelli, P. Costa, and G.P. Picco, "HyperCBR: Large-Scale Content-Based Routing in a Multidimensional Space," *Proc. IEEE INFOCOM,* 2008.

[2] I. Aekaterinidis and P. Triantafillou, "PastryStrings: A Comprehensive Content-Based Publish/Subscribe DHT Network," *Proc. IEEE Int'l Conf. Distributed Computing Systems (ICDCS),* 2006.

[3] H. Jafarpour, B. Hore, S. Mehrotra, and N. Venkatasubramanian, "CCD: Efficient Customized Content Dissemination in Distributed Publish/Subscribe," *Proc. ACM/IFIP/USENIX Int'l Conf. Middleware,* 2009.

[4] F. Cao and J. Pal Singh, "MEDYM: Match-Early with Dynamic Multicast for Content-Based Publish-Subscribe Networks," *Proc. ACM/Usenix/IFIP Int'l Conf. Middleware,* 2005.

[5] G. Li, V. Muthusamy, and H.A. Jacobsen, "Adaptive Content-Based Routing in General Overlay Topologies," *Proc. ACM/Usenix/IFIP Int'l Conf. Middleware,* 2008.

[6] 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 J. Selected Areas in Comm.,* vol. 22, no. 1, pp. 41-53, Jan. 2004.

[7] A. Rowstron and P. Druschel, "Pastry: Scalable, Distributed Object Location and Routing for Large-Scale Peer-to-Peer Systems," *Middleware: Proc. ACM/Usenix/IFIP Int'l Conf. Distributed Systems,* 2001.

[8] S.Q. Zhuang, B.Y. Zhao, A.D. Joseph, R.H. Katz, and J. Kubiatowicz, "Bayeux: An Architecture for Scalable and Fault-Tolerant Wide-Area Data Dissemination," *Proc. 11th Int'l Workshop Network and Operating Systems Support for Digital Audio and Video (NOSSDAV),* 2001.

[9] R. Baldoni, C. Marchetti, A. Virgillito, and R. Vitenberg, "Content-Based Publish-Subscribe over Structured Overlay Networks," *Proc. IEEE Int'l Conf. Distributed Computing Systems (ICDCS),* 2005.

[10] A. Gupta, O. Sahin, D. Agrawal, and A. El Abbadi, "Meghdoot: Content-Based Publish/Subscribe over P2P Networks," *Proc. ACM/Usenix/IFIP Fifth Int'l Middleware Conf.,* 2004.

[11] M. Charikar, C. Chekuri, T. Cheung, Z. Dai, A. Goel, S. Guha, and M. Li, "Approximation Algorithms for Directed Steiner Problems," *Proc. ACM-SIAM Symp. Discrete Algorithms,* 1998.

[12] A. Henig and D. Raz, "Efficient Management of Transcoding and Multicasting Multimedia Streams," *Proc. Ninth IFIP/IEEE Int'l Symp. Integrated Network Management,* 2005.

[13] T. Lambrecht, B. Duysburgh, T. Wauters, F. De TurckBart Dhoedt, and P. Demeester, "Optimizing Multimedia Transcoding Multicast Trees," *Computer Networks,* vol. 50, no. 1, pp. 29-45, Jan. 2006.

[14] R. Shah, Z. Ramzan, R. Jain, R. Dendukuri, and F. Anjum, "Efficient Dissemination of Personalized Information Using Content-Based Multicast," *IEEE Trans. Mobile Computing,* vol. 3, no. 4, pp. 394-408, Oct.-Dec. 2004.

[15] Y. Diao, S. Rizvi, and M.J. Franklin, "Towards an Internet-Scale XML Dissemination Service," *Proc. Very Large Databases (VLDB) Conf.,* Aug. 2004.

[16] K. Nahrstedt, B. Yu, J. Liang, and Y. Cui, "Hourglass Content and Service Composition Framework for Pervasive Environments," *Pervasive and Mobile Computing,* Elsevier, 2005.

[17] C.-Y. Chang and M.-S. Chen, "On Exploring Aggregate Effect for Efficient Cache Replacement in Transcoding Proxies," *IEEE Trans. Parallel and Distributed Systems,* vol. 14, no. 7, pp. 611-624, June 2003.

[18] G. Eisenhauer, K. Schwan, and F.E. Bustamante, "Publish-Subscribe for High-Performance Computing," *IEEE Internet Computing,* vol. 10, no. 1, pp. 40-47, Jan./Feb. 2006.

[19] U. Srivastava, K. Munagala, and J. Widom, "Operator Placement for In-Network Stream Query Processing," *Proc. 24th ACM SIGMOD-SIGACT-SIGART Symp. Principles of Database Systems (PODS),* 2005.

[20] Y. Zhu and M. Ammar, "Algorithms for Assigning Substrate Network Resources to Virtual Network Components," *Proc. IEEE INFOCOM,* 2006.

Hojjat Jafarpour received the PhD degree in computer science from the University of California, Irvine. He is currently a research staff member in Data Management Department at NEC Labs America. His research interests include scalable data management, distributed middleware systems, and cloud computing.

Bijit Hore received the MS and PhD degrees in computer science from the University of California Irvine in 2003 and 2007, respectively, and the MSc degree in applied statistics and informatics from the Indian Institute of Technology (IIT) Bombay. He is currently a postdoctoral researcher in the Information Systems Group at UCI. His research interests include security and privacy issues in data outsourcing and sharing applications, secure data management in the cloud, query optimization, publish/subscribe systems, graph theory, and algorithms. His current focus is on secure query processing in hybrid clouds and other mixed security environments. He has been on the program committee of WWW 2011 poster track, IEEE ISI 2010 and 2011 conferences. He has also been an invited reviewer for reputed journals like *VLDBJ*, *ACM TWEB*, *ACM TISSEC*, *IEEE TKDE*, etc. He has also been an external reviewer of numerous conference papers in the database area.

Sharad Mehrotra is a professor in the School of Information and Computer Science at the University of California, Irvine and the director of the Center for Emergency Response Technologies (CERT) at UCI. He has served as the director and PI of the RESCUE project (Responding to Crisis and Unexpected Events) funded by the US National Science Foundation (NSF) through its large ITR program. He is a recipient of numerous research and teaching awards including Outstanding Graduate Student Mentor Award, at UCI, C.W. Gear Outstanding Junior Faculty Award at UIUC, and numerous best paper awards including SIGMOD Best Paper award in 2001. His current research focuses on building sentient spaces using multimodal sensors, data privacy, and data quality.

Nalini Venkatasubramanian received the PhD degree in computer science from the University of Illinois at Urbana-Champaign in 1998. She is a professor of Computer Science at University of California, Irvine. Her research interests are in Distributed Systems Middleware, Multimedia Systems and Applications, Mobile and Ubiquitous Computing, Formal Methods, Data Management, Grid Computing. She has served as program committee of reputed journals and conferences including ICDCS, Middleware, ACM Multimedia.