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

APPENDIX A

NP-hardness

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 \( n \) brokers too. To prove the theorem it is enough to 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 tree rooted at \( r \), such that all vertices in \( S \) are included in the tree [11]. It is easy to see that any instance of the directed Steiner tree problem is equivalent to the degenerate CCD problem where \( G = CAG \), the vertices’ in \( S \) correspond to the set of formats 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. □

APPENDIX B

Multilayer Graph representation of CCD

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 (CAG) as follows:

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 \( G_{ML} = (V, E) \) such that \( V = V_d \times V_c \) where \( V_c \) denotes the set of vertices in the CAG and \( V_d \) denotes the set of nodes in the dissemination tree. Each vertex in \( V \) is therefore associated with exactly one pair of nodes, where the first member 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 \( G_{ML} \) 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 between 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. 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 $G_{ML}$. 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 in the minimum cost Steiner tree. Also, for each operator $O(s,w)$ assigned to node $N_i$ in the plan, the conversion edge between corresponding nodes for $N_i$ between the layers associated with $F_s$ and $F_w$ exists 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.

As an example consider the CAG and dissemination tree depicted in figure 1. Figure 2 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.

Since finding the minimum cost directed Steiner tree is an NP-hard problem we can use any one of the approximate algorithms in the literature [11] to compute a good CCD plan. However, even for best approximation algorithm the time complexity is very high which makes them unsuitable for large graphs. For instance, Charikar et al., proposed a $(i(i − 1)k^2)$-approximate algorithm with time complexity of $O(v^2 k^2)$ where $v$ is the number of vertices in the directed graph and $k$ is the number of terminals for the Steiner tree [11]. To better understand the complexity of such algorithm assume there are 1000 brokers in the system and 70% of them have subscribed to half of the formats in the CAG that consists of 10 formats. Considering $i = 2$ the complexity of achieving a 83-approximate solution is becomes very high (There are 10,000 nodes in the multilayer graph and 3,500 terminals). As it can be seen when the multilayer graph size increases the complexity of computing minimum Steiner tree significantly increases and the resulting approximation also becomes less accurate(83-approximate).

Therefore, in the next two sections we present two CCD algorithm that compute the CCD plan with lower cost. Our first algorithm is designed to find the optimal CCD plan when the number of formats in CAG is small (less than 5). Our second algorithm is an iterative heuristic that computes the CCD plan when the number of formats in the CAG is large (more than 5).

**APPENDIX C**

**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). 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.

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 $F_i^{N_i}$ from its parent (as shown in figure 4). 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). We will describe how these minimum-cost dissemination plans for child nodes are achieved. 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 3. 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 \( \{ F_2 \} \) be \( \{ \{ F_2 \} \} \). 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 \( \mathbb{F}_j^* \) 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 \( S(\{ F_2 \} \to f_i^*) \)). 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.

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 (\( \text{ChildSubPlans}_{N_i} \)). As mentioned, the algorithm assumes that the minimum cost plan for all input format sets are available for every child of \( N_i \). This is achieved by recursively computing the minimum-cost plan for each node. The minimum-cost plan for a leaf node in the tree for a given set of input formats is the cost of producing the requested formats in the leaf node from the input formats. For the intermediate nodes in the tree the minimum cost we calculate the minimum cost plan for each possible input set for all child nodes and then use the algorithm to compute the minimum cost plan for the given input formats. We then initialize the empty plan \( P \) 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 [11] 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.

**Algorithm 1 OptimalCCD**

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

To find the minimum cost plan for a given dissemination tree, we call the `OptimalCCD` algorithm in the RP broker with \( F_{in}^{RP} = \{ F_1 \} \). 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 \( O(nk_{avg}2^{3mS}) \), 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 \( S \) 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 \( S \) (which is assumed to be almost linear due to the small value of \( m \)), then from the analysis done in the example above (figure 4), we can show that the worst case complexity of Algorithm 1 is \( O(nk_{avg}2^{3mS}) \).

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.

**APPENDIX D**

**CCD Problem for Large CAGs**

In this section we tackle the case of large CAGs. We provide a heuristic based iterative algorithm which uses greedy heuristics based on some key properties of the CCD problem. We show through extensive experimentation that these heuristics work very well in practice. In fact, we empirically showed 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.

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).

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, \( K \) which is provided by the user. In the remainder of this section, we present details about the initialization, node selection and plan refinement steps of our iterative algorithm.

**D.0.1 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 strategies described below.

We refer to the first two initial plans as *All-in-root* and *All-in-leaves* plans. Both of these algorithms avoid in network placement of customization operators and perform all the required operators either at the dissemination tree root or at the leaves. The *All-in-root* CCD algorithm generate all the required formats in the dissemination tree by performing all the necessary operators in the root (RP). Then the generated content 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 converts the content into its requested formats by performing the necessary customization operators. We refer to the third initial plan as *Single-format*. In this plan only one format is transmitted over each link in the dissemination tree where the format has the smallest possible transmission cost.

**D.0.2 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 above 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 applying refinement. Specifically, we propose a greedy heuristic for node selection based on comparing the current cost of a sub-plan with an estimated lower bound to the minimum achievable cost for the sub-plan. In this heuristic, `SelectNode` returns the node with the highest potential cost reduction 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

**Algorithm 2 Iterative CCD algorithm for large CAG**

```plaintext
1: INPUT: P: The initial plan, K: Number of iterations
2: OUTPUT: P: The refined plan
3: 4: for all j = 0 to K do
5:     5: N_i = SelectNode(P)
6:     6: RefinePlan(P, N_i)
7: end for
8: return P;
```
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 \( N_i \)). Thus, the heuristic bases the node selection on the projected cost reduction using our lower bound estimates. We refer to this as the SLACK scheme. Since our cost model consists of content transmission and content conversion costs, to find a lower bound for a plan we first find 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 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 \( \{ F_{k1}^{min}, ..., F_{km}^{min} \} \), 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 tree as well. The set of formats requested at any internal node \( N_i \) is simply taken to be the union of formats requested at any client of a node in the subtree rooted at \( N_i \). 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 \( O(m \log(m)) \) 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 \( T_{low}(t) \) for a subtree \( t \) or simply by \( T_{low} \) 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 \( C_{low} \). 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.

D.0.3 Step 3: Plan Refinement using Multilayer Graph

The \texttt{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 \texttt{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 [11] 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 reflect the new operations and transmissions in the dissemination tree as described below (lines 15-17).

**Algorithm 3 RefinePlan.**

1: INPUT: \( P \): The initial plan, \( N_i \): Selected node ;
2: 3: \( \mathcal{G}_{ML}(V, E) = \text{createMLGraph}(N_i) \);
4: Source \( \leftarrow \phi ; // \text{Set of source vertices}; \)
5: Terminal \( \leftarrow \phi ; // \text{Set of terminal vertices}; \)
6: for every \( v \in V \) do
7: if \( v.node = N_i \text{ AND } v.format \in F_{in}^{N_i} \) then
8: Source \( \leftarrow \text{Source} \cup \{v\} ; \)
9: end if
10: if \( N_j \in \text{Children}(N_i) \text{ AND } v.node = N_j \text{ AND } v.format \in F_{out}^{N_j} \) then
11: Terminal \( \leftarrow \text{Terminal} \cup \{v\} ; \)
12: end if
13: end for
14: SteinerTree \( \leftarrow \text{MinSteiner}(\mathcal{G}_{ML}, \text{Source}, \text{Terminal}); \)
15: if SteinerTree.cost < SubPlanCost(\( P, N_i \)) then
16: Update(\( P \));
17: end if
The Update 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 multilayer graph for a node and its children only, the size of the graph is significantly smaller than the multilayer graph for all 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 $K$ iteration is $O(160^2 \times 150^4 \times K)$ which is significantly less than $O(10000^2 \times 3500^4)$ which was the complexity of the example in the paper.

**APPENDIX E**

**ADDITIONAL EXPERIMENTAL RESULTS**

**E.1 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. 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 5 depicts the corresponding CAG where the costs were computed based on our extensive prototyping.

**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 library1. 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 6.

**E.2 Experiments**

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

1. For information on FFmpeg please refer to “http://www.ffmpeg.org/”.

![Fig. 5. Sample content and CAG for Annotated Map scenario.](image)

![Fig. 6. Video dissemination CAG with subset of edges.](image)
- The effect of using tree discovery vs. tree estimation and CCD plan refinement.
- The quality of the heuristic CCD algorithm results.
- The effect of different parameters in the heuristic CCD algorithm.
- The effect of the relationship between communication and computation costs on the heuristic CCD algorithm.
- The effect of broker and link heterogeneity on the heuristic CCD 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 CCD algorithm. 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.

![Figure 7: Goodness of the heuristic CCD algorithm compared to the optimal algorithm.](image)

**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 7 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.

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 plan, 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, 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 8 plots the costs of three initial dissemination plans for different matching ratios in three different scenarios. As it is seen when the communication resources have more importance in the system (\( \alpha = 0.1 \)), 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 is the worst initial plan when \( \alpha = 10 \). 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, heuristic CCD algorithm compute all possible initial CCD plans and select the one with the smallest cost as the initial plan for refining the plan using iterations.

![Figure 8: Initial plan comparison for different \( \alpha \) and \( \beta \) values.](image)

**Next step selection:** In this experiment we evaluate the random and slack based selection techniques. Figure 9 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 enough number of iterations. This means if there are enough resources available for large number of iterations, both techniques achieve the same final refined dissemination plan.

**Heterogeneity:** In this set of experiments we evaluate the effect of network heterogeneity in the dissemination plan cost. We consider two heterogeneity model in this experiment. In the first, we associated a link and node cost model motivated by the PlanetLab statistics published in [20]. In particular, we simulated 60% of links to have a bandwidth between $[10, 100]$ Mbs and 40% between $[0, 10]$ Mbs. The bandwidth was assumed to be uniformly distributed within those bands. We considered the 10 Mbs network as the baseline and thus associated the transmission cost of $10/x$ with the link associated with $x$ Mbs. Likewise, to model node heterogeneity, we set CPU availability in the range of $[80, 100]$, (uniformly distributed in that range.) We took 100% availability as the baseline and the node cost of a $x/2$ available CPU was modeled as $100/x$. Thus the range of node cost varied from $[1, 1.25]$. Since this model does not provide significant difference between broker costs we refer to this model as weak heterogeneity. In the second model we assign broker and link cost by random values in $[1,100]$ which provides considerable difference between broker and link costs. We refer to this model as strong heterogeneity.

Figure 10 represent the cost reduction percentage for different matching ratios when plans take heterogeneity into account compared to a homogeneous system. In the weak model the improvement in plan cost is almost zero due to a very slight difference between broker and link costs. On the other hand we achieve around 10% reduction by considering heterogeneity in computing the dissemination plan.

**Fig. 9.** Next node selection effect on cost reduction rate.

**Fig. 10.** Plan cost reduction percentage for heterogeneous network.

### References


