Multi-Constrained Core Selection for Group Communications

Ayşe Karaman and Hossam Hassanein
School of Computing
Queen's University
Kingston, ON K7L 3N6, Canada
{karaman, hossam}@cs.queensu.ca

***Abstract* - The core-based approach in multipoint communication broadens the solution space in terms of QoS-efficiency of solutions in inter and intra-domain routing. In an earlier work [KH03a], we showed that the constrained cost minimization solutions in core-based approach proposed to date are restrictive in their search to a subrange of solutions, and we proposed *SPAN*, a generic framework to process in the extended solution space. In this paper, we study the core selection component of *SPAN* and propose two novel algorithms, *SPAN/COST* and *SPAN/ADJUST*. Our algorithms consistently outperform their counterparts proposed to date and can be considered pioneering in their optimization range of multiple metrics and processing in the extended solution space.**

1. INTRODUCTION

Multipoint communication is the simultaneous delivery of data stream from each source to a set of receivers in a group for an efficient transmission according to predetermined metrics. Multipoint communication has numerous applications in Internet group communications. Design objectives in Internet routing architectures are characterized under Quality-of-Service (QoS) constraints for the allocation of network resources particular to application demands. Broadband group applications are often delay-sensitive and demanding on network resources. The prominent QoS problem in multipoint communication is *constrained cost minimization* – minimization of total network resources in one metric while meeting a given end-to-end delay bound between source-receiver pairs as a second metric.

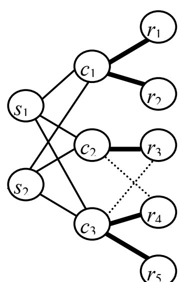
The core-based approach broadens the solution space by efficient placement of cores in the domain especially in the sparse mode. In this approach, the delivery trees are not necessarily rooted at a source any more, but at a domain node that ranks high in efficiency with respect to the operating metrics. Cost-efficiency and reduced maintenance overhead of the communication path are achieved through simplified and separately manage tree structures. The approach additionally offers the placement of cores in distinct autonomous systems allowing tree management "locally" on the availability of the AS information within the particular domain. Core selection – selecting the locations of the cores preliminary to tree construction operations – is crucial in this respect for protocol performance [KH03b]. An efficient core-selection process is likely to improve the performance of multipoint communication protocols significantly.

Core-based QoS-routing solutions proposed to date restricted the processing range to a subset of the solution space, which in turn restricted the efficiency of the potential results [KH03a]. In [KH04], we proposed a basic core selection algorithm, *SPAN*, operating for a distributed, constrained solutions in the extended solution space under the core-based architecture. *SPAN* processed solely on the connectivity information each candidate core provides the group members, yet consistently outperformed its counterparts in the literature. In this paper, we study the core selection algorithms in the extended solution space and propose two algorithms, *COST* and *ADJUST*. *COST* and *ADJUST* define the core-selection component of *SPAN* to account for the cost-approximation of the resulting multipoint communication path to achieve QoS guarantees. In the next section, we layout the problem and the terminology before we examine the existing solutions in the literature. In Section 3, we introduce *COST* and *ADJUST*. We evaluate the performance of *COST* and *ADJUST* against their counterparts in the literature in Section 4. We conclude the paper in Section 5.

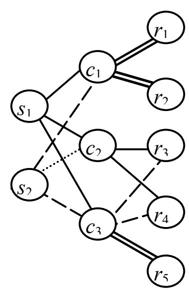
2. PRELIMINARIES

We analyze a solution for a multipoint communication group under the core-based approach as a union of core-based trees each spanning a subset of receivers, and the source-based trees each spanning the set of cores connecting the receivers in respective core-trees to the source and thus serving the source for the group application. The trees are maintained separately and potentially share links.

We say that a receiver r is *dominated by* a core c *for* a particular source s if there exists a path p connecting s to r so that p passes through c without violating the delay bound. Equivalently, we say that a core c serves r_s. We use the notation $D(c,s)$ to indicate the set of receivers that are dominated by the core c for source s. Similarly, we indicate by $D(c,S')$ the domination of a set of receivers that are dominated by the core c for each source in a subset S' of the set of sources. We refer by *multipoint tree* the union of paths serving a $S' \subseteq S$ for all receivers in the group where each core tree constituting the path is identically serving all sources in S'. Consequently on such a multipoint path, the domination sets of the cores in the cluster of cores describing the path partitions the receiver set. Observe that a multipoint tree does not necessarily have a tree structure although it specifies distinct trees each of which is serving a particular source for the delivery of its stream to the receivers. In Figure 1-a, we present an example for a multipoint tree. The source and receiver sets in the multipoint communication group are $S=\{s_1,s_2\}$ and $\{r_1,r_2,r_3,r_4,r_5\}$ respectively. There are 3 cores in the group. Dominating sets are $D(c_1, S) =\{r_1,r_2\}$, $D(c_2, S) =\{r_3\}$, $D(c_3, S) =\{r_4,r_5\}$. There are three core-rooted trees, each spanning the receivers in the domination sets of their respective cores. The multipoint tree is serving both sources in the group. There are two source-rooted trees, one for each source, each spanning the entire cores in the core-cluster corresponding to the multipoint tree. The links on source and core trees are indicated by solid and double lines, respectively. All links except those indicated by dashed line are on the multipoint path. Figure 1-b presents an alternate solution to the same problem in which different trees with non-identical core trees serve the sources. The links serving sources s_1 and s_2 are indicated solid and dashed lines, respectively. Double lines indicate the links serving both sources. The domination sets are $D(c_1, S) =\{r_1,r_2\}$, $D(c_2, s_1) =\{r_3, r_4\}$, $D(c_2, s_2) =\{\}$, $D(c_3, s_1) =\{r_5\}$, $D(c_3, s_2) =\{r_3, r_4,r_5\}$. Unlike the case in Figure 1-a, the receiver set is partitioned differently by the core clusters for each source, with one partition and the dominating core, c_1, common to both. We name the range of solutions that involve respectively a unique core cluster and multiple core clusters for a particular multipoint communication problem as *singular* and *non-singular* solutions. Note that any union of paths constituting a solution to a constrained multipoint communication problem can be formulated as a set of core clusters.



(*a*)



(*b*)

Figure 1. An example constrained-multipoint communication problem. *a*.) a singular solution, *b*.) a non-singular solution for the same problem instance.

The search for solutions in the singular solution space examines the domination of each receiver for all sources in the group. An algorithm to span the solutions in the non-singular space, on the other hand, examines the domination sets separately for each source rather than across all sources. The non-singular solution space expands the range of potential solutions to achieve more efficient results for constrained-group communication. Thus, a heuristic to provide a solution to constrained multipoint communication problem in the entire solution space is considerably preferable to a heuristic to provide a solution in the singular solution space [KH03a].

Only few studies have been conducted on multiple shared trees for multi-source groups. Zappala, Fabbri and Lo [ZFL02], in their *Senders-to-Many* architecture, partition the receiver set among a set of cores and generate a unique core cluster to serve the group. Senders-to-Many is distributed and does not consider a given delay-bound. In its architecture, Senders-to-Many separates core selection and tree construction processes into modular units. Each core and the receivers in the partition corresponding to that core constitute a tree rooted at the core. A source willing to deliver its stream simply sends its data packets to the core from whereon they are transmitted to the receivers via the core-trees.

Salama's [Sa96] multi-core based architecture, *GREEDY*, operates to develop a set of core-based trees each spanning a receiver partition is defined, this time meeting a given delay bound in the solution generated if such a solution exists. Salama's solution is particularly relevant since it is the only routing architecture for multipoint communication in literature providing multi-core solutions for delay-constrained communication groups with multiple sources. A distinctive property of *GREEDY* is that, it assumes bi-directional trees during core selection so that the data stream to be delivered to the multipoint group does not necessarily need to travel to the tree root to be transmitted throughout the tree. Instead, each on-tree node receiving the stream acts as the tree "root", and relays the packets simply to everyone of its links except the incoming link. The architecture necessarily incorporates core selection into tree construction in consideration of bi-directional utilization of the trees in the design stage. Bi-directional development of trees widens the solution space for delay-constrained problems. However, its applicability is restricted to symmetric networks, and it is infeasible for distributed implementation for it necessarily operates on distances between pair of domain nodes rather than node pairs in candidate core and multipoint group member sets. *GREEDY* restricts its search to the singular solution space.

SPAN [KH04] is a distributed, asymmetric framework that operates in the non-singular solution space for constrained groups. *SPAN* initially identifies the potential cores into a pool as candidates by examining their domination characteristics. A domain node is a candidate core only if it connects at least one source-receiver pair within the delay-bound of the application. Each candidate core tests itself on the local state information for its domination for the group, and reports its results to the designated coordinator node in the domain for the selection of cores across these results. The ultimate core set selected among the candidate cores includes the cores that lead to a multipoint path to approximate the optimum solution. The construction of the trees follows core selection and is coordinated separately by the root of each tree.

Let $T_{(c,s)}$ be a core-tree rooted at core c to serve source s for all receivers in $D(c,s)$. According to this definition, $T_{(c,s)}$ "totally" serves s. Consider that $T_{(c,s)}$ serves a source $s' \in S$, so that $D(c,s) \cap D(c,s') \neq \emptyset$ and $D(c,s) \neq D(c,s')$. We say, in this case, that $T_{(c,s)}$ serves s' partially. In an attempt to minimize the multipoint path structure in terms of the number of links on it, the approach is the maximization of combined partial and total utilization of multipoint paths for simplification of ultimate trees in conjunction with efficiency of the transmission. SPAN imposes that a tree $T_{(c,s)}$ constructed to serve its defining source s for all the receivers in $D(c,s)$ also serves, with no modification on the on-tree paths, the rest of the sources for all receivers in set $D(c,s) \cap D(c,s') \ \forall s' \in S$. For all $s' \in S$, a domination $r_{s'} \in T_{(c,s)}$ is served on $T_{(c,s)}$ along the path that serves r_s. According to this, $r_{s'} \in T_{(c,s)} \Leftrightarrow r \in D(c,s) \cap D(c,s')$. Consider the definition of *domination count* of a potential core-tree $T_{(c,s)}$ as follows:

$$domination\ count_{(c,s)} = \sum_{s' \in S} | D(c,s) \cap D(c,s') |$$

Literally, domination count of a (c,s) tuple specifies the number of source-receiver pairs the core tree $T_{(c,s)}$ is capable of serving when the receiver set being served is restricted to $D(c,s)$. The extended solution space offers, during core selection, efficient choice of a core in consideration of each source separately, diverting from the shared tree approach for the efficiency of path construction, update and management during the communication session. However, higher utilization of $T_{(c,s)}$ in terms of the number of sources it is partially or totally serving improves the overall multipoint path structure through tree sharing as the case in core-based approach literature. Furthermore, higher utilization of $T_{(c,s)}$ in terms of the number of receivers it is dominating improves the efficiency of transmission as the case in multicast routing literature. Therefore, high domination count is desirable for core-selection criterion. *SPAN* iterates to select the potential core tree $T_{(c,s)}$ to return the highest domination count for the currently un-dominated receivers. Note that, whenever there is a delay-bound solution, each source in the group is a potential core to serve itself for all the receivers in the group. This implies that the number of cores selected by *SPAN* is at most as great as $|S|$, the size of the source set.

3. CORE SELECTION ALGORITHMS

SPAN conducts its search for cores based solely on the domination count of each core as the selection criterion. In this section, we present two core selection algorithms, *COST* and *ADJUST*. Both algorithms operate within the framework describing *SPAN* itself, and thus both algorithms are distributed, asymmetric, and execute on the non-singular as well as singular solution space. *SPAN/COST* examines the cost distances between the group members and candidate cores as part of the core selection criterion during the selection process. *SPAN/ADJUST*, on the other hand, first executes *SPAN* and processes on the selected core trees to improve the efficiency of the already found solution.

3.1 *SPAN/COST*

SPAN/COST defines core-selction within *SPAN,* taking into account the cost distances to be traveled between the source-core and core-receiver pairs on the trees to be constructed for a better approximation of the resultant paths on the cost metric. As in *SPAN*, it still operates in the non-singular solution space and uses the domination count as a metric for core selection. We use the cost of the minimum delay-distance paths for our approximations. Let *cost*(i,j) denote the cost of the minimum delay-distance path between the nodes i and j. We extend the selection criterion as follows:

$$Criterion_{(c,s)} = R\_factor_{(c,s)} + S\_factor_{(c,s)} + D\_factor_{(c,s)}$$

where

$$R\_factor_{(c,s)} = \frac{\sum_{r \in D(c,s)} \left[(\cos t(c,r) + m) * | S_{(c,s),r} | \right]}{\sum_{r \in D(c,s)} | S_{(c,s),r} | * ave\_dist}$$

$$S\_factor_{(c,s)} = \frac{\sum_{r \in D(c,s)} \left[\frac{\sum_{s' \in S_{(c,s),r}} \cos t(c,s')}{\left| S_{(c,s),r} \right|} + m \right]}{\sum_{r \in D(c,s)} | S_{(c,s),r} | * ave\_dist}$$

$$D\_factor_{(c,s)} = 1 - \frac{\sum_{s' \in S} \left| D(c,s) \cap D(c,s') \right|}{| S | * | R | - \sum_{(a,b)} \sum_{b' \in S} \left| D_u(a,b) \cap D_u(a,b') \right|}$$

$$ave\_dist = \frac{\sum_{\substack{r \in D(c,s), \\ (c,s) \in CxS}} \left[\cos t(c,r) * | S_{(c,s),r} | \right]}{\sum_{\substack{r \in D(c,s), \\ (c,s) \in CxS}} | S_{(c,s),r} |} + m$$

The second term in the $D\_factor_{(c,s)}$ expression is literally the proportion of the domination count of the potential core tree $T_{(c,s)}$ to the currently un-dominated receivers across the sources. $D\_factor_{(c,s)}$ in return is merely a transformation of $domination\_count_{(c,s)}$ to a scale in a definite range in which it is to be minimized for the highest value of $domination\_count_{(c,s)}$. In the basic *SPAN*, $Criterion_{(c,s)} = D\_factor_{(c,s)}$.

Let $S_{(c,s),r}$ be the set of all sources served by the tree $T_{(c,s)}$ to dominate the r. The $R\_factor_{(c,s)}$ accounts for the cost distances between a core c, and receivers dominated on its respective core tree. The value cost(c,r)*$|S_{(c,s),r}|$ accumulates the cost distances to be traveled between the core c and the receiver r separately for each source in $S_{(c,s),r}$ transmitting its stream to r via the tree $T_{(c,s)}$. The numerator of *R_factor*, $\sum_{r \in D(c,s)} \left[\cos t(c,r) * | S_{(c,s),r} | \right]$ in return is the sum of all core-receiver distances across the receivers dominated on the tree $T_{(c,s)}$. Note that this value is an approximation, since the paths leading to multiple receivers on the tree $T_{(c,s)}$ are potentially shared, and thus the cost of transmission from a source to multiple receivers sharing paths leading to them on the core tree is less than their additional cost value. *R_factor* divides by $\sum_{r \in D(c,s)} | S_{(c,s),r} | = | \{ r_{s'} | r_{s'} \in T_{(c,s)} \} |$ to take the simple average of the approximated cost.

S_factor measures the cost distances between core-source pairs. $\sum_{s' \in S_{(c,s),r}} \cos t(c,s') / \left| S_{(c,s),r} \right|$ represents the average of all transmission costs between each source served on $T_{(c,s)}$ for a given receiver r dominated on the tree. We accumulate the average source-to-core transmission costs for all the receivers dominated on the tree $T_{(c,s)}$. The packets transmitted from a source to a core are delivered to multiple receivers on the core-tree. Therefore, the more receivers are served for a particular source *s'* on $T_{(c,s)}$, the less the cost between *s'* and *c* is accounted for. Due to this, we diminish each $\sum_{s' \in S_{(c,s),r}} \cos t(c,s') / \left| S_{(c,s),r} \right|$ further by the by the size of the set $\{s' | r_{s'} \in T_{(c,s)},\ s' \in S_{(c,s),r}\}$, and divide *S_factor* by $\sum_{r \in D(c,s)} | S_{(c,s),r} | = |\{r_{s'} | r_{s'} \in T_{(c,s)}\}|$ rather than $|D(c,s)|$ to take the simple average across the receivers served on $T_{(c,s)}$.

```
.C1.. for (each c∈C)   for (each s∈S)   {   // O(|S||C|)    [O(|S|^2|R||C|)]
C2..      // compute domination_counts
C3..        domination count(c,s) = |D(c,s)|;
C4..        for (each s'∈S, s'≠s)                        //  O(|S|)    [O(|S||R|)]
C5..            domination count(c,s) += |D(c,s) ∩ D(c,s')|;   // O(|R|)
C6.. }
C7..  for (each c∈C)   for (each s∈S)
          for (each r∈ D(c,s))   for (each s'∈S)    // O(|S|^2|R||C|).
          //    compute S(c,s),r values
C8..            if (r∈ D(c,s'))  {
C9..                S(c,s),r = S(c,s),r ∪ {s'}
C10..               S(c,s),r .cost += cost(s',c);
C11..               |S(c,s),r|++;
C12..           };
C13.. compute ave_dist =                                          // O(|S||R||C|)
```

$$\sum_{\substack{r \in D(c,s), \\ (c,s) \in CxS}} \left[\cos t(c,r) * | S_{(c,s),r} | \right] \Big/ \sum_{\substack{r \in D(c,s), \\ (c,s) \in CxS}} | S_{(c,s),r} | + m;$$

```
C14..  total_count = 0;
C15..  repeat   // O(|Cu|) = O(|S|)    [O(|S|^3|R||C|)]
C16..    if (|C| =|S|) exit;
C17..    min_value = ∞;
C18..    for (each c∈C)   for (each s∈S)  {  // O(|S||C|)    [O(|S||R||C|)]
         // compute criterion for each candidate core and select the best scoring one
C19..        NR = 0;   NS = 0;   D = 0;
C20..        for (each r∈ D(c,s))   {   //   O(|R|)
C21..            NS += S(c,s),r.cost/|S(c,s),r|;   // numerator of S_factor(c,s);
C22..            NR += cost(c,r)*|S(c,s),r|;    // numerator of R_factor(c,s);
C23..            D += S(c,s),r|;    // denominator of both factors
C24..        };
C25..        R_factor(c,s) =  NR / (D*ave_cost_dist)
C26..        S_factor(c,s) =  NS / (D*ave_cost_dist)
C27..       D_factor(c,s)=1- domination count(c,s) /(|R|*|S|-total_count)
C28..        criterion(c,s)  = R_factor(c,s) + S_factor(c,s) + D_factor(c,s);
C29..        if (criterion(c,s) <min_value)  {
C30..            min_value = criterion(c,s);
C31..                core = c;
C32..                source = s;
C33..        };
C34..    };
C35..    total_count += domination count(core,source);
C36..    domination count(core,source) = 0;
C37..    Du(core,source) = D(core,source);
C38..    Cu = Cu ∪{(core, source)}
C39..    C = C \ {core};
C40..    for (each s∈S)  for (each r∈ D(core,source))   // O(|S||R|)    [O(|S|^2|R||C|)]
C41..        if (r∈ D(core,s))    {
C42..          Du(core,s) = Du(core,s) ∪{r};
C43..          for (each c∈C, c ≠ core)   {   // O(|C|)    [O(|S||C|)]
C44..                S(c,s),r = ∅;
C45..                for (each s'∈S)      S(c,s'),r = S(c,s'),r \ {s};      // O(|S|)
C46..                if (r∈ D(c,s))    domination count(c,s) --;
C47..                D(c,s) = D(c,s) \{r};
C48..            };
C49..     };
C50..  until (total_count = |R|*|S|)
```

Figure 2. Pseudo-code of *SPAN/COST*.

In an attempt to normalize *R_factor* and *S_factor*, we further divide both factors by *ave_dist*, which is the value to approximate the average of all cost-distances to be traveled between the potential cores and the receivers to be dominated on their respective trees. We shift the cost values by *m*, the minimum of the link costs in the domain to allow equal consideration of *cost*(*c,r*) in *R_factor* for the case *c*=*r* when it multiplies. The value *m* appears in *S_factor* and *ave_dist* in the simplified form of the formulae.

In Figure 2 we present the *SPAN/COST* algorithm. D_u and C_u respectively denote the ultimate domination and core sets for the selected core trees. Note that $D_u(core,s)\ \forall s \in S$ for specify a selected core tree $T_{(core,source)}$. The algorithm starts out with the initialization of the sources set attributes (lines C7-12) and domination counts (lines C1-6) for all possible core trees. The domination counts and the attributes of each $S_{(c,s),r}$ for the remaining cores are updated whenever a new core is selected throughout the algorithm (lines C40-49). Line 13 indicates the one-time computation of *ave_dist*. *SPAN/COST* does not have an inherent bound on the number of cores selected. We force the upper bound on the ultimate core set for *SPAN* for *SPAN/COST* as well, and terminate the algorithm whenever the core set size exceeds the number of sources and assume *SPAN* to be invoked for the core selection for the particular problem instance (line C16). The block in lines C18-35 computes $Criterion_{(c,s)}$ for each candidate-core and source tuple (lines C18-28) and compares the value to the current minimum value, *min_value* (lines C29-33). *total_count* keeps track of the number of current dominations, i.e., the number of distinct r_s dominated by an already selected core tree. The main loop (lines C15-50) iterates to select exactly one core tree each time, and terminates when all the receivers are dominated for all sources.

The computational complexity of each computation step is depicted on the line as comments. The lines of which the computation time is bound by a constant are not specified. The values indicated in square brackets denote the overall complexity of the loop-statements when the nested blocks are accounted. It can be shown that *SPAN/COST* executes in time $O(|S|^3|R||C|)$.

3.2 *SPAN/ADJUST*

SPAN selects the cores based on their domination counts. The resulting cores combine all the currently un-dominated receivers on their respective trees with no regard to the cost-proximity of the receivers to the cores. In this section, we introduce an alternative non-singular algorithm, which we call *ADJUST*. *ADJUST* first runs *SPAN* to select the cores using domination count as the sole criterion, then adjusts the existing trees by moving the receiver between the trees for their less-costly dominations. The cost function relevant for this case to measure the cost distances between the core-receiver pairs needs also to account, Δ, the delay-bound of the application, in order to determine whether the path being examined meets the delay bound:

$$cost'(c,r_s) = \begin{cases} cost(c,r) & ;\text{ if } delay(c,r)+delay(s,c)<\Delta \\ \infty & ;\text{ otherwise} \end{cases}$$

Consider *main domination* and *ordinary domination* to define $r_s \in T_{(c,s)}$ and $r_{s'\neq s} \in T_{(c,s)}$ respectively. Observe that an ordinary domination for a particular receiver can be on a given tree only if the main domination for that receiver is on that tree. *ADJUST* distinctively considers the following cases for the movement of dominations between the trees:

a) <u>from ordinary to main domination</u>: Let $r_{s'}$ be an ordinary domination on a core tree $T_{(c,s)}$ so that $s' \neq s$. The necessary condition for the movement of $r_{s'}$ to a core tree $T_{(c'',s'')}$ for main domination requires that $s'=s''$ and $cost'(c'', r_{s'}) < \infty$.

b) <u>from main to any domination</u>: Let r_s be a main domination on a core tree $T_{(c,s)}$. The necessary condition for the movement of r_s to a core tree $T_{(c',s')}$ for any domination requires that $s=s'$ (which is the sole necessary condition if $r_{s'} \in T_{(c,s)} \Rightarrow s'=s$) or $r_{s'} \in T_{(c',s')}$, and $cost'(c', r_{s''}) < \infty$ for all $r_{s''}$ where $r_{s''} \in T_{(c,s)}$.

c) <u>from ordinary to ordinary domination</u>: Let $r_{s'}$ be an ordinary domination on a core tree $T_{(c,s)}$ so that $s' \neq s$. The necessary condition for the movement of $r_{s'}$ to a core tree $T_{(c'',s'')}$ for ordinary domination requires that $r_{s''} \in T_{(c'',s'')}$ and $cost'(c'', r_{s'}) < \infty$.

```
// from ordinary to main
A1.. for (each core tree T(c,s))     // O(|Cu|) = O(|S|)   [O(|S|^2|R|)]
A2..    for (each r∈R)               // O(|R|)   [O(|S||R|)]
A3..       if (rs∉ T(c,s))   {
A4..          locate T(c',s') where rs∈ T(c',s');    // O(|Cu|) = O(|S|)
A5..          if ( (cost'(c, rs)< cost'(c', rs) ) & (s'≠s) )
A6..                  { move rs from T(c',s') to T(c,s) }
A7..       }
// from main to any
A8.. for (each core tree T(c,s))   for (each r∈R)  // O(|Cu||R|)=O(|S||R|)    [O(|S|^2|R|^2)]
A9..     if ( (rs∈ T(c,s)) & (rs is not marked) ) {
A10..       min_value = cost'(c, rs);
A11..       for (each core tree T(c',s'))       // O(|Cu|) = O(|S|)    [O(|S||R|)]
A12..          if ( (rs'∈ T(c',s')  ||  s'=s) && (cost'(c', rs) < min_value) )   {
A13..             dependants = true;
A14..             for (each rs''∈ T(c,s))   //   O(|R|)
A15..                 if (cost'(c', rs'') = ∞)   dependants = false;
A16..             if (dependants)    { min_value = cost'(c', rs);  toTree = T(c',s');  }
A17..          }
A18..          mark and move rs' ∀s'∈S from T(c,s) to toTree;  // O(|S|)
A19..    }
// from ordinary to ordinary
A20.. for (each s∈S)  for (each r∈R)   if (rs is not marked) { //O(|S||R|)   [O(|S|^2|R|)]
A21..     locate T(c',s') where rs∈ T(c',s');      // O(|Cu|) = O(|S|)
A22..     if (s'≠s)  {    // if not a main domination
A23..         min_value = cost'(c', rs);
A24..         for (each core tree T(c'',s''))       // O(|Cu|) = O(|S|)
A25..             if ( (rs'' ∈ T(c'',s'')) & (cost'(c'', rs) < min_value) )
A26..                 {  min_value = cost'(c'', rs);  toTree = T(c'',s''); }
A27..         mark and move rs from T(c',s') to toTree;
A28..     }
A29.. }
```

Figure 3. Pseudo-code of *SPAN*/*ADJUST*.

ADJUST (Figure 3) applies the intuition to initially establish the main dominations based on their ultimate locations on the core trees for which the distance of the receiver in the domination is closest in cost to the core of the tree. The algorithm first examines the ordinary dominations for their main dominations on the selected core trees (Case (*a*), lines A1-7). When there is a core tree rooted at a core, say *c*, satisfying the necessary condition for this case, the cost distance of the receiver to its currently dominating core and the core *c* are compared, and the domination is moved if the cost distance on the newly found tree turns out to be less. In case (*a*), the fist tree found to provide a better cost is selected to move the domination. Case (*b*) (lines A8-19) is examined after, and complements case (*a*) in that this case examines the main dominations for their least costly dominations across *all* satisfactory core-trees. In this case, all the core trees satisfying the necessary condition are examined for the least-costly core tree. At the end of this stage, all main dominations are set at their ultimate trees to provide potential ordinary dominations. At the final stage (lines A20-29), the algorithm adjusts the ordinary dominations for their least-costly locations on the set of core-trees.

Lines A8-19 of *ADJUST*, examining the movement of the main dominations, additionally iterate on the receivers set to see whether there exists a "dependant" ordinary domination on the main domination to be moved to violate the necessary condition in this case. Due to this, the computational complexity of the block executing Case (*b*) is higher by a factor of $|R|$ than the rest of the algorithm and is $O(|S|^2|R|^2)$, which then also is the overall complexity of the algorithm.

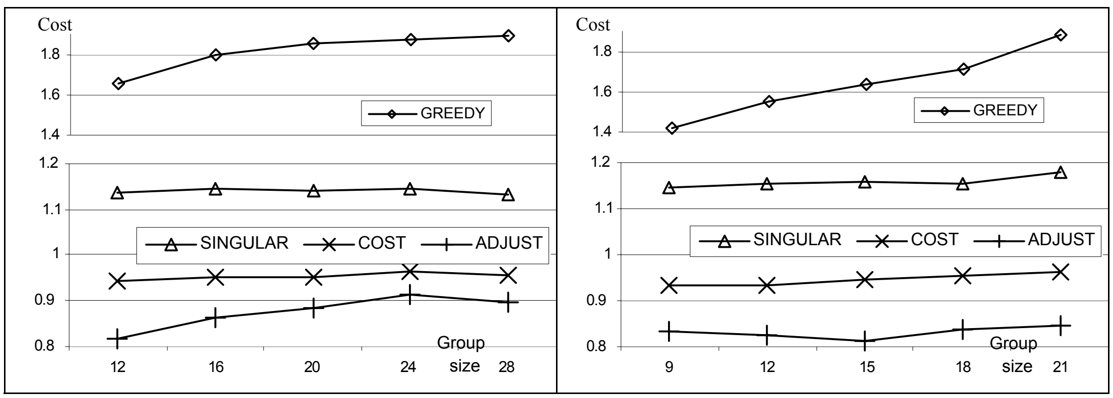
4. Performance Evaluation

As we noted in Section 2, the prominent model for design specifications for a comparison to a multipoint communication problem is *GREEDY* [Sa96]. *GREEDY* operates in singular solution space and is centralized. For a direct comparison of our model to their distributed counterparts in the singular solution space, we generated a model, *SINGULAR*, which applies the entire architecture of *SPAN* and its extensions this time to process in the singular solution space. *SINGULAR* differs from *SPAN* in its domination count which now is an attribute of a core rather than a core-source pair as the attribute uniquely describing a core-tree, i.e., $r_s \in T_{(c,s)} \Leftrightarrow r \in D_u(c,S) \;\; \forall r \in R,\ s \in S,\ c \in C_u$. According to this, the domination count is described for a core rather than core-source tuple, and specifies the number of receivers dominated by the core for all sources in the group. The candidate core *c* returning the highest value for $|D(c,S)|$ as the primary criterion and *c* being closest in average cost-distance to source set as the secondary criterion is selected to be the next member of the core set.

We tested the performance of all heuristics in terms of the cost metric, which is the sum of the costs of all the links traversed by exactly one data packet from each source to each receiver in the group. We used sample domains of size 60, and tested groups sparsely distributed throughout the domain. We used Waxman's model [W88] for sampling the domains. Our domains have the average node degree in range (3.5,5). In all cases, the source and receivers in the group are randomly distributed in the domain. We maintained 10% confidence intervals with a 90% confidence level in our measurements.

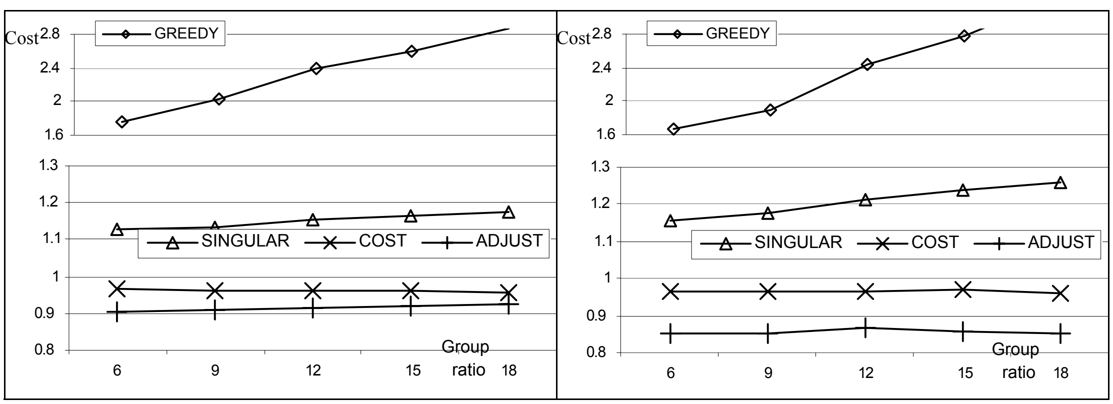
We conducted all measurements on normalized delay-bound values for each case. Consider the source and receivers sets, respectively *S* and *R* in a group. We define *critical-delta,*

$\Delta_{critical}$, for the group in a given domain as $\max_{s\in S, r\in R}\{delay(s,r)\}$. In other words, $\Delta_{critical}$ is the minimum delay-bound that leads to a successful solution for the group, and the domain provides no solution for the given multipoint communication group sample if the delay-bound is any smaller than $\Delta_{critical}$. Consider also the definition of *maximum-delta,* Δ_{max}, which is $\max_{s\in S, r\in R, c\in C}\{delay(s,c)+delay(c,r)\}$. Maximum-delta specifies the "boundary" where the results to be returned by anyone of the algorithms are no longer affected by the delay-bound parameter. In other words, maximum-delta is the upper-bound for the delay-bound range, beyond which any algorithm would be feasible. The range $[\Delta_{critical}, \Delta_{max}]$ specifies, for a problem instance on a given model the minimal delay-bound range of all possible solutions. We also normalized the cost results of each algorithm for the results of the "reference" model, which we chose as *SPAN*. According to this, the cost results of each one of the other models are divided by the corresponding outcome obtained from the reference under the same measurement setup, and the indicated results on cost are relative performances to that of the reference. In our figures, the performance of the reference model is unity and is not explicitly depicted.



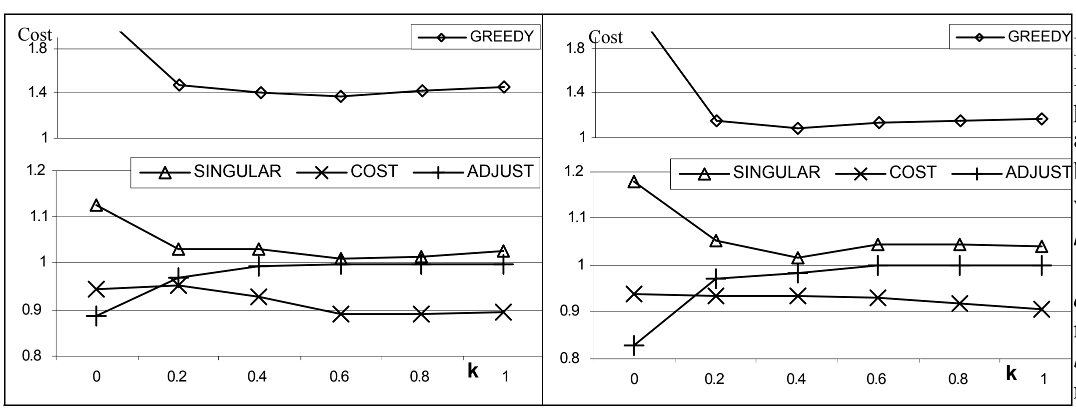
(*a*) (*b*)

Figure 4. Evaluation for cost performance on group size. $|R| \in [9, 12, 15, 18, 21]$, $|S| \in [3, 4, 5, 6, 7]$, $|S|/|R| = 1/3$. $\Delta=\Delta_{critical}$. *a*.) No sources are receivers. *b*.) All sources are receivers.



(*a*) (*b*)

Figure 5. Evaluation for cost performance on group ratio. $|R| = 24$. $|S| \in [6, 9, 12, 15, 18]$. $\Delta=\Delta_{critical}$. *a*.) No sources are receivers. *b*.) All sources are receivers.



(*a*) (*b*)

Figure 6. Evaluation for cost performance on application delay-bound. $\Delta=\Delta_{critical}+kV$ where $V=\Delta_{max}-\Delta_{critical}$, $k\in[0, 0.2, 0.4, 0.6, 0.8, 1.0]$. $|R| = 18$. $|S| = 9$. *a*.) No sources are receivers. *b*.) All sources are receivers.

Figures 4-6 show the results of our evaluations of the models tested on the group size, sources-to-receivers ratio, and the delay-bound of the application respectively. A prominent finding of our measurements is the poor performance of *GREEDY* compared to the models tested. From our analysis perspective, *GREEDY* does not consider core-trees and source-trees separately and constructs, for each receiver partition, a tree rooted at the core spanning all the sources in the group. The resulting tree combines the core trees and source trees and the data stream from a particular source is redundantly delivered to the other sources as well as to the receivers, adding on the delivery cost. Our figures depict *GREEDY* at its extreme end of the scale, leaving the relatively close performance of the remaining models in their "magnified" range for a sophisticated comparison. The difference in the performances of *SPAN* and *SINGULAR* attributes solely to their respective core selection algorithms since both models use the exact same tree construction module. Thus, *SPAN*'s higher cost-performance is a direct verification of our claim that non-singular solution space offers potential improvement on the efficiency of the solutions.

ADJUST outperforms *COST* for different group sizes and ratios of group members whenever the delay-bound is tight. At tight delay-bounds, *ADJUST* picks up the high performance of *SPAN* and further improves the results for cost efficiency. Domination count is a measure of the structure of the resulting multipoint paths – optimization on the domination count of the selected cores leads to less complicated trees. *ADJUST*'s performance advantage over *COST* at critical values of the application delay bound shows the significance of the path structure on the cost-efficiency of the path. However, *SPAN* tends to reduce the number of core selected when the delay-bound of the application is relaxed, leaving no room for *ADJUST* to find further improvements among the core trees. *COST'*s relatively steady performance over *SPAN* further reflects the effectiveness of the domination count as a criterion for measuring the transmission cost of the resulting solution.

ADJUST's performance improves when the source and receiver sets overlap under the same experimental setup (Case (*b*) compared to (*a*) in all figures). This result is expected, since *SPAN* places the group members on the first selected core tree qualifying to serve/dominate each member. *ADJUST* further moves the members among the trees for their placement for cost efficiency, and the potential gain on the accurate placement of a member which is both a source and a receiver in the group is higher.

5. CONCLUSIONS

In this paper, we presented two core selection algorithms, *COST* and *ADJUST*. our algorithms, are the first core selection algorithms in the literature [KH03b] processing on multiple metrics, cost and delay, for the optimization of the results for QoS orientation. Both algorithms consistently outperformed their counterparts in the literature. *ADJUST* further improved the cost efficiency up to 20% compared to the algorithms proposed to date. *ADJUST* is particularly preferable for any group application demanding tight delay bounds while *COST* is at its best performance at relatively relaxed bounds compared to the model proposed. Both algorithms operate in a distributed, asymmetric framework for constrained solutions, and are applicable to Internet routing domains operational today.

Broadband group communication over the Internet is becoming ubiquitous over a wide range of services. These applications are usually delay-sensitive and demanding on network resources. The core-based architecture offers the significant advantage of partitioning the inter-domain route construction on QoS-demands of the applications into intra-domain problems by the placement of core nodes within the autonomous routing domains to coordinate transmission to receivers and/or to border routers for further transmission across ASs. *SPAN* and its extensions presented in this paper operate on local distance-vector information available at the routers, with no modification on their functionality. Our models can further enhance the support of inter-domain groups for participants across ASs through the construction and management of the intra-domain routes coordinated by the efficient placement of cores in each of the domains.

REFERENCES

[KH03a] Karaman, A., Hassanein, H. S., *DCMC – Delay-Constrained Multipoint Communication with Multiple Sources,* IEEE Symposium on Computers and Communications (ISCC), Kemer, ,Turkey, 2003

[KH03b] Karaman, A., Hassanein, H. S., *Core-based Approach in Multicast Routing Protocols*, International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS), Montreal, Canada, 2003

[KH04] Karaman, A., Hassanein, H. S., *Extended QoS-Framework for Multipoint Communications*, to appear, IEEE Symposium on Computers and Communications (ISCC), Alexandria, Egypt, 2004.

[Sa96] Salama, H. F., *Multicast Routing for Real-Time Communication on High-Speed Networks*, Ph.D. Thesis, Department of Electrical and Computer Engineering, North Caroline State University, 1996

[W88] Waxman, B., M., Routing of Multipoint Connections, IEEE Journal of Selected Areas in Communication, pp.1617-1622, 1988

[ZFL02] Zappala, D., Fabbri, A. and Lo, V., *An Evaluation of Multicast Trees with Multiple Active Cores*, Journal of Telecommunication Systems, Kluwer, March 2002.