Overlay Subgroup Communication in Large-Scale Multicast Applications [*]

Jangwon Lee[†]
Qualcomm Inc.
jangwonl@qualcomm.com

Gustavo de Veciana
Electrical and Computer Eng. Dept.
University of Texas at Austin
gustavo@ece.utexas.edu

ABSTRACT

In the paper, we consider a *preference heterogeneity* problem in large-scale multicast applications. Abundant content, data type and diverse members' interests naturally lead to preference heterogeneity within a multicast session requiring frequent communication within subgroups of members sharing common interests/requirements. In this paper, we take an overlay approach which builds topology-sensitive subgroup communication (TSC) structures to support efficient subgroup communications in large-scale multicast applications. Our TSC mechanism completely eliminates additional creation of multicast groups while minimizing the exposure of unnecessary packets to members and links. Our mechanism exploits the spatial locality of members within a given subgroup, and enables members to autonomously build a TSC structure consisting of multiple unicast and scoped multicast connections.

Simulations using real topology data show that TSC mechanism performs well for diverse configurations with different densities and distributions of nodes in a subgroup.

Keywords

Multicast, Subgroup, Topology-sensitive, Overlay

1. INTRODUCTION

IP multicast is an efficient one-to-many or many-to-many delivery method which can provide a number of operational advantages for content and network providers by reducing the overall resources consumed to achieve such distribution. A single packet transmitted by the source traverses each link in the multicast distribution tree to all receivers in the multicast group. Due to intensive needs for high bandwidth requirement, large-scale interactive applications such as distributed interactive simulations (DIS), video conferencing tools and multi-player games can benefit from IP multicast. Although the members in such applications join a multicast session for some common goal, abundant content, data type and heterogeneity in members' interests naturally lead to *preference heterogeneity* within sessions [1], requiring frequent communication within *subgroups* of members sharing common interests/requirements.

[*]Partial work of this paper was presented at Proceedings of the 4th International Workshop on Networked Group Communication (NGC), Oct. 2002

[†]Corresponding author, Tel:858-350-9755, Fax:858-845-2651.

A multicast session shared by all members (referred to as the global multicast session) can be used to support subgroup communication. However, this may lead to inefficiency, i.e., packets are delivered to the entire tree, which results in wasted bandwidth and CPU processing power to transmit and handle unnecessary packets. This is referred to as the *exposure* problem. The exposure problem can be completely eliminated if data is only forwarded along a tree induced by the members of each subgroup, as required. This can be achieved by creating a new multicast session for each subgroup. However, this requires routers to store multicast forwarding state information for each subgroup, which can cause a significant scalability problem as the number of subgroups increases [2, 3]. Thus, mechanisms to handle preference heterogeneity should consider the both exposure problem and the scalability of multicast forwarding state problem.

Most existing approaches to the preference heterogeneity problem focus on developing *clustering* frameworks, i.e., given a limited number of multicast sessions, determine how to best cluster multiple subgroups into multicast sessions based on a preference matrix [1] or players' positions in a virtual cell [4].

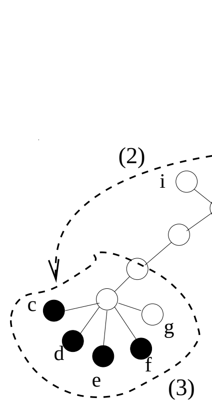


Figure 1: An illustration of a subgroup communication in a TSC mechanism.

In the paper, we propose a topology-sensitive subgroup communication (TSC) mechanism to support efficient subgroup communication in large-scale multicast applications. Our TSC mechanism allows members in a subgroup to autonomously build a TSC structure consisting of multiple unicast and scoped multicast connections.

For example, consider a distribution tree for a multicast session, G, in Figure 1. All end nodes are members of G and the black end nodes are members of a subgroup, S. In our scheme, when a wishes to send packets to other members in S, packets will be delivered as follows: (1) $a \longrightarrow b$ via unicast; (2) $b \longrightarrow c$ via unicast; and, (3) $c \longrightarrow \{d, e, f\}$ via multicasting with a TTL scope of 2 as shown in Figure 1. We assume that the multicast tree for G is a bidirectional shared one. [1]

Note that in the example, the use of unicast can suppress the exposure and the use of scoped multicast can reduce duplicate packets traversing the same link. Our approach does not require the creation of new multicast sessions, which can completely eliminate any additional multicast forwarding state except those of the global session. It tries to minimize the exposure by exploiting spatial locality among members within a given subgroup.

Throughout simulations, we study which environments are advantageous to apply the proposed mechanism and other existing approaches, e.g., global multicast tree or unicast. Simulation results show that under various configurations of density and distribution modes of a subgroup, the sensitivity of our TSC mechanism is small compared to others. This is especially beneficial in the case where information about subgroups is not available - as is likely to be the case in practice.

The paper is organized as follows. We discuss related work and contrast them with our work in Section 2. In Section 3 we discuss how to construct and maintain a TSC structure. In Section 4, we evaluate and compare the proposed TSC mechanism with other schemes in various environments. Section 5 concludes the paper.

2. RELATED WORK

The scalability of state associated with multicast forwarding by routers has been one of the challenges in a wide deployment of IP multicast. Reduction of multicast forwarding state at routers can be achieved through aggregation or elimination of non-branching approaches. In [2], multiple multicast forwarding entries are aggregated if entries have adjacent group address prefixes and matching incoming and outgoing interfaces. The goal of dynamic tunnel multicast [8] and REUNITE [9] is to reduce multicast states by eliminating non-branching point. That is, only fan-out (branching) points keep state information, which is mostly beneficial in a sparse distribution of members.

The clustering schemes aim to efficiently cluster members into a limited number of multicast sessions based on a preference matrix [1] or players' position in a virtual cell [4]. Note that the first two approaches (aggregation and non-branching elimination) are at the routing level, that is, trying to eliminate multicast forwarding state at each router. However, the clustering schemes are at the application level, i.e., aim at reducing the number of multicast groups using application specific information. Thus, the first two approaches can be applied to any single multicast group and the clustering schemes are for large-scale multicast applications consisting of lots of subgroups, which is our target in the paper.

The proposed TSC scheme completely eliminates creation of additional multicast groups and takes a full end-to-end approach for subgroup communication. A single multicast channel is efficiently used in the scheme for multiple roles: (1) data forwarding for the entire multicast session G, (2) providing a control channel for discovery of subgroup members and constructing a TSC forwarding structure, and (3) forwarding data to subgroup members via scoping. Unlike existing clustering frameworks, it does not require correlated information among subgroups, which eliminates the need for a central point where the information is collected and grouping decisions are made.

Overlay solution in our approach has a similarity with a number of recent application-level multicast studies, e.g., [10], [11], [12], [13], [14]. However, our approach should be contrasted with them. First, the goal of application-level multicast is the replacement of IP multicast due to a number of challenges such as infrastructure modification, reliability, flow and congestion control. However, we use the end-to-end approach for reduction of multicast forwarding state in large-scale multicast applications. In our view IP multicast and application-level multicast may coexist and IP multicast will survive as an important delivery mechanism to serve very large-sized groups. Second, our TSC mechanism is not a simple adaptation of unicast overlay solution to the preference heterogeneity problem. It uses scoped multicast by exploiting spatial locality among members. By varying the exposure threshold, it can position itself in the middle of two extreme points: a global multicast and unicast overlay solution. Third, application-level multicast approaches need a out-of-band bootstrap mechanism to allow new nodes to join and initiate constructing overlay structures. On the other hand, our approach does not require such a mechanism since it uses the existing multicast session for such purpose.

3. TSC MECHANISM

For each subgroup, a *TSC structure* is created in a TSC mechanism. The TSC structure serves as a communication channel for members in the given subgroup. In this section, we introduce a TSC structure, and describe how to construct and maintain it.

3.1 TSC structure

A TSC structure consists of multiple unicast and scoped multicast connections among members in a subgroup. For example, given G and $S = \{a,\ b,\ c,\ d,\ e,\ f\}$ in Figure 1,

[1] Our mechanism targets many-to-many large-scale multicast applications where each member can be a sender and/or receiver. For such applications, it is generally agreed that shared multicast routing protocols are more efficient than source based ones. Even though PIM-SM [5], widely deployed for shared multicast routing, takes a unidirectional forwarding mechanism, we argue that bidirectional forwarding mechanisms are more efficient. The larger the multicast session and the more the demand for local communication, the larger the overhead incurred by using a unidirectional tree. Reflecting these observations, the long term inter-domain routing solution, Border Gateway Multicast Protocol(BGMP) [6] currently under development, constructs bidirectional shared trees like CBT [7].

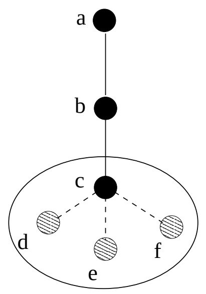


Figure 2: A TSC structure for Figure 1.

Figure 2 depicts an example of a TSC structure exhibiting an overlay structure among members in the subgroup S. The solid and dashed lines represent unicast and TTL scoped multicast respectively.

Subgroup members in a TSC scheme are classified into two types: *normal* and *head* members. A normal member is associated with a head member. Head members, denoted by a set H_S, communicate with each other via unicast connections in a TSC structure. The role of the head members is two-fold: (1) they participate in constructing a unicast overlay structure, and (2) they perform scoped multicast forwarding to their associated normal members. We define an *island* as a set of nodes consisting of a head and its normal nodes. Note that it is possible to have one member island where there are no normal nodes associated with the head node. For example, $H_S = \{a,\ b,\ c\}$ and $\{a\}$,$\{b\}$, $\{c,\ d,\ e,\ f\}$ are collection of islands of the TSC structure in Figure 2.

When constructing TSC structures, there is a tradeoff between member exposure and bandwidth wastes. If there are only one-member islands in a TSC structure, i.e., no use of scoped multicast, it completely eliminates the *member exposure* problem. However, this will introduce performance penalties, i.e., duplicate packets on the same physical links. The use of TTL scoped multicast may introduce a member exposure, (e.g., a member g in Figure 1 is exposed to S subgroup communication with the TSC structure in Figure 2), but it can reduce bandwidth wastes especially in the case where subgroup members are clustered with each other.

Thus, our goal is to build an efficient TSC structure which minimizes wasted bandwidth while limiting the exposure of non-subgroup members. Building TSC structures involves two stages: constructing islands and then connecting islands.

3.2 Constructing islands

Since we envisage that a head node forwards packets to its normal nodes via a TTL scoped multicast for subgroup communication, an island can be specified by a head node and an associated TTL scope, called *radius*. Thus, constructing islands requires (1) each member in S to decide its role between head and normal, (2) a normal node, say a, to decide its head node, *head*(a), and (3) a head node, $h \in H_S$, to decide its radius, $r_S(h)$.

Each node initially considers itself as a head candidate and computes its radius. Thus, constructing islands consists of two algorithms : (1) radius selection algorithm and (2) head election algorithm. Observe that there is an important trade-off in selecting the radius. If it is too large, there may be a high exposure, and if it is too small, we underutilize the use of scoped multicasts. In order to control the degree of exposure, we use *exposure ratio* defined as below.

Let the set $N(a,t)$ be the set of members in G within a TTL distance of t from $a \in G$ (excluding a itself), i.e., $N(a,t) = \{b \mid b \in G,\ 0 < d(a,b) \leq t\}$ where $d(a,b)$ represents TTL distance between a and b. Note that if a performs a scoped multicast with a TTL scope of t, packets will be delivered to all the nodes in $N(a,t)$. Let $N_S(a,t)$ denote the set of subgroup S members in $N(a,t)$, i.e., $\{n \mid n \in S \cap N(a,t)\}$. For example, in Figure 1, $N(c,2) = \{d,e,f,g\}$ and $N_S(c,2) = \{d,e,f\}$. For $a \in S$ and a given t, the *exposure ratio* $\beta_S(a,t)$, is defined as follows:[2]

$$\beta_S(a,t) = \begin{cases} 1 & \text{, if } |N(a,t)| = 0, \\ \frac{|N(a,t) \setminus N_S(a,t)|}{|N(a,t)|} & \text{, otherwise.} \end{cases}$$

Next, we describe how each member can compute an exposure ratio for a given TTL scope and subgroup. Let L_a be a set of subgroups which a member a wishes to join. Each member, $a \in G$, periodically multicasts a *subgroup advertisement* packet containing L_a with a fixed TTL distance k. Then each member can maintain a *TTL-neighbor profile* storing tuples of all neighboring members and their subgroup lists along with TTL distance up to k. Distance information between members can be obtained by senders inserting initial TTL value in packets. This enables a receiver node to compute its TTL(path) distance from the sender by simply subtracting the value in TTL field from initial TTL value.

Figures 3 and 4 show an example of the TTL-neighbor profile of member c in Figure 1 and its exposure ratio respectively when k is 5. With the TTL-neighbor profile, each node can easily obtain exposure ratios up to TTL scopes of k. Note that the scope k value should be large enough to create an efficient and large island, but also should be small enough not to incur too much traffic. (We explore its impact on the performance in Section 4.)

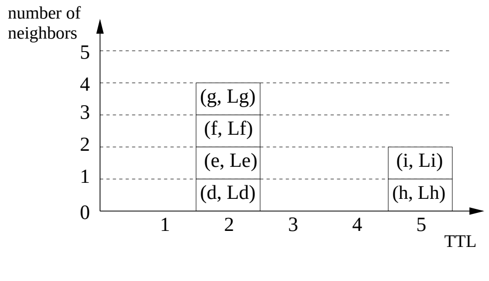


Figure 3: TTL-neighbor profile of c in Figure 1

Note that the exposure ratio $\beta_S(a,t)$ can indicate whether a scoped multicast performed by a with a TTL scope of t is efficient or not. That is, when the exposure ratio is

[2] $A \setminus B$ represents A minus B, i.e., elements from A that are not in B. $|A|$ denotes the cardinality of a set A.

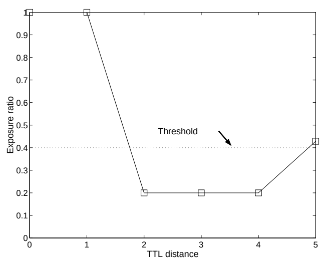


Figure 4: Exposure ratio of c in Figure 1

```
1: Let $T = \{t \mid \beta_S(a,t) < \epsilon, 0 \leqq t \leqq k\}$ where $0 \leqq \epsilon \leqq 1$
2: if $|T| = 0$ then
3:     $r_S(a) = 0$ and $\beta_S(a,0) = 1$
4: else
5:     $t_m = \max\{t \mid t \in T\}$
6:     $t_n = \min\{t \mid N_S(a,t) = N_S(a,t_m), t \in T\}$
7:     $r_S(a) = t_n$
8: end if
```

Figure 5: Radius selection algorithm

low, scoped multicast can be considered an efficient delivery method.

Figure 5 describes the radius selection algorithm. The key intuition behind it is to make the radius as large as possible for a given allowable exposure threshold value, ϵ. T is a set of TTL scopes whose associated exposure ratios are less than exposure threshold value ϵ (line 1). If there are no TTL scope values with exposure ratios that are less than ϵ(line 2), a node sets its radius and exposure ratio to 0 and 1 respectively (line 3). Line 5 indicates that each node chooses as large a radius as possible among T. As described earlier, there is a tradeoff between high exposure and underutilization of scoped multicast when selecting radius. Thus, we wish to avoid large radius due to high exposures. However, since we fix the given allowable exposure threshold ϵ, our goal is to make radius large to utilize the scoped multicast. Line 6 tries to reduce the radius if there are unnecessary bandwidth wastes where if there exists a smaller TTL value that can cover the same subgroup members (resulting in the same exposure ratio). For example, consider two cases where a node c chooses its radius as 2 and 4 respectively in Figure 1. Even though the exposure ratios of both cases are the same, with the selection of a larger radius i.e, 4, packets will traverse more links than with a radius of 2. That is, the goal of the radius selection algorithm is to minimize bandwidth waste while satisfying the exposure threshold constraint. For example, Figure 4 shows that we set exposure ratio threshold to be 0.4. Then, t_m and t_n are 4 and 2 respectively, and thus, c decides 2 as its radius.

Once a radius is determined, each node, a, advertises its exposure ratio, $\beta_S(a, r_S(a))$ and radius, $r_S(a)$, only to neighbors within its radius, i.e., $N(a,r_S(a))$. This can be done by multicast with a scope of its radius. Figure 6 shows a head selection algorithm for a node a. Among nodes whose exposure ratios and radii are advertised to a, a head node is selected based on three quantities: (1) exposure ratio, (2) radius, and (3) ID of a node. (We consider ID of a node

```
1: $X = \{n \mid a \in N(n, r_S(n))\}$
2: if $|X| \neq 0$ then
3:     $Y = \underset{n \in X}{\text{argmin}}\ \beta_S(n, r_S(n))$
4:     $Z = \underset{n \in Y}{\text{argmin}}\ r_S(n)$
5:     $head(a) = \underset{n \in Z}{\text{argmin}}\ ID(n)$
6: else
7:     $head(a) = a$
8: end if
```

Figure 6: Head node selection algorithm

is uniquely assigned, e.g., a IP address of a node.) Let us define a *weight vector* of a, $w_S(a)$, as a 3-tuple including the exposure ratio, radius and node ID, i.e., $< \beta_S(a, r_S(a))$, $r_S(a)$, $ID(a) >$, and make the elements of the weight vector be ordered in a lexicographical manner. Then, the algorithm in Figure 6 implies that the lower the weight vector of a node is , the higher its priority to assume the role of head becomes. The intuition behind this is that the algorithm favors the node with lower exposure ratio for the head since the goal is to minimize the exposure ratios. Note here that there is no cyclic relationship happens in the selection of head node. This is because with weight vectors, there is a total ordering among members (since ID of node will be uniquely assigned). This idea is similar to the algorithm for organizing mobile nodes into clusters proposed in [15].

Thus, after gathering neighbors' weight vectors, each node nominates the one with the lowest weight vector to be its head and notifies its head of its decision. Once a node is elected by at least one normal member, the node becomes a head member. A node can nominate itself if there is no other node with lower weight vector. This includes the case where a node is so far away from other nodes that other nodes' weight vectors are unavailable (line 7). Note that the algorithm may generate overlapping islands.

3.3 Connecting islands

Once each member decides its role, the head members, H_S, are responsible for connecting islands, i.e., building an overlay structure consisting of unicast connections among head nodes. This problem is similar to recent research being conducted on application level multicasting [10], [11], [12], [13], [14]. The overlay structure among head members should be built considering target application requirements (e.g., delay or bandwidth needs). Our goal in connecting islands is to minimize duplicate packets traversed at the same link, i.e., minimizing the number of links traversed by packets necessary for communicating among head members. Then, connecting island problem can be formulated as a variant of Steiner tree problem as follows.

Steiner tree problem is to find a minimum cost spanning tree for a subset of nodes in a graph. It can be formulated as follows: Given a graph $K(V, E)$ (where V is a set of nodes and E is a set of edges) a nonnegative weight for each $e \in E$, and a subset $X \in V$, find a subnetwork Y of K such that there is a path between every pair of nodes in X, and the total cost of Y is a minimum. It is well-known that the Steiner tree problem is NP-complete [16]. Note that this problem can be considered to find a minimum cost multicast tree Y for a group X. In our problem setting, we set a graph to be the global multicast tree G, X to be H_S, and the cost for each edge to be 1. Then, the optimal solution to the Steiner tree problem is simply the tree induced by H_S.

If communication among nodes in H_S is restricted to unicast, then we have the following problem.

Problem 1. *Given a tree G, and a set of end nodes H_S, build a* **logical complete** *graph for H_S where the cost of each logical link between nodes in H_S is a distance between them in G. Given a logical complete graph for H_S, find a minimum spanning tree (MST).*

For example, Figure 7 shows a logical complete graph for H_S and its MST for Figure 1 where the number beside a logical link between nodes represents distance (number of hop) between them.

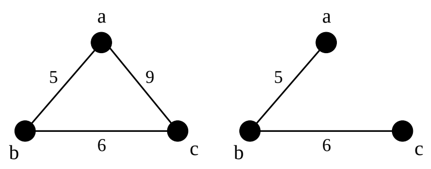


Figure 7: A logical complete graph for H_S and its MST for Figure 1.

In Problem 1, we assume that underlying graph is a global multicast tree G. However, in reality, unicast may not follow paths in G and cause some error in logical link costs. This problem can be solved once we know the "real" unicast cost and set it to be the cost of the logical link. However, in practice, collecting those information requires to send probing packets among members and incurs additional overhead. In contrast, the distance information in G is easy to get due to subgroup advertisement packets. Besides, as pointed in [17], the distance in a multicast tree does not deviate much from unicast distance. Below, we first describe our approach to build overlay structure in a distributed manner, and then show that the structure is the solution of the Problem 1, i.e., minimum spanning tree for a logical complete graph for head members.

Our solution is to build filial relationships among head members. Each head member finds its parent head member. If a head node nominates itself as a parent node, it becomes a *root* head node for the subgroup, S. This strategy, i.e., finding its own parent, guarantees that every head member participates in constructing the overlay structure.

Suppose a *reference* node, $r \in G$, periodically sends *reference* packets to the entire session. The reference node is not dependent on any subgroup, but serves the entire set of subgroups. Practically, reference node can be any node in the tree, e.g., the source node initiating the entire session G. Note that the overlay structure depends on the location of the reference node. However, there will be only one reference node and once the reference node is fixed, constructing efficient TSC structure will follow the same approach described in the paper. When receiving reference packets, each member can calculate the distance between itself and the reference node. In order to obtain distance information, packets will need to carry addition piece of information, Initial_TTL, corresponding to the intial values of the TTL fields. Clearly with this information in hand, a receiver can immediately compute its TTL distance from the source [18]. This distance information is used to build parent-child relationships among head members in the session.

```
1: $C = \{n \in H_S \mid d(n,r) < d(h,r)\}$.
2: if $|C| \neq 0$ then
3:     $D = \underset{n \in C}{\text{argmin}}\ d(n,h)$
4:     $p(h) = \underset{n \in D}{\text{argmin}}\ ID(n)$
5: else
6:     $E = \{n \in H_S \mid d(n,r) = d(h,r),\ ID(n) < ID(h)\}$.
7:     if $|E| \neq 0$ then
8:         $F = \underset{n \in E}{\text{argmin}}\ d(n,h)$
9:         $p(h) = \underset{n \in F}{\text{argmin}}\ ID(n)$
10:    else
11:        $p(h) = h$
12:    end if
13: end if
```

Figure 8: Parent node selection algorithm

Figure 8 describes a parent selection algorithm for a node $h \in H_S$. The parent node, $p(h)$ of a node h, is selected based on the three elements: (1) distance from the reference to a node, (2) distance from the node to h, and (3) ID of the node. As with weight vector in Section 3.2, if a node with smaller elements in the lexicographical order, it has higher chance to become a parent of h.

Figure 9 depicts an example of the parent selection process. In Figure 9, the number in parenthesis is the ID of the node and the arrows represent child-parent relationship. Nodes d, e select their parents according to Line 4. (For d, $D = \{a, b, c\}$, and for e, $D = \{d\}$.) Note that nodes b, c select their parents according to Line 9, and node a becomes root according to Line 11.

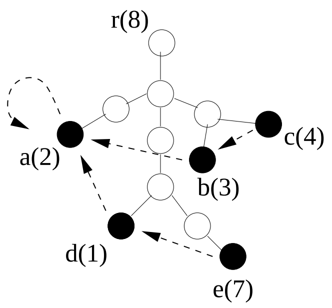


Figure 9: An example of parent selection algorithm

Lemma 1. *The proposed parent selection algorithm builds the minimum spanning tree which is a solution to Problem 1.*

Proof. Parent selection algorithm is a distributed version of PRIM's algorithm. PRIM's algorithm builds upon a single partial minimum spanning tree, at each step adding an edge connecting the vertex nearest to but not already in the current partial minimum spanning tree. In parenet selection algorithm, first vertex to form a MST is a root (near the refrence node). Also, the condition that a child tries to find the nearest parent in our algorithm corresponds to that the partial MST adds an edge connecting the vertex nearest to the current partial MST. □

Note that there is no loop in the overlay structure since the algorithm creates a tree.

The only required information in the above algorithm are TTL distances among head members. Thus, each head member, $h \in H_S$, puts two additional pieces of information in its subgroup advertisement packets: (1) $d(h, r)$ and (2) the fact that it is a head node. However, in the case where its parent head node may be further than k (the scope of subgroup advertisement packets) hops away, an expanding ring search [19] can be used to find the parent head node. The basic idea of expanding ring search is to increase TTL one by one, but this may incur too much overhead to find the parent. To reduce such overhead, one may choose TTL incremental value which is larger than 1, and use the entire G session to solicit the parent node after some trials. Once each node, h, sends a *parent nomination* packet, then its parent node, $p(h)$, sends back a *parent confirmation* packet, which sets up a filial relationship. Finally, note that each node does not have to know all of nodes (C in line 1 in Figure 8) whose distance to the reference node is smaller than itself in practice. This is because a node will eventually choose the one which is the closest to itself among C.

3.4 Forwarding over and maintenance of TSC structures

In this section, we describe how to communicate among subgroup members over TSC structures and maintain those structures. Once a TSC structure is constructed, communication among subgroup members can be done by broadcast over the TSC structure.

While each node, $a \in S$, sets up a relationship among members in S, it needs to build a routing table, $T_S(a)$ for subgroup S communication. A normal node simply maintains an entry for its head node. A head node stores its radius and entries of nodes which have filial relationships, i.e., one parent and its children, if any. The radius entry in the routing table represents a scoped multicast in the G session with the TTL scope of the radius. Then the following two rules suffice for subgroup S communication: (1) if a node, a, is a source node, broadcast packets over entries in $T_S(a)$ and (2) if a node is a relay node, broadcast packets over entries in $T_S(a)$ except the one from which packets are received.

Note that during the multicast session, the interests of members may change and members may leave or join the global multicast session. Such dynamics are handled by periodic subgroup advertisement packets injected by each member. A change of interest or membership will produce different TTL-neighbor profile, leading to a change in the weight vectors. If a normal node wishes to change its head node, it notifies the previous head node of its intention, so that the head which no longer has any normal members, can become a normal member. When a head node leaves or becomes a normal node, it notifies its parent and children of the event, so that they update their routing tables and find other parents. Consider the case where nodes abruptly fail or the network is partitioned, wherein explicit notification is impossible. In this case, periodic subgroup advertisement packets indicate the liveness of members, thus our mechanism can dynamically adapt to the situation. However, since there is a scope limit to subgroup advertisement packets (k in Section 3.2), parent and child nodes whose distance is farther than k, need to periodically exchange acknowledgment packets.

4. PERFORMANCE EVALUATION

We conduct simulations to study various issues and tradeoffs in applying the proposed TSC mechanism in multicast applications. Our main goal is to investigate in which environments it is advantageous to apply the proposed mechanism. For comparison, we examine the performance of the following schemes for subgroup communication.

• *Global Multicast:* This represents a scheme that simply uses the original global multicast group G for subgroup communications.

• *Unicast-Only:* This scheme constructs unicast overlay trees among subgroup members. Though there have been numerous overlay schemes presented, we use our methods for constructing overlay structures. That is, this scheme can be considered as a TSC mechanism with 0 exposure ratio.

• *TSC-ϵ:* This represents our proposed scheme with an ϵ exposure threshold.

4.1 Metrics and Methodologies

To evaluate our TSC mechanism, we use the following metrics.

• *Cost ratio*: Let us define the *cost* of a subgroup communication using scheme f by $C_f(S)$, the total number of links traversed by packets generated to distribute a unit amount of data for S subgroup communication. For simplicity, we assume that a link cost is symmetric and unit cost. However, the cost can be generalized with inclusion of asymmetric and variable link costs. For example, in the case where a new multicast session is created, the cost of subgroup S communication, denoted by $C_{new}(S)$, is simply the number of links in the tree induced by subgroup members. When we ignore the overhead for maintaining multicast session, $C_{new}(S)$ can be considered as an optimal cost since no multiple packets will traverse in links which minimizes the cost. Thus, we define a *cost ratio* γ_f as the ratio of $C_f(S)$ to $C_{new}(S)$, i.e., $\gamma_f = C_f(S)/C_{new}(S)$. A value close to 1 for the cost ratio metric represents an efficient use of bandwidth. The higher cost ratio represnets the scheme that requires more cost. For example, in Figure 1, $C_{new}(S) = 13$, and the cost involved for TSC mechanism is 17 coming from three components : (1) 5 from a to b, (2) 6 from b to c, and (3) 6 from scoped multicast from c to other members.

• *Global member exposure ratio*: Let $E_f(S)$ be a set of members in G exposed while applying a scheme f for subgroup communication among a set of users. The *global member exposure ratio*, β_f, is defined as $\frac{|E_f(S)\backslash S|}{|E_f(S)|}$. The higher the value of β_f, the more members are exposed. Note that this is different from the local view of exposure ratio previously defined. This global member exposure ratio is used as a metric to evaluate TSC mechanisms and each individual node cannot have this information.

We measure the two above metrics by varying the following elements.

• *Topologies:* We use real multicast trees gathered in [17]. Note that unicast packets will follow the same path taken by multicast packets in our simulation environment, which may not be the case in real world. However, as shown in [17] there is a topological closeness between unicast paths and multicast paths. Thus, we believe that the performance results are valid.

• *Subgroup density:* Subgroup density is the portion of subgroup members compared to the total number of members in G. It is measured in percentage (%). We vary density of subgroup members from sparse, to mid-range, to dense.

• *Subgroup membership distribution:* We follow the same methodology as proposed in [3] to model topological correlation within a subgroup, i.e., as with random, affinity/disaffinity or distributed clusters. Affinity mode emulates subgroup distributions with members that tend to cluster together and the disaffinity mode is for the subgroup member distribution that tends to be spread out.

We create a subgroup S with m nodes for affinity and disaffinity modes as follows: initially S has no members and we choose subgroup members one by one from the global session G until $|S| = m$. The first node is randomly selected. For kth node selection, we assign a probability $p_i = \frac{\alpha}{g_i^\theta}$ to each node $n_i \in G \backslash S$, where $g_i = \min_{n_j \in S} d(n_i, n_j)$ and α is calculated such that $\sum_{n_i \in G \backslash S} p_i = 1$. Then, we randomly select a subgroup member among $C = \{n_i \mid n_i \in G \setminus S, p_i \geq p\}$ where p is a random value from 0 to 1. If $|C| = 0$, then choose different p value. We use $\theta = 15$ and $\theta = -15$ for affinity and disaffinity respectively.

In the distributed clusters mode, a few clusters are randomly scattered in the tree and each cluster is modeled according to the affinity mode. We modeled a number of clusters that was linearly increasing as a density of subgroup increases, i.e., $0.3 * density + 2$.

• *Scope of subgroup advertisement packet:* We vary scope of subgroup advertisement packet to investigate its impact on the performance of our TSC mechanism.

4.2 Performance Results

Since our results for the various topologies in [17] show similar trends, we only present results for the real multicast tree shown in Figure 10. It consists of 2359 nodes and 1487 end nodes(members).

The following figures show the cost ratio and member exposure ratio results varying subgroup densities for the various node distributions. We set 7 as the TTL scope for subgroup advertisement packets. In each figure, we present performance metrics for global multicast, unicast-only, TSC-0.2 and TSC-0.6 schemes. Each point in the figures represents an average over 100 different subgroup distributions for given distribution mode and density. We do not include the member exposure ratio of the unicast-only scheme since it is always 0.

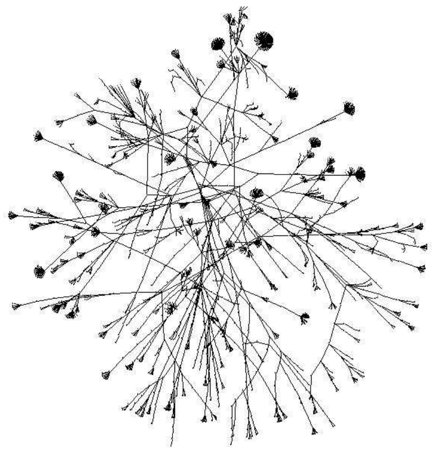
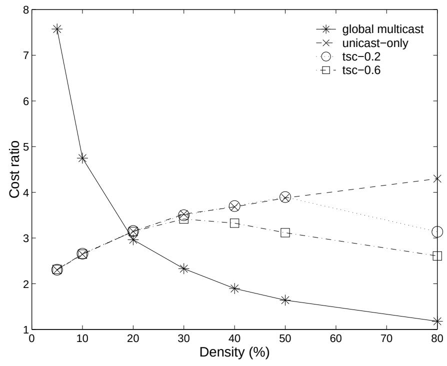


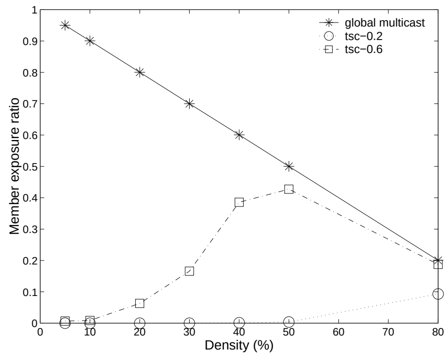
Figure 10: A global multicast tree topology

Figure 11 shows the results for random node distributions. We observe that the cost ratio of global multicast scheme heavily depends on the density of subgroups: the larger the density of a subgroup is, the lower cost ratio of global multicast is; above 20% density global multicast beat all the other schemes in terms of the cost ratio. However, the member exposure ratio of global multicast scheme is higher than other schemes for all densities. Also note that the member exposure ratio of global multicast is linearly decreasing as the density of subgroup increase. For low density regimes, we observe that unicast-only and TSC schemes show similar results. This is because members are randomly distributed and the density is low, TSC generates one-member islands in most cases. As subgroup density increases, TSC outperforms unicast-only by using scoped multicast. TSC-0.6 achieves better cost ratio than TSC-0.2 as density increases since TSC-0.6 scheme aggressively forms non-one member islands. However, better cost ratio performance is at the expense of more member exposure ratio as shown in Figure 11 (b).

Figure 12 shows the performance results for subgroups with nodes placed based on the affinity distribution. Note that since in the affinity mode, members are spatially clustered together, a global multicast scheme causes an excessive cost ratio, e.g., γ= 23 at 5% density. The cost ratio of TSC mechanism is almost two times lower than unicast-only scheme for the range of densities. TSC-0.2 and TSC-0.6 have almost the same cost ratio results since members are clustered so that the exposure threshold value is not a major factor for creating islands any more. Also note that TSC mechanisms achieve fairly low member exposure ratios.

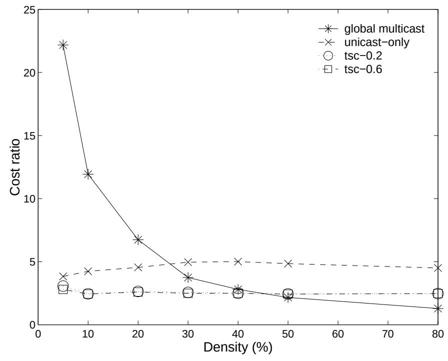


(a) Cost ratio

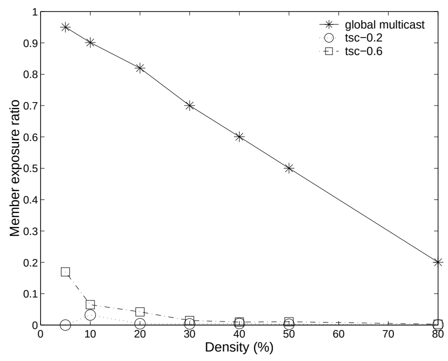


(b) Member exposure ratio

Figure 11: Random mode



(a) Cost ratio



(b) Member exposure ratio

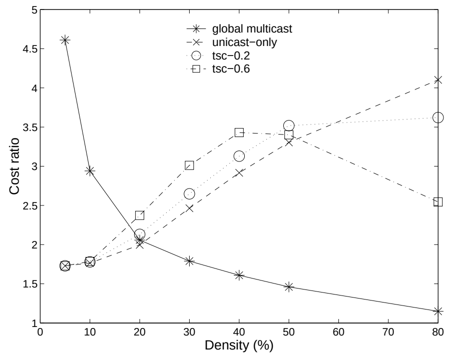
Figure 12: Affinity mode

Figure 13 depicts the results for the disaffinity distribution mode. Both cost ratio and member exposure ratio results show similar trends to those for the random case except that in the mid-range of densities, the cost ratio of TSC mechanisms is slightly higher than that of the unicast-only scheme. This can be explained since members are spread out from each other in a disaffinity mode, the effort to form islands in a TSC mechanism leads to more link exposure by scoped multicasts. However, for high densities, the cost ratio eventually benefits TSC mechanisms.

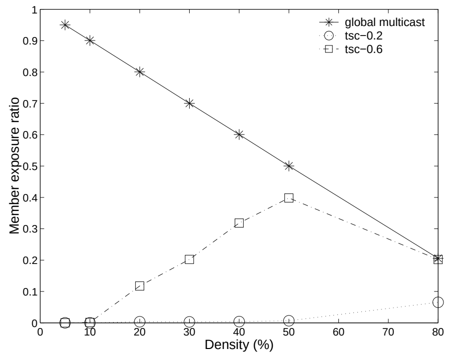
Figure 14 shows the performance results for the distributed clusters distribution mode. We observe that the results are similar to the affinity mode.

Figures 15, 16, 17 and 18 show the performance results for random, affinity, diaffinity and distributed clusters respectively varying the scope (k) of subgroup advertisement packets, i.e., $k = 2, 4, 7, 10$ with TSC-0.6 scheme. Note that the scope k provides a hard limit for the radius of islands, i.e., the radius cannot be larger than k. We observed that all four distribution modes show similar results for varying scopes of subgroup advertisement packet as follows. First, as the scope becomes smaller, member exposure ratio decreases. This is intuitive since a smaller scope does not allow large islands, which can reduce member exposure. At the extreme case where $k = 0$, the member exposure ratio is 0. Second, $k = 4$ is the best choice for the cost ratio metric for all distribution modes. Though smaller scopes generate smaller member exposure ratio, it may underutilize scoped multicast to reduce the cost ratio. Also, if TTL scopes are too large, they generate high cost ratios due to large islands. Thus, an intermediate scope value can produce the lowest scope value, which is $k = 4$ for our simulation results. Third, even a small scope can generate pretty low cost ratio for all distribution modes. For example, $k = 2$ achieves slightly higher cost ratio compared to $k = 4$. This result demonstrates that most benefit from scoped multicast can be achieved with even small scopes. This is an encouraging result since the overhead for control messages for TSC mechanism can be significantly reduced by using subgroup advertisement packets with small scopes.

Through the simulation studies, we observed that different subgroup membership distributions and varying subgroup densities heavily influence the performance of the schemes for subgroup communication. As expected, the TSC mechanism benefits greatly from clustered distributions (affinity and distributed clusters modes). The TSC mechanism also achieves fairly stable cost ratios (from 1.5 to 4) irrespective of variations in density or distribution modes. This feature

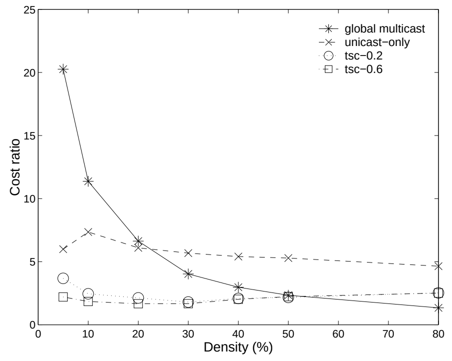


(a) Cost ratio

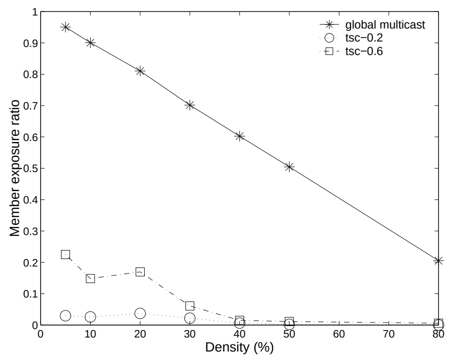


(b) Member exposure ratio

Figure 13: Disaffinity mode



(a) Cost ratio



(b) Member exposure ratio

Figure 14: Distributed clusters mode

may be helpful in the situation where the information about the density and distribution characteristic of subgroups is unavailable. Also note that one can make a trade-off between the cost ratio and member exposure ratio by varying exposure thresholds.

4.3 Signal cost

As seen above, TSC mechanism can achieve small cost ratios and small member exposure ratios for many environments even without creating any new multicast group. However, this benefit comes with the price : signal overhead to construct TSC fowarding structure. To understand the impact of varoius environments on the signal overhead, we implemented the proposed TSC mechanism in ns-2 simulator. Due to the scalability issue of ns-2 simulator, 500 end-node topology [17] is used to have the below result.

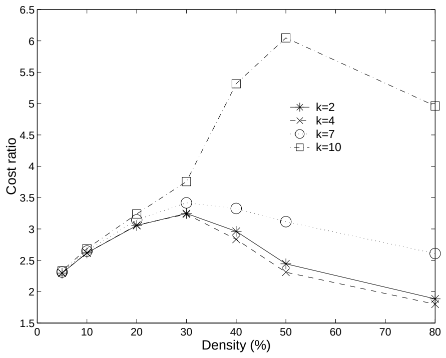
First, we define the *signal cost* as the total number of links traversed by signal packets to construct a TSC structure. Since comparing signal cost of TSC mechanism with other approach, e.g., generating new multicast groups (signal overhead incurred to maintain multicast trees) is not clear, we compare the signal cost of TSC mechanisms with that of unicast-only mechanism (pure end-to-end approach). Thus, we define *signal cost ratio* of scheme *f* as the ratio of the signal cost of scheme *f* to the signal cost of unicast-only mechanism.

Figure 19 depicts the signal cost ratio results for TSC-0.6 with k = 6, while varying densities. First, all signal cost ratios are less than 1, which means that the signal cost of TSC scheme is less than that of unicast-only scheme regardless of subgroup membership distribution modes and subgroup densities. Second, as subgroup density increase, we observes that the signal cost ratio becomes smaller, which implies that much less signal packets are required compared to that of unicast-only mechanism. Finally, note that affinity and distributed cluster modes have much smaller signal cost ratios than those of the random and disaffinity modes. This again indicates that the TSC mechanism benefits greatly from clustered distributions (affinity and distributed clusters modes) from not only cost but also signal cost perspectives.

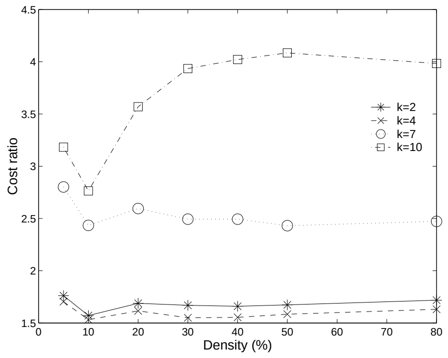
For different exposure thresholds, we obtained the similar results and omit them. We also redo the whole experiments in Section 4.2 using our ns-2 implementation and obtained the consistent results as with ones presented in Section 4.2.

5. CONCLUSIONS

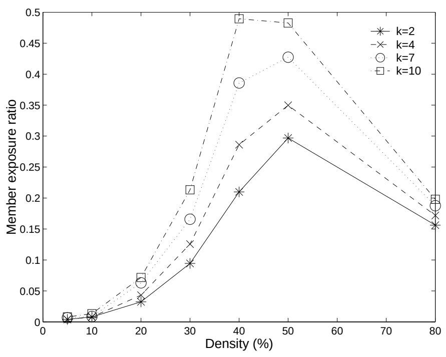
In this paper, we designed and evaluated a topology-sensitive subgroup communication mechanism to handle the preference heterogeneity problem in large-scale multicast appli-



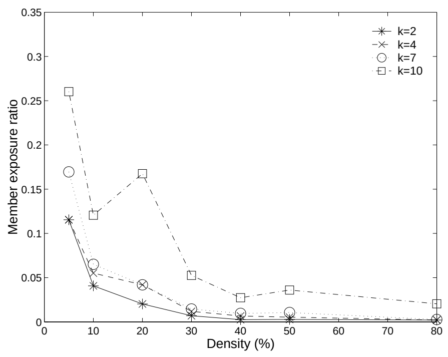
(a) Cost ratio



(a) Cost ratio



(b) Member exposure ratio



(b) Member exposure ratio

Figure 15: Random mode (TSC-0.6)

Figure 16: Affinity mode (TSC-0.6)

cations. Our TSC mechanism takes a complete end-to-end approach which eliminates additional creation of multicast groups. Depending on the local density of subgroup members, members in the session self-configure into islands and forwarding structures. Within islands, scoped multicast is used to derive benefit from clustered membership distribution and between islands, unicast is used to reduce unnecessary exposure. To construct a unicast overlay structure, we also propose a simple algorithm based on TTL distance, which creates a mimimum spanning tree among nodes.

Throughout our simulations, we observe that our TSC mechanism performs in a consistent way over diverse densities and distribution modes of the subgroup.

6. ACKNOWLEDGMENTS

We would like to thank Daniel Ng for help on simulation in ns-2. This work is supported by the National Science Foundation Grant CNS-0509355.

7. REFERENCES

[1] T. Wong, R. Katz, and S. McCanne, "An evaluation of preference clustering in large-scale multicast applications," in *Proc. IEEE Infocom*, 2000.

[2] D. Thaler and M. Handley, "On the aggregatability of multicast forwarding state," in *Proc. IEEE Infocom*, 2000.

[3] T. Wong and R. Katz, "An analysis of multicast forwarding state scalability," in *International Conference on Network Protocols(ICNP)*, 2000.

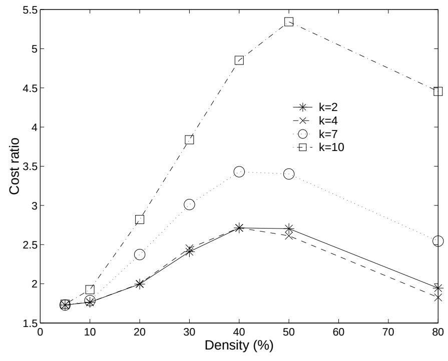
[4] E. Lety and T. Turletti, "Issues in designing a communication architecture for large-scale virtual environments.," in *Proc. of Networked Group Communication Workshop*, 1999.

[5] D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering, M. Handley, V. Jacobson, C. Liu, P. Sharma, and L. Wei, "Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification," RFC 2362, Jun. 1998.

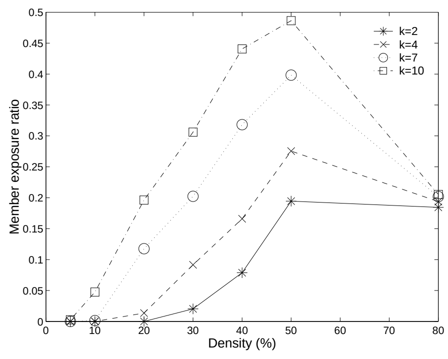
[6] D. Thaler, D. Estrin, and D. Meyer, "Border gateway multicast protocol (BGMP):Protocol specification," IETF draft, draft-ietf-bgmp-spec-01.txt, Mar. 2000.

[7] T. Ballardie, P. Francis, and J. Crowcroft, "Core based trees(CBT): An architecture for scalable inter-domain multicast routing," in *Proc. ACM SIGCOMM*, 1993.

[8] J. Tian and G. Neufeld, "Forwarding state reduction for sparse mode multicast communication," in *Proc. IEEE Infocom*, 1998.

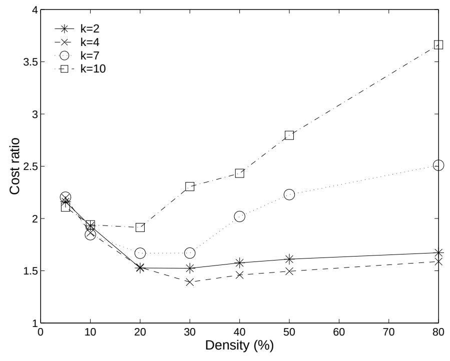


(a) Cost ratio

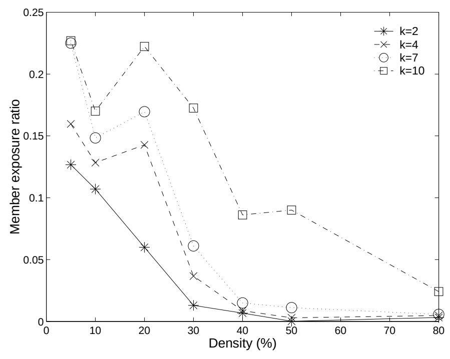


(b) Member exposure ratio

Figure 17: Disaffinity mode (TSC-0.6)



(a) Cost ratio



(b) Member exposure ratio

Figure 18: Distributed clusters mode (TSC-0.6)

[9] I. Stoica, T. E. Ng, and H. Zhang, "REUNITE: A recursive unicast approach to multicast," in *Proc. IEEE Infocom*, 2000.

[10] P. Francis, "Yoid: Extending the Internet multicast architecture," in *Tech. reports, ACIRI, http://www.aciri.org/yoid*, 2000.

[11] Y. Chawathe, *Scattercast: An architecture for Internet broadcast distribution as an infrastructure service*, Ph.D. thesis, University of California, Berkeley, 2000.

[12] J. Jannotti, D. Gifford, K. Johnson, F. Kasshoek, and J. O'Toole, "Overcast: Reliable multicasting with an overlay network," in *USENIX OSDI*, 2000.

[13] Y. Chu, S. Rao, S. Seshan, and H. Zhang, "Enabling conferencing applications on the Internet using an overlay multicast architecture," in *Proc. ACM SIGCOMM*, 2001.

[14] D. Pendarakis, S. Shi, D. Verma, and M. Waldvogel, "ALMI: an application level multicast infrastructure," in *3rd USENIX Symposium on Internet Technologies and Systems (USITS)*, 2001.

[15] J. Gao, L. J. Guibas, J. Hershberger, L. Zhang, and A. Zhu, "Discrete mobile centers," in *17th Symposium on Computational Geometry(SoCG)*, 2001.

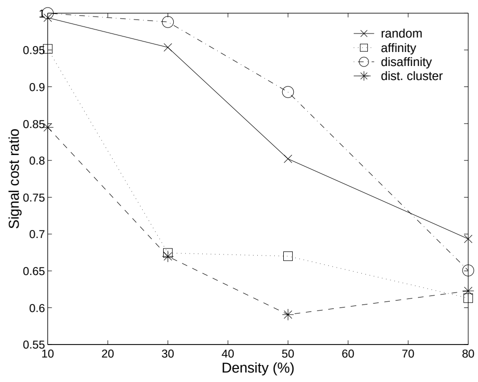


Figure 19: Signal cost ratios for TSC-0.6 with k = 6

[16] M. Garey and D. Johnson, *Computers and Intractability*, Freeman, 1979.

[17] R. C. Chalmers and K. C. Almeroth, "Modeling the branching characteristics and efficiency gains in global multicast trees," in *Proc. IEEE Infocom*, 2001.

[18] J. Lee and G. de Veciana, "Resource and topology discovery for ip multicast using a fan-out decrement

mechanism," in *Proc. IEEE Infocom*, 2001.

[19] S. Keshav, *An Engineering approach to computer networking:ATM Networks, the Internet, and the Telephone Network*, Addison-Wesley, Inc., 1997.