An Evaluation of Topology-sensitive Subgroup Communication Mechanism in Large-Scale Multicast Applications*

Jangwon Lee, Gustavo de Veciana
Department of Electrical and Computer Engineering
University of Texas at Austin
Austin, Texas, 78712
(jangwlee, gustavo)@ece.utexas.edu

ABSTRACT

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 propose a topology-sensitive subgroup communication (TSC) mechanism to support efficient subgroup communications in large-scale multicast applications. Our mechanism takes an end-to-end approach which can completely eliminate additional creation of multicast groups. Our TSC mechanism exploits the spatial locality of members within a given subgroup, and enables members to autonomously build a TSC forwarding structure consisting of multiple unicast and scoped multicast connections. Through simulations, we observe that TSC mechanisms perform well for diverse densities and distributions of nodes in a subgroup.

Categories and Subject Descriptors

C.2 [**Computer Systems Organization**]: Computer Communication Networks

General Terms

Performance

Keywords

Multicast, Subgroup, Topology-sensitive

1. INTRODUCTION

Due to its bandwidth efficiency, IP Multicast is the preferred data delivery method for large-scale interactive applications such as distributed interactive simulations (DIS), video conferencing tools and multi-player games. 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 [13], requiring frequent communication within *subgroups* of members sharing common interests/requirements. 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 [10, 12]. 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 [13] or players' positions in a virtual cell [6].

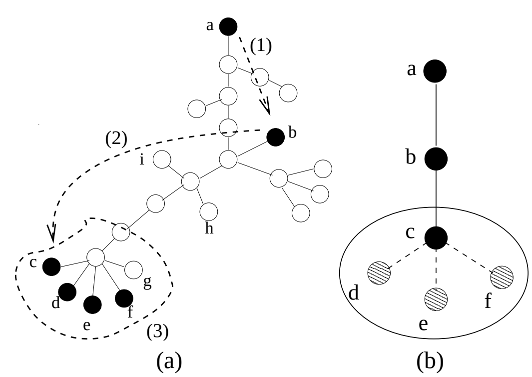


Figure 1: An example of TSC forwarding structure

In this 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 forwarding structure consisting of multiple unicast and scoped mul-

*This work is supported by the National Science Foundation Grant ANI-0081360.

NGC'02 October 23-25, 2002, Boston, Massachusetts, USA.

ticast connections. For example, consider a distribution tree for a multicast session, G, in Figure 1(a). 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(a). 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.

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

2. TSC FORWARDING STRUCTURE

Given G and $S = \{a,\ b,\ c,\ d,\ e,\ f\}$ as shown in Figure 1(a), Figure 1(b) depicts an example of a TSC forwarding structure exhibiting an overlay structure among members in 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* or *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 forwarding 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 islands in Figure 1(b).

Note that if there are only one-member islands in a TSC forwarding structure, this 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 reduce such bandwidth wastes in the case where subgroup members are clustered with each other. Thus, our goal is to build a TSC forwarding structure which minimizes wasted bandwidth while limiting the exposure of non-subgroup members given a multicast session G and subgroup preferences for each member in G. Building TSC structures involves two stages: constructing islands and then connecting islands.

2.1 Constructing islands

Let the set $N(a,t)$ be the set of neighbors of a, i.e., nodes within a TTL distance of t of a node $a \in G$, excluding a itself. 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., $\{n | n \in S \cap N(a,t)\}$. 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]

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

Before proceeding, we briefly describe how each member can compute an exposure ratio for a given TTL scope. 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.[3] Figure 2 shows an example of the TTL-neighbor profile of member c in Figure 1 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.

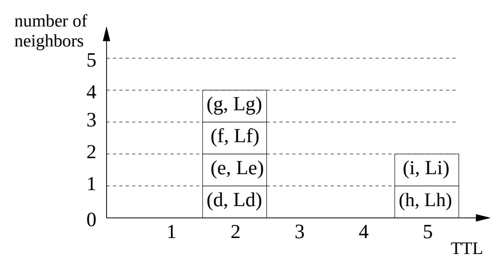


Figure 2: TTL-neighbor profile of c in Figure 1

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

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. Note that constructing islands requires (1) each member in S to decide its role between head and normal, (2) a normal node to decide its head node, and (3) a head node, $h \in H_S$, to decide its TTL scope, called *radius*, $r_S(h)$. Note 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. To make a trade-off, we introduce an *exposure threshold* ε to control the degree of exposure. Thus, the goal is to make the radius as large as possible for a given allowable exposure threshold. Since each node, $a \in S$, initially considers itself to be a head candidate, each node computes its radius as shown in Figure 3.

[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, 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) [9] currently under development, constructs bidirectional shared trees.

[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.

[3]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.

1: Let $T = \{t | \beta_S(a,t) < \varepsilon, 0 \leqq t \leqq k\}$ where $0 \leqq \varepsilon \leqq 1$
2: **if** $|T| = 0$ **then**
3: $r_S(a) = 0$ and $\beta_S(a,0) = 1$
4: **else**
5: $t_m = \max\{t | t \in T\}$
6: **if** there exists $t_n \in T$ such that $N_S(a,t_n - 1) \neq N_S(a,t_n) = N_S(a,t_n + 1) = \ldots = N_S(a,t_m - 1) = N_S(a,t_m)$ **then**
7: $r_S(a) = t_n$
8: **else**
9: $r_S(a) = t_m$
10: **end if**
11: **end if**

Figure 3: Radius selection algorithm

If there are no TTL scope values with exposure ratios that are less than ε for a node a (line 2), a 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 given an exposure threshold. Line 6 and 7 try to reduce the radius if there are unnecessary bandwidth wastes. For example, consider two cases where a node c chooses its radius as 2 or 4 respectively in Figure 1 (a). 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. Figure 4 shows an example of the radius decision rule for node c in Figure 1 where an exposure threshold, ε, is 0.4. In this example, t_m and t_n are 4 and 2 respectively.

A *weight vector* of a, $w_S(a)$, is defined as a 3-tuple including the exposure ratio, radius and node ID, i.e., $< \beta_S(a, r_S(a))$, $r_S(a)$, $ID(a) >$. An IP address can be used as an ID of a node. The elements of the weight vector are ordered in a lexicographical manner. That is, a weight vector for a node a is *less* than that of a node b, $w_S(a) < w_S(b)$, if
1) $\beta_S(a, r_S(a)) < \beta_S(b, r_S(b))$ or
2) $\beta_S(a, r_S(a)) = \beta_S(b, r_S(b))$ and $r_S(a) < r_S(b)$ or
3) $\beta_S(a, r_S(a)) = \beta_S(b, r_S(b))$ and $r_S(a) = r_S(b)$ and $ID(a) < ID(b)$.

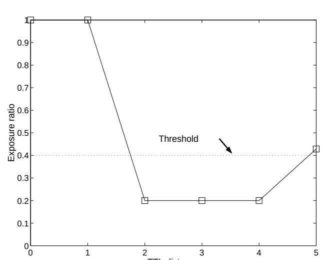


Figure 4: Radius selection of c in Figure 1

Once each member, a, has a weight vector, the choice of a head node is based on the weight associated with each node: the lower the weight of a node, the higher its priority to assume the role of head. This idea is similar to the algorithm for organizing mobile nodes into clusters proposed in [3]. Each node, a, advertises its weight vector only to $N(a, r_S(a))$, i.e., performs a scoped multicast with scope $r_S(a)$. 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, a, is elected by at least one normal member, a becomes a head member. Note that a node can nominate itself if there is no other node with lower weight vector.[4] Also note that this algorithm may generate overlapping islands.

1: $C = \{n \in H_S | d(n,r) < d(h,r)\}$ where $d(h,r)$ denotes TTL distance between h and 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: $\bar{C} = \{n \in H_S | d(n,r) = d(h,r)\}$.
7: $p(h) = \underset{n \in \bar{C}}{\text{argmin}}\, ID(n)$
8: **end if**

Figure 5: Parent head node selection algorithm

2.2 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 [2, 4, 7]. Our solution is to build filial relationships among head members based on TTL distance information. 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.[5] When receiving reference packets, each member can obtain the distance between itself and the node r. This distance information is used to build parent-child relationships among head members in the session. The parent node, $p(h)$ of a node $h \in H_S$, is selected based on the following rules: Choose the one with the closer distance to r than that of itself. If multiple nodes satisfy this condition, then choose the node which is the closest to itself. If there are again multiple nodes, then choose the one with the lowest ID. Figure 5 and 6 present the detailed algorithm and an example of the parent selection process respectively. In Figure 6, the number in parenthesis is the ID of the node. Note that node c would select its parent in Step 4 of Figure 5 while nodes a, b would select their parents on Step 7.

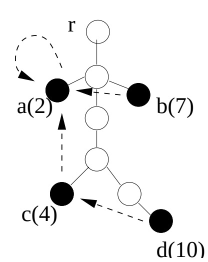


Figure 6: An example of parent selection algorithm

The only required information in the above algorithm are TTL distances among head members. Thus, each head member, $h \in H_S$,

[4] This includes the case where a node is so far away from other nodes that other nodes' weight vectors are unavailable.

[5] Note that the reference node is not dependent on any subgroup.

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 [5] can be used to find the parent head node. 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.

2.3 Forwarding and maintenance of a TSC mechanism

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 2.1), parent and child nodes whose distance is farther than k, need to periodically exchange acknowledgment packets.

3. EVALUATION

We conduct simulations to study various issues and trade-offs in applying the proposed TSC mechanism in multicast applications. Our 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.

3.1 Performance Metrics

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.[6] 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. Since $C_{new}(S)$ can be considered optimal, 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.

• *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 $\frac{|E_f(S)\setminus S|}{|E_f(S)|}$. The higher the value of β_f, the more members are exposed.

3.2 Methodologies

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

• *Topologies:* We use real multicast trees gathered in [1]. 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 [1] there is a topological closeness between unicast paths and multicast paths. Thus, we believe that the performance results are valid.

• *Subgroup density:* We vary density of subgroup members from sparse, to mid-range, to dense.

• *Subgroup membership distribution:* We follow the same methodology as proposed in [12] 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 $n_i \in G \setminus S$, where $g_i = \min_{n_j \in S} d(n_i, n_j)$ and α is calculated such that $\sum_{n_i \in G\setminus S} p_i = 1$. Then, we randomly select a subgroup member among $C = \{n_i | n_i \in G \setminus S, p_i \geq p\}$ where p is a random value from 0 to 1.[7] 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.

3.3 Results

Since our results for the various topologies in [1] show similar trends, we only present results for the real multicast tree shown in Figure 7. 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

[6]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.

[7]If $|C| = 0$, then choose different p value.

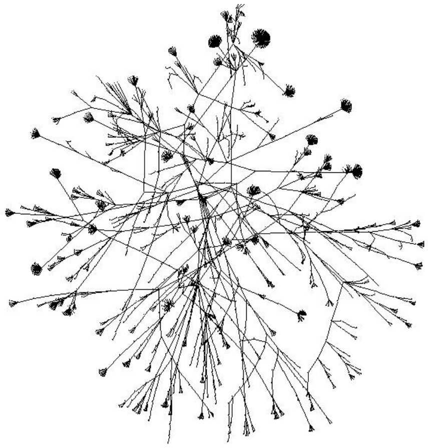
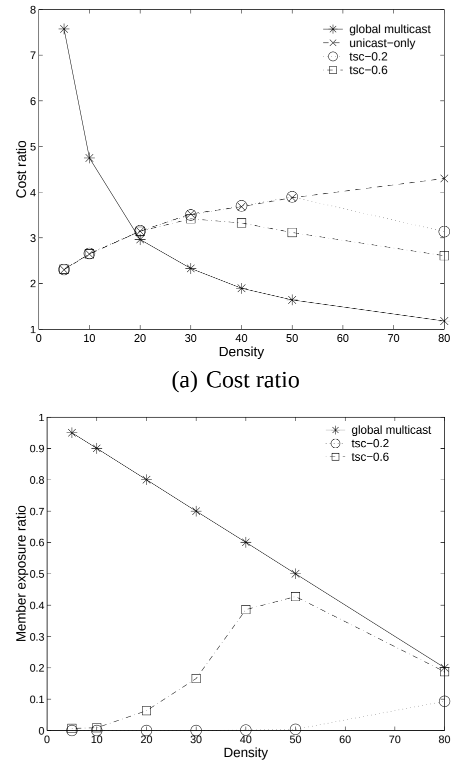


Figure 7: A global multicast tree topology



(a) Cost ratio

(b) Member exposure ratio

Figure 8: Random mode

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 8 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 constant regardless of subgroup members' distributions. 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 8 (b).

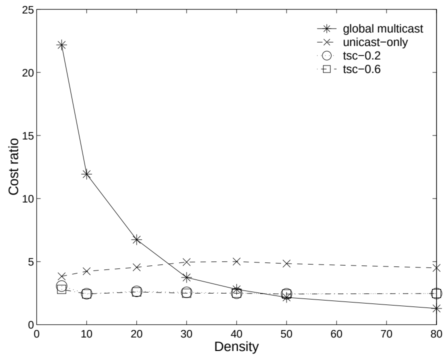
Figure 9 shows the performance results for subgroups with nodes placed based on the affinity distribution. Note that since in the affinity mode, members are spatially clustered together, a global multicast scheme causes an excessive cost ratio, e.g., γ= 23 at 5% density. The cost ratio of TSC mechanism 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.

Figure 10 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.

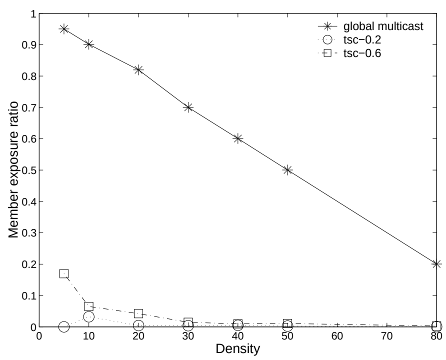
Figure 11 shows the performance results for the distributed clusters distribution mode. We observe that the results are similar to the affinity mode.

Figures 12, 13, 14 and 15 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

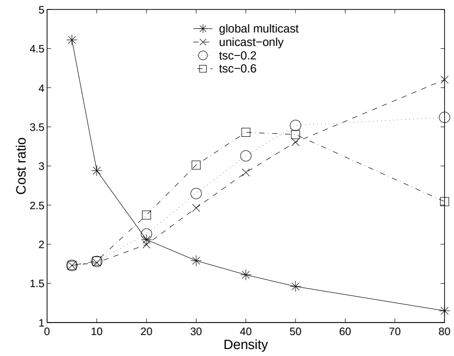


(a) Cost ratio

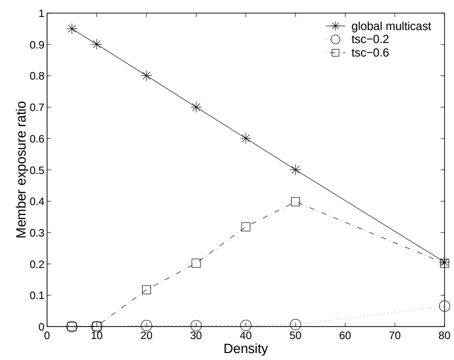


(b) Member exposure ratio

Figure 9: Affinity mode



(a) Cost ratio



(b) Member exposure ratio

Figure 10: Disaffinity mode

4) irrespective of variations in density or distribution modes. This feature 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.

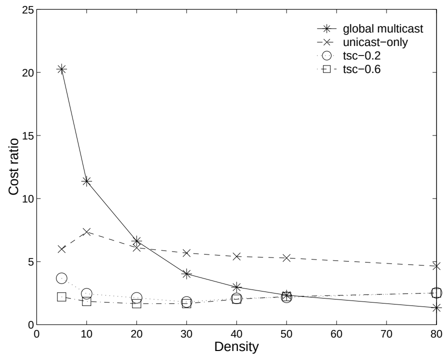
4. RELATED WORK

The scalability of state associated with multicast forwarding by routers has been one of the significant issues for the wide deployment of IP multicast. Reduction of multicast forwarding state at routers can be achieved through aggregation or elimination of non-branching approaches. In [10], 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 [11] and REUNITE [8] 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 [13] or players' position in a virtual cell [6]. 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 requiring lots of subgroup communication, which is our aim in this paper.

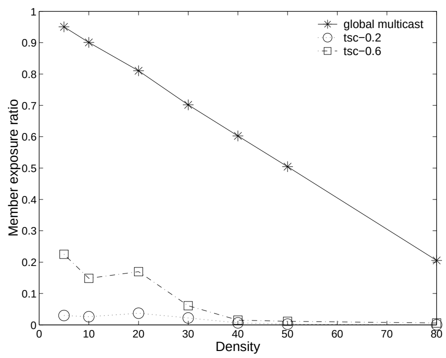
Note that our TSC scheme completely eliminates creation of additional multicast groups and takes a full end-to-end approach for subgroup communication. Unlike existing clustering frameworks, it does not require correlated information among subgroups, which eliminates the need for a central point where the information is collected and grouping decisions are made. Our approach also should be contrasted with a number of recent application-level multicast studies, e.g., [2, 4, 7]. 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 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.

5. CONCLUSIONS AND FUTURE WORK

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



(a) Cost ratio



(b) Member exposure ratio

Figure 11: Distributed clusters mode

exposure. Throughout our simulations, we observe that our TSC mechanism performs in a consistent way over diverse densities and distribution modes of the subgroup. As future work, one might consider a dynamic application of different schemes provided one has on-line information on subgroup density and distribution mode. In addition, other interesting topics would be finding a more refined head member selection algorithm to reduce the overlap among islands and issues concerning signaling overhead.

6. ACKNOWLEDGMENTS

We would like to thank the anonymous NGC'02 reviewers for the helpful feedback.

7. REFERENCES

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

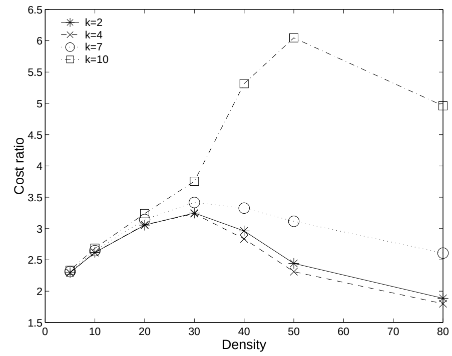
[2] Y.-H. Chu, S. Rao, and H. Zhang. A case for end system multicast. *ACM SIGMETRICS*, 2000.

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

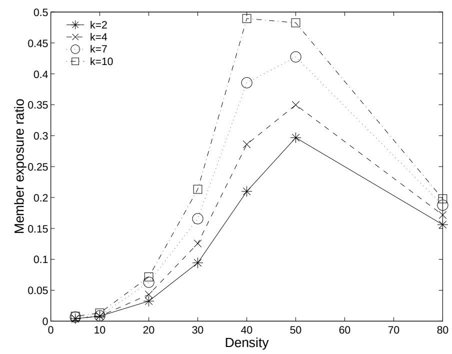
[4] J. Jannotti, D. Gifford, K. Johnson, F. Kasshoek, and J. O'Toole. Overcast: Reliable multicasting with an overlay network. *USENIX OSDI*, 2000.

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

[6] E. Lety and T. Turletti. Issues in designing a communication architecture for large-scale virtual environments. *Proc. of*

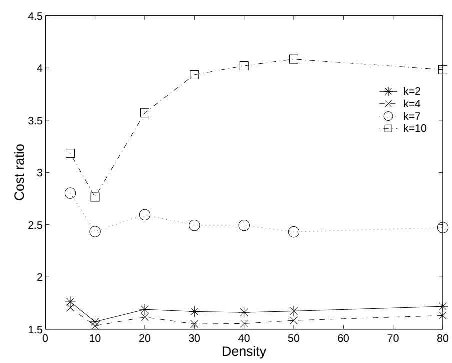


(a) Cost ratio

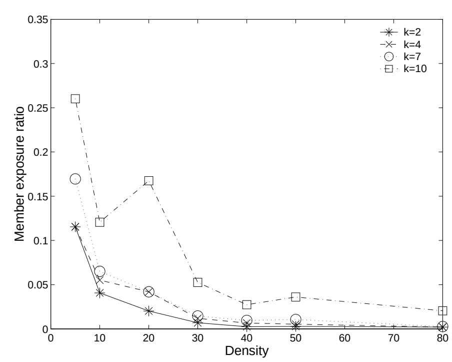


(b) Member exposure ratio

Figure 12: Random mode (TSC-0.6)

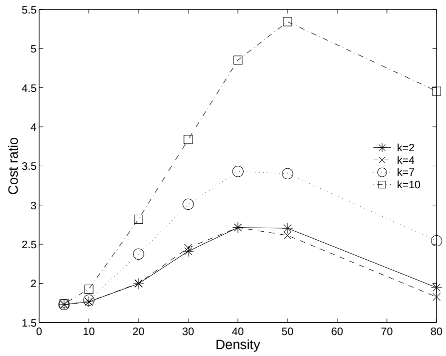


(a) Cost ratio

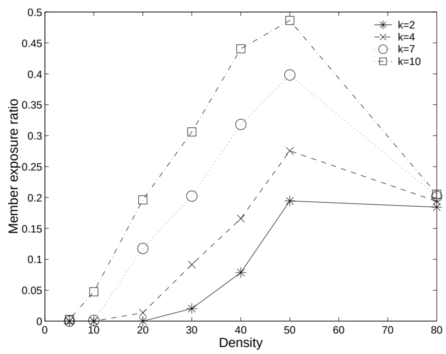


(b) Member exposure ratio

Figure 13: Affinity mode (TSC-0.6)

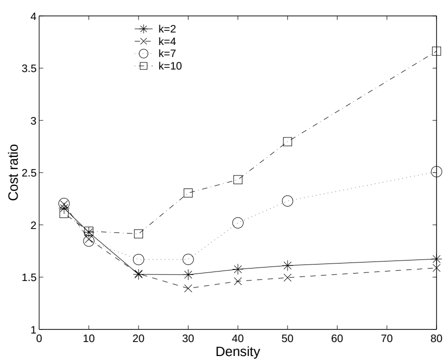


(a) Cost ratio

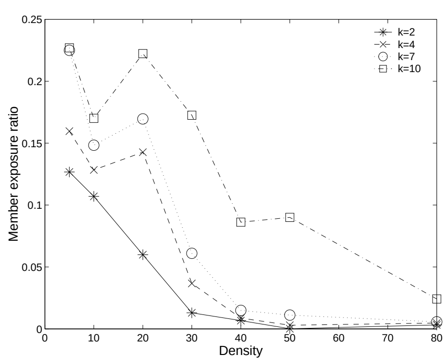


(b) Member exposure ratio

Figure 14: Disaffinity mode (TSC-0.6)



(a) Cost ratio



(b) Member exposure ratio

Figure 15: Distributed clusters mode (TSC-0.6)

Networked Group Communication Workshop, 1999.

[7] D. Pendarakis, S. Shi, D. Verma, and M. Waldvogel. ALMI: an application level multicast infrastructure. *USENIX Symposium on Internet Technologies*, 2001.

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

[9] D. Thaler, D. Estrin, and D. Meyer. Border gateway multicast protocol (BGMP):Protocol specification. Mar. 2000.

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

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

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

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