Multicast Routing in Dense and Sparse Modes: Simulation Study of Tradeoffs and Dynamics *

Liming Wei
Computer Science Department
University of Southern California
Los Angeles, CA 90089-0781
lwei@catarina.usc.edu

Deborah Estrin
Computer Science department / ISI
University of Southern California
Los Angeles, CA 90089-0781
estrin@usc.edu

Abstract

PIM (Protocol Independent Multicast) is capable of supporting sparse mode (SM) and dense mode (DM) operations. In sparse mode, PIM can use shared trees (RPT) or shortest path trees (SPT) to deliver data packets. One frequently asked questions was how to decide when to use sparse mode and when to use dense mode. What is the criteria? Another unanswered question was whether packets can be lost when receivers switch from RPT to SPT. This paper reports a study of: 1) the overhead tradeoffs between dense mode operations and sparse mode operations. 2) the behaviors of PIM when receivers transitioning from RPT to SPT;

One important result presented is the cross-over point of sparse mode and dense mode overheads, which gives a hint for selecting protocol modes according to the group density metric.

Keywords: Multicast, Routing, Sparse mode, Dense mode, Overhead

1 Introduction

PIM[1] has been introduced as a scalable multicast architecture that can optimaly support multicast groups that are either sparsely distributed or densely distributed. The dense mode PIM operation is characterized by periodic broadcasts and prunes. As in DVMRP[2], a source's data packets are broadcast delivered to network nodes that are absent of state for that source and multicast group, then the few leaf subnetworks that don't have group members will set up "negative" state and prune the unwanted branches from the tree. The negative state will time out after a certain period of time, resulting in the broadcasting of data packets and pruning of unwanted branches again. In sparse mode operation, data packets are not broadcasted and only routers on the multicast tree need to keep state information for a group. The state of a multicast tree is set up when receivers' designated routers send join messages toward a Rendezvous Point (RP) or a source. In sparse mode, receivers may stay on the RP-rooted shared tree (abbreviated as RP tree or RPT) for low rate sources, while for high speed sources the receivers can choose to switch to the source rooted shortest path trees (SPT). For more detailed protocol descriptions, please see [1, 3].

In this paper, we investigate the tradeoffs of different operating modes and the boundary conditions for some transient processes. We will analyze the phenomena and provide formulae for some situations. When the complexity of the situation makes analytical methods inadequate, we use simulation to conduct experiments closely reflecting the way the protocol operates in a real network.

1.1 Overhead tradeoffs of sparse mode and dense mode

We evaluate the overhead of sparse mode and dense mode operations in terms of control bandwidth consumption and state storage requirements in all routers.

In terms of control traffic, the broadcast and prune in dense mode is a global behavior; while in sparse mode control traffic is constrained to be along multicast tree branches. In both modes, control traffic is sent periodically, normally with different frequencies. The currently suggested dense mode SPT entry timer is 3 times the sparse mode periodic refresh timer [3]. In dense mode, the number of broadcast packets can be reduced by using a very long timer value for the negative state, at the cost of keeping state for extinct groups for longer times and not being able to adapt to routing changes as quickly as needed.

In terms of storage overhead, groups operating in PIM dense mode will maintain state, either positive or

[0]This work has been supported by NSF under contract number CDA-9216321, Sun microsystem Inc and Cisco systems Inc.

negative, in all routers. Groups in sparse mode, however, only maintain state in routers on packet delivery trees.

When evaluating both of the criteria, we must consider group membership distributions. There are obviously two extreme cases of group membership distributions: one extreme is a very large group for which every router either is needed to forward packets for some downstream members or the router itself has group members attached; the other extreme is a very small group which only requires a small number of on-tree routers to deliver packets to all members. Scenarios between these extreme cases will be evaluated in detail in section 2.

Another factor that can affect an application's preference for dense mode or sparse mode is join latency. Are join latencies different in the two modes?

The join latency is incurred in two cases:

1. A new receiver joining a group. In dense mode, the receiver's designated router will send a graft message toward existing sources according to the router's negative cache state. When a graft message reaches a router already on a multicast tree (attachment point), data packets will be able to flow downstream to the new receiver. This process takes a round trip time between the new receiver and the attachment point. In sparse mode, a new receiver joins a group by sending a join message toward the Rendezvous Point (RP). When the join message reaches a router that is already on the RP tree, data packets from all existing sources will be able to flow downstream toward the new receiver. This also takes a round trip time between the new receiver and the attachment point on the RP tree. The difference in join latency is negligible in this case [1].

2. A new source appears and an existing receiver needs to join it. In dense mode, the first (or a few) packet is broadcast delivered to all routers. An existing receiver receives the first packet after a one-way delay time between the source and the receiver. In sparse mode, the new source sends the first few data packets encapsulated in RP-register messages toward the RP [3]. The RP will decapsulate the RP-register packets and forward the data packets down the RP tree. The receivers will receive the data packets after a one-way delay between the source and the receiver along the RP tree. The difference in join latency is the difference between the RP tree delay and the SPT delay. Again this is negligible in general.

It is our opinion that join latencies will not be an issue when choosing between dense mode and sparse mode operations. In the rest of the paper, we will not be concerned with the join latency issues.

[1]The difference in join latency can be in fact around a few milliseconds. But since the new receiver joins in the middle of the transmissions, and all packets sent in past history were 'gone' for the new receiver, such minor difference in joining time should be negligible in general.

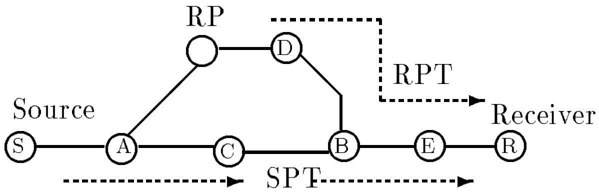

Figure 1: The receiver leaves RPT and joins SPT

1.2 Sparse Mode Dynamics: the transition from RPT to SPT

In sparse mode PIM, all receivers join the RP tree first, then the receivers switch to source-rooted shortest path trees when needed. We will asses the possibility of packet losses when switching from the RP tree to the SPT. We call a time interval a *blackout period*, if during that interval a receiver can not receive any data packets from a source due to the lack of forwarding state inside the network.

1.2.1 Black out periods during RPT to SPT transitions

Here we briefly review the PIM protocol actions when receivers switch from a RP tree to a source rooted shortest path tree. See the example in figure 1, the receiver R already on the RP tree receives a series of data packets from source S and decides to switch to S rooted shortest path tree. R's designated router initiates a SPT-join toward the source, which will set up the SPT state from S to R. B continues to accept packets from the RP tree before the upstream SPT state is fully set up. When a data packet from S arrives at B along the SPT branch, B knows that the upstream SPT state has been completely setup (because the incoming interface for SPT is different from that of the RPT's), and will prune itself off the RP tree for source S [2]. A flag '*SPT-bit*' is set in B to signal the completion of this transition [3].

In figure 1, once the transition from RPT to SPT is completed, router B will reject all S's data packets coming down the path $RP \rightarrow D \rightarrow B$, and will only accept data packets from S that come down the SPT path. Assume packets sent by S are numbered 1, 2, ..., n according to the order of departure times at S, and the

[2]Such prunes will set up negative cache state along the RP tree so that packets from S will not be forwarded along the RP tree any more.

path $A \rightarrow RP \rightarrow D \rightarrow B$ is longer than the path $A \rightarrow C \rightarrow B$. After router A has established SPT routing state, assume the first data packet A forwards down the SPT path is packet $\#i$. When packet $\#i$ arrives at B, B will set its SPT-bit, indicating the completion of RPT to SPT switch. If by this time, packet $\#(i-1)$ is still in the transmission path $RP \rightarrow D \rightarrow B$, it will be rejected by B when it eventually arrives at B. Section 3 will give a quantitative analysis of such black out periods when they do exist.

1.2.2 Are there duplicate packets during the transitions?

Can there be duplicates when control messages are lost during the switch from RPT to SPT? One possibility is to have the negative cache state along the RPT be mistakenly deleted so that receivers will receive packets coming down both SPT and RPT. But in PIM even if control messages are lost, such mistakes can not occur. This can be shown in figure 1, if both the SPT and RPT paths are forwarding data packets from S to B, B with its *SPT-bit* set, will do an incoming interface check and only accept packets from the SPT path and drop all packets coming from the RPT path, i.e. no duplicate packets. However, when B's incoming interface is a multiaccess LAN, i.e. the incoming interfaces for the RPT and the SPT are the same, the incoming interface check will not be able to distinguish packets forwarded by SPT from those forwarded by the RPT. In this situation, the *assert* mechanism [3] will be activated so that the router that forwards packets from the RPT onto the multiaccess LAN is pruned off. Therefore in the rest of this paper, we will not be concerned with duplicate packets.

1.3 Design of PIMSIM, a PIM simulator

A packet level simulator PIMSIM [4] has been developed to exercise PIM mechanisms. The goal of such a simulator is to capture the details of the dynamic behaviors of basic PIM mechanisms. We chose MaRS (Maryland Routing Simulator) [5, 6] as the basis of the simulator, and utilized or adapted MaRS's event management routines, basic network construction models and X window user interface routines. The new simulator has PIM specific routing modules, multicast traffic modules, multicast capable node modules and multiaccess LANs.

Before applying the simulator to real simulations, we tested the simulator over a number of topologies, and with various kinds of multicast groups. We verified that the correct control messages were sent and correct state information was established in the network. For more details about PIMSIM design and usage, please see [4].

2 Trade-off of overheads in sparse mode and dense mode

In this section, we define 3 basic metrics and formulate the overheads of sparse mode and dense mode operations. We will also present simulation results showing the temporal dynamics of bandwidth overhead in a real network.

2.1 Metrics and Formulae

The overhead comparisons of different modes are based on two measures: the total *state storage* required network wide and the total *control bandwidth* consumed to maintain certain multicast groups inside the network.

2.1.1 Storage Overhead

Let $v_1, v_2, ...v_n$ be the nodes inside the network, $C_i(g, S_j)$ be the storage cost on node i for group g, source S_j (or RP). The overall storage cost under a certain mode is defined as (assume there are n nodes in the network and m sources for g):

$$Cost_{storage}(mode) = \sum_{i=1}^{n}\sum_{j=1}^{m} C_i(g, S_j) \tag{1}$$

Since the difference in dense mode and sparse mode routing entries is rather small [3, 7], it is sufficient to compare the total number of entries under different operating modes in the storage comparisons. In the rest of this section, we assume dense mode and sparse mode entries are of the same size.

Let the cost of each routing entry be 1. In dense mode, each router must maintain a routing entry for each (source, group) pair — either positive or negative. The total storage cost of dense mode multicast routing entries in the network will be:

$$Cost_{storage}(dense\_mode, G) = n \times m \tag{2}$$

In sparse mode, the storage cost consists of the costs of the SPT entries, RPT (*,G) entries and the negative cache (S,G,RPbit) entries. Let $N_{(*,g)}(G)$ be the number of RPT entries for group set G, $N_{(s,g)}(G)$ be the number of SPT entries for group set G, and $N_{(s,g,RPbit)}(G)$ be the number of negative cache entries.

If every multicast group in G uses *shared tree* in sparse mode, the storage cost will be:

$$Cost_{storage}(rpt\_mode, G) = N_{(*,g)}(G) + N_{(s,g)}(G) \tag{3}$$

Note that if due to low data rate, all RPs do not establish SPT state and receive all data packets encapsulated in register messages [3], $N_{(s,g)}(G)$ in the above formula will be zero. This mode is also representative of CBT as well [8].

If members switch to SPT, the storage cost will be:

$$Cost_{storage}(spt\_mode, G) = N_{(*,g)}(G) + N_{(s,g)}(G) + N_{(s,g,RPbit)} \quad (4)$$

2.1.2 Bandwidth (control) Overhead

For a multicast group g, let $P(t, l, g)$ be the number of tree maintenance packets sent on link l from time 0 to time t. The total number of tree maintenance packets in the network is:

$$Cost_{ctrl\_band}(t, g) = \sum_{l=1}^{k} P(t, l, g) \quad (5)$$

The local IGMP or PIM query and report messages have no global effects and are the same for dense mode and sparse mode. Such local messages will be ignored in the definitions and experimental measurements in this paper.

Since different protocol modes use different kinds of tree maintenance packets, dense mode and sparse mode bandwidth overheads need to be measured in different ways. In dense mode, data packets are broadcast delivered to propagate routing information. A prune packet is triggered by an unwanted data packet, which will delete an outgoing interface in a routing entry. The bandwidth overhead of dense mode operation is thus defined as the total number (or bytes) of unwanted data packets transmitted over all network links, plus the total number (bytes) of periodic prune messages. In the following discussions, bandwidth overhead is measured in unit of packet count. The bandwidth overhead in bytes can be estimated based on overhead packet count and application packet sizes.

Let $D_{unwanted\_pkt}(t)$ be the total number of unwanted data packets from time 0 to t, and $D_{prune}(t)$ be the total number of prunes sent during the same period, then *dense mode* bandwidth overhead can be defined as,

$$Cost\_DM_{ctrl\_band}(t, G) = D_{unwanted\_pkt}(t, G) + D_{prune}(t, G) \quad (6)$$

In sparse mode, the bandwidth overhead can be defined as the total number of PIM control messages (D_{pim_msg}) sent (i.e. S,G join/prune, *,G join, S,G RPbit prune):

$$Cost\_SM_{ctrl\_band}(t, G) = D_{pim\_msg}(t, G) \quad (7)$$

2.1.3 Density of a Multicast Group

The *density* of a multicast group reflects the percentage of "on-tree" links vs the total number of links in the network.

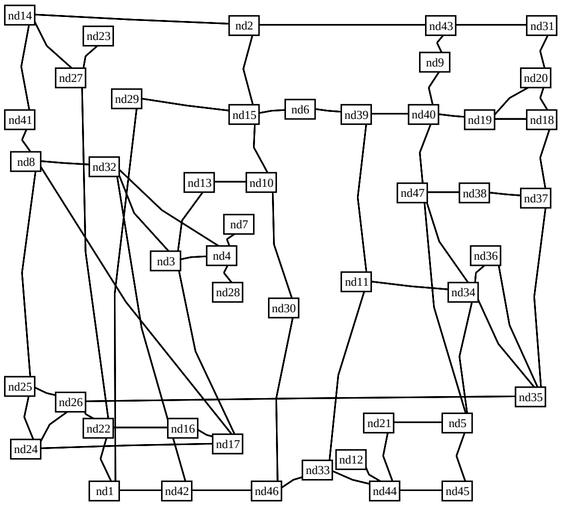

Figure 2: Arpanet topology used in dense mode and sparse mode simulations

Let g be a global-scope multicast group with only one sender s; let l be the total number of links on the tree rooted at s, using a shortest path tree algorithm to calculate the multicast tree[3]. Let L be the total number of links within the network. The following measure is called the *density of g with respect to* source s: [4]

$$Density(s, g) = \frac{l}{L} \quad (8)$$

If g has more than one sender, the density metric for the whole group is defined as the average of densities over all sources. Let $s_1, s_2, ..., s_m$ be the sources of group g, where m is the number of sources. Let l_i $(1 < i < m)$ be the number of links on the shortest path tree rooted at source s_i. The density of group g, taken into account all sources, is defined as: [5]

$$Density(g) = \frac{\sum_{i=1}^{m} l_i}{m \times L} \quad (9)$$

For a group with known density metrics, the following provides a lower bound for the *dense mode control bandwidth cost* (formula 6), where the equality holds when there is only one copy of each unwanted data packet traveling on each link during each broadcast and prune cycle (assume negative cache time-out interval is T_{DM}):

$$Cost\_DM_{ctrl\_band}(t, g) \geq (1 - Density(g)) \times L \times 2 \times m \times \frac{t}{T_{DM}} \quad (10)$$

[3] If there is more than one shortest path tree, choose the one with the least number of on-tree links.

[4] The *sparseness of g with respect to* source s is defined as the reciprocal of $Density(s, g)$.

[5] The corresponding *sparseness* metric considering all sources is defined as the reciprocal of *Density(g)*.

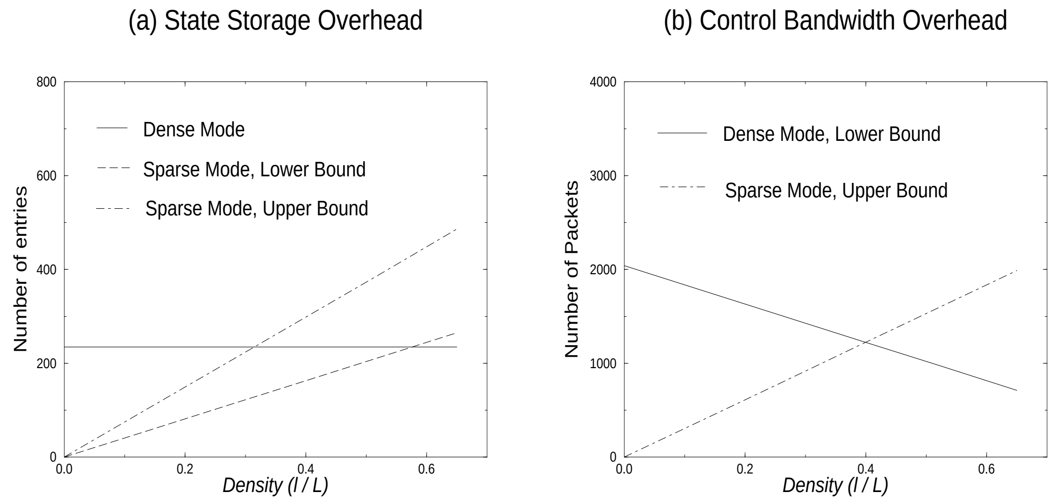

Figure 3: Tradeoffs of Sparse mode (SPT) and Dense mode in arpanet (47-node), for 5-sender multicast groups

When all receivers use SPTs in sparse mode operation, if the RP is placed at a receiver [6], control messages will only travel through links that are on at least one of the source rooted shortest path trees. When different shortest path trees and the RP tree overlap, control messages are aggregated into one control packet. The following gives an upperbound to the *sparse mode control message overhead*, in terms of number of control packets (c.f. formula 7). Assume the sparse mode refresh interval is T_{SM}:

$$Cost\_SM_{ctrl\_band}(t,g) \leq Density(g) \times L \times m \times \frac{t}{T_{SM}} \quad (11)$$

If the RP is placed at member s_i which is also a sender, the number of SPT entries will be $Density(g) \times L \times m$, the number of RPT (*,g) entries will be $Density(s_i, g) \times L$. The *sparse mode storage cost* represented by formula 4 can be rewritten as:

$$\begin{aligned} Cost_{storage}(spt\_mode, g) \quad = \quad & Density(g) \times L \times m + \\ & Density(s_i, g) \times L + N_{(s,g,RP)} \quad (12) \end{aligned}$$

Since the number of negative cache entries is smaller than the total number of SPT entries when the RP is placed at a member/sender, the following inequalities provide an upper and a lower bound for $Cost_{storage}(spt\_mode, g)$:

$$\begin{aligned} & Density(g) \times L \times m + Density(s_i, g) \times L \\ & \leq Cost_{storage}(spt\_mode, g) \leq \\ & 2 \times Density(g) \times L \times m + Density(s_i, g) \times L \quad (13) \end{aligned}$$

Figure 3 shows the memory and bandwidth overhead curves of the above formulae for the arpanet topology (fig 2) with an audio conference of 5 senders. Fig 3(a) shows the dense mode vs sparse mode *memory tradeoff*: the dense mode storage cost represented by formula (2) and the upper/lower bounds of the sparse mode SPT operation (inequality (13)). Figure 3(b) shows the sparse mode and dense mode *control bandwidth* tradeoff: the lower bound of dense mode bandwidth cost represented by inequality (10) and the sparse mode bandwidth overhead upper bound from inequality (11)[7]. The tradeoffs of dense mode and sparse mode under different group densities are very obvious in these two graphs.

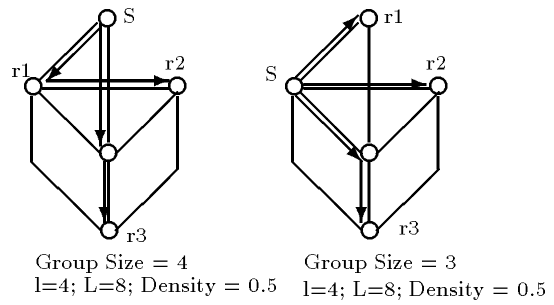

Figure 4: Example of two multicast groups having the same density

The group size and the density of a tree are related, but there is no one-to-one correspondence. For a certain network and a given source, there can exist a number of groups with *different sizes* but with the *same density*. Fig 4 shows an example of two multicast groups having the same density. In the topology shown, the maximum size of a group with density 0.5 is 4, the minimum size of a group with the same density is 3.

[6] or the receiver's first hop router

[7] Note that in figure 3, the density axis doesn't extend to 1, this is because none of the multicast trees can include all network links.

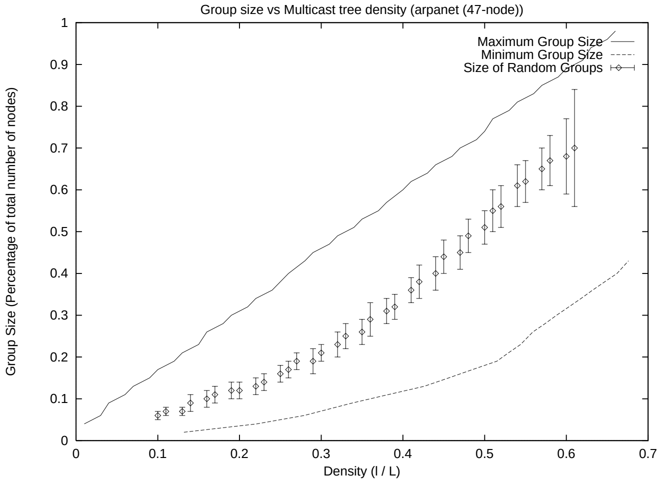

Figure 5: Maximum and Minimum group size vs density

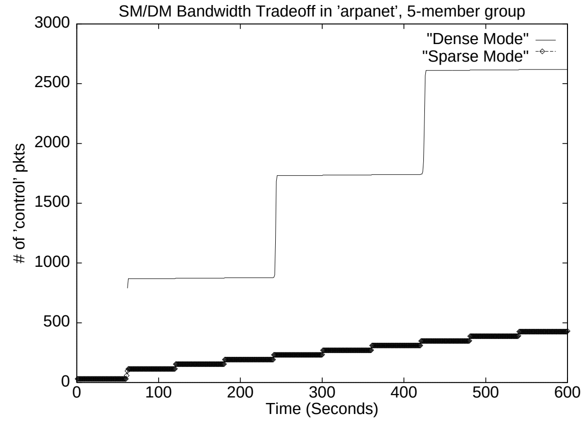

Figure 6: SM/DM control bandwidth overhead supporting a sparse group in arpanet

It is easy to show that the maximum group size under a certain density is $density \times L$[8]. The minimum group size under a certain density, however, is dependent on the topology and can not be expressed in a simple formula. We constructed an experiment over the arpanet topology and two 100-node random topologies, measured the corresponding minimum group sizes for each density value. Figure 5 shows three curves of the maximum group size, minimum group size and the average size of random groups over the arpanet topology. The error bars for random groups represent the standard deviations — there are about 50 - 100 random groups for each density value.

Note that all curves terminate at the density of about 0.7 — at density of 0.7 there is no way to increase the density of the tree further, all leaf domains are members of the group and all network nodes are on tree. In fact the maximum group size is n in a n-node L-link strongly connected network. Only $n-1$ out of the L links are used to construct a tree with n nodes in this case, the maximum density of a group in a n-node network is $\frac{n-1}{L}$. Therefore, for networks of different sizes, if they have the same node degree, the maximum group size curve (in units of *percentage of nodes*) should remain the same. We speculate that under the same node degree, their minimum group size curves and their random group size curves should also have little difference [9].

[8] We can prove by construction that if the size of a group is incremented by adding new nodes as members, in the same order a breadth-first-search algorithm starting from the source would visit the nodes, the size of the group is kept maximum for the density of the tree constructed.

[9] We ran experiments on two 100-node random topologies, one with average node degree of 3.1, the other with node degree of 5. The result from the network with node degree of 3.1 is very close to that shown in fig 5. The result from the other 100-node topology has similar shapes, except that every curve is "compressed" in the horizontal direction to the left by about 40%.

2.2 SM/DM tradeoff experiments

The figure 3 upperbound and lowerbound overhead curves for groups with certain densities are useful for estimating the ranges of overhead. Whereas simulations on a specific multicast group inside a particular topology can make it possible to measure precisely how the overhead is incurred over time.[10]

In the subsequent simulation experiments, sparse mode groups and dense mode groups are put in the same network separately in different runs. Reported here are simulations over the Arpanet topology shown in figure 2.

The group involved in the simulation is assumed to have global scope. All sources share the same sending behavior. The simulated scenario is a 5-participant audio conference. The 5 participants are located at nodes 2, 24, 26, 27 and 29 of the arpanet topology (fig 2). In sparse mode operation, node 24 is chosen as the RP for the group. In the first run the group is started in dense mode, and all sources start sending at 60 seconds offset from the network startup time. In the second run, the group is started in sparse mode, and the same experiment repeated.

To log parameters as the experiments progress, a special monitor component is created in the simulator to periodically collect the storage usage and various packet counts at different nodes and links. The sampling times $t_0, t_1, ...t_i, t_{i+1}, ..., t_n$ can be configured into the monitor component. For this particular experiment, all measurement periods are set to the same small constant ($500ms$).

[10] The storage overhead for dense mode can be treated as a constant (fig 3(a)). The storage overhead for sparse mode depends on the number of on tree nodes. It can be derived from the tree costs as reported in [9]. Therefore we won't replicate the experiments already done in [9] and only present the results for control bandwidth overhead here.

Figure 6 shows the bandwidth overhead when supporting this group with 1) dense mode (thin solid line); 2) sparse mode with SPT (thick line). The step function of the dense mode measurement reflects the dense mode periodic broadcast and prune behavior. In this experiment, all timers are set to the default value suggested in the PIM specification document [3]. In a real network, the timers can be set differently in order to achieve a more suitable bandwidth/adaptability tradeoff. In general a longer negative cache timer will result in less periodic broadcast and prune traffic, and will also result in slower adaptation to routing changes. The small steps in the sparse mode measurement reflects the artifacts of the way the simulator is initialized — all PIM routing modules are started at the same time. The group density for this conference is 0.17.

It can be seen that for this 5-sender/receiver group, the number of dense mode control packets increases much faster than the sparse mode control packets. The advantage of sparse mode for small groups is obvious in terms of the control overhead.[11]

3 Black out period when switching from RPT to SPT

If a multicast group operates in sparse mode, all receivers will join the RP tree first. When a source's packet rate is high enough, the receivers will switch to the source-rooted shortest path tree. As described in subsection 1.2, there may exist a black out period during the switch from RPT to SPT.

The length of the black out period is dependent on the difference between relevant sections along the RPT and the SPT paths. The number of packets lost is proportional to the length of the black out period and the source's rate.

First, we formally define the relevant parameters and metrics.

Let s be a source and r be a receiver, $D_{spt}(s,r)$ be the propagation delay for packets from source s to receiver r along the shortest path. Let $D_{rpt}(s,r)$ be the one-way delay along the RP tree path from s to r. Let $R(s)$ be s's sending rate in unit of *packets/second*.

[11] The bandwidth overhead of a network supporting mixed sparse mode and dense mode groups has an *additive property* — the total bandwidth overhead is equal to the sum of the bandwidth overhead when the network has only sparse mode groups, and the bandwidth overhead when the network has only dense mode groups. This is because dense mode control messages and sparse mode messages are always sent in separate packets. Storage overhead also has the *additive property*. Experimental results gained from networks with only sparse mode groups and results with only dense mode groups, can be combined to predict network overhead with mixed sparse and dense mode groups.

The number of packets in flight along the shortest path tree from s to r at stable state is F_{spt}:

$$F_{spt}(s,r) = D_{spt}(s,r) * R(s) \qquad (14)$$

The number of packets in flight along the RP tree from s to r at stable state F_{rpt} is:

$$F_{rpt}(s,r) = D_{rpt}(s,r) * R(s) \qquad (15)$$

When a receiver switches from RPT to SPT, the number of packets lost during the transition $L_{rpt->spt}$ is approximately:

$$L_{rpt->spt}(s,r) = F_{rpt}(s,r) - F_{spt}(s,r) \qquad (16)$$

In the most favorable situation RPT to SPT switch can be free of packet loss if,

1. The path from the source to the receiver along the RP tree is exactly the same as the path along the SPT rooted at the source. I.e. the physical packet delivery path does not change during the RPT to SPT transition;

2. The difference in delay between the RPT path and the SPT path is smaller than the time interval between consecutive packets.

When loss does happen, the worst case is that the inter-packet intervals are much smaller than the delay difference between the RPT path and SPT path. This worst case scenario may happen resulting from the selection of an extremely non-favorable router as an RP for a widely dispersed multicast group.

Note that when the source rate is fixed, packet loss during the RPT to SPT transition is directly related to the data packet size — the larger the packet size, the longer the inter-packet interval, the fewer the number of packets lost. For example, a vat[12] pcm2 source (71Kb/s, 40ms frames) with a 50 ms SPT path to a receiver and a 100 ms RPT path, the maximum number of packet loss can be two 355 byte packets. But if the vat uses pcm4 encoding (68Kb/s 80ms frames), no packet will be lost duing the RPT to SPT switch!

To fully understand the transitioning process, it is ideal if one simple experiment setup can cover all possible scenarios. The following statement effectively reduces the experiment space without sacrificing the generality of our simulation results.

Statement 1 *In PIM sparse mode, when a receiver switches from RPT mode to SPT mode, the number of packets dropped during the black out period is only dependent on two factors:*

[12] Vat is an audio terminal tool developed by Van Jacobson and Steve McCanne at LBL.

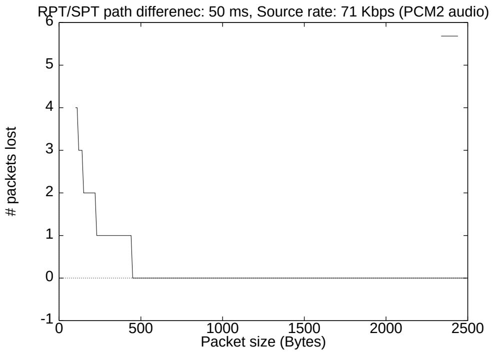

Figure 7: Packet loss as a function of packet size (source rate fixed)

1. *the delay difference between the RPT path and SPT path from the source to the receiver; and*
2. *the source's sending behavior (rate, packet size).*

Other factors, such as topological features in a particular network, are irrelevant to the packet loss during this period.

Hence it suffices to simulate the topology shown in figure 1 with ranges of different link and source parameters. The results will hold in other topologies and group membership distributions if the RPT path and SPT path delay difference and the source's sending behavior are the same.

Figure 7 shows simulated packet loss as a function of packet size in the network of figure 1. The source rate is fixed at $71Kbps$. The difference in delay along the RPT path and SPT path is 50 ms — roughly the worst case scenario for arbitrary RP placement inside the continental United States. One useful fact in this picture is that for a PCM encoded audio source, there is no packet loss when the packet size is larger than 500 bytes.

Figure 8 shows packet loss as a function of both source rate and packet size in the network of figure 1, for a PCM encoded audio source when the RPT/SPT path length difference is $50ms$. The contour lines on the base plain show the boundaries of regions having the same drop rates.

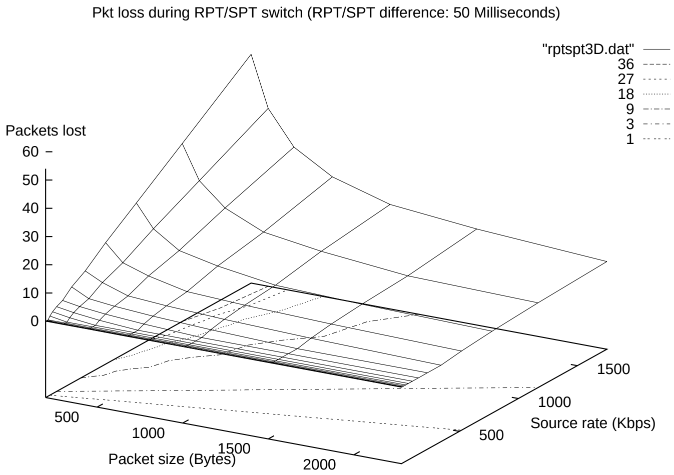

Figure 8: Packet loss as as function of source rate and packet size

4 Conclusion

The tradeoffs between sparse mode and dense mode PIM operations are evaluated via two measures: the state storage overhead, and the control bandwidth overhead. With known measures of group *density*, the state storage and the control bandwidth overheads can be calculated for dense mode operations. The bounds for such overheads can be estimated for sparse mode operations. Simulations were run over the arpanet topology and results presented. The arpanet experiments showed that for groups with densities of less than 0.3, sparse mode operation has less storage and bandwidth overhead in general. A density value of 0.3 in the arpanet corresponds to groups with 10% to 20% of the total number of network nodes. We speculated and verified with random topologies that the results from the arpanet topology can be generalized to networks of different sizes with the same average node degree. To estimate the overhead tradeoff in networks with different node degrees the results need to be factored along the density axis by the ratio of the two networks' node degrees.

The number of packets lost during the transition from RPT to SPT is a function of the path length difference between the RPT and SPT branches and the source's inter-packet intervals (or the source's rate and packet sizes). Slow to moderate rate applications such as audio sources normally suffer no or insignificant packet losses in normal operating enrironments. High speed sources can incur higher packet loss rates, especially when the difference between the RPT path and SPT path is significant. Since such losses only occur when switching from RPT to SPT, it is advised to avoid frequently switching between the two different tree types unnecessarily.

References

[1] S. Deering, D. Estrin, D. Farrinacci, V. Jacobson, C. Liu, and L. Wei. An architecture for wide-area multicast routing. In *Proceedings of the SIGCOMM*, London, 1994.

[2] S. Deering and D. Cheriton. Multicast routing in datagram internetworks and extended lans. *ACM Transactions on Computer Systems*, pages 85–111, May 1990.

[3] S. Deering, D. Estrin, D. Farrinacci, V. Jacobson, C. Liu, and L. Wei. Protocol independent multicast (pim): Protocol specification, 1995. ftp://catarina.usc.edu:/pub/estrin/PIM/ietf-idmr-pim-01.ps.

[4] L. Wei. The design of the usc pim simulator (pimsim). Technical Report TR 95-604, Computer Science Department, USC, feb 1995.

[5] Cengiz Alaettinoglu, A Udaya Shankaar, Klaudia Dussa-Zieger, and Ibrahim Matta. Design and implementation of mars: A routing testbed. Technical Report CS-TR-2964, Computer Science Department, University of Maryland, sep 1992.

[6] Cengiz Alaettinoglu, A Udaya Shankaar, Klaudia Dussa-Zieger, and Ibrahim Matta. Mars (maryland routing simulator) - version 1.0 user's manual. Technical Report CS-TR-2687, Computer Science Department, University of Maryland, jun 1991.

[7] Dino Farrinacci and Puneet Sharma. Private communications, 1995.

[8] A. J. Ballardie, P. F. Francis, and J. Crowcroft. Core based trees. In *Proceedings of the ACM SIGCOMM*, San Francisco, 1993.

[9] L. Wei and D. Estrin. The tradeoffs of multicast trees and algorithms. In *Proceedings of the 1994 International conference on computer communications and networks*, San Francisco, September 1994.