Reliable End System Multicasting with a Heterogeneous Overlay Network

Jianjun Zhang, Ling Liu, Calton Pu and Mostafa Ammar
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332, U.S.A.
{zhangjj, lingliu, calton, ammar}@cc.gatech.edu

Abstract

This paper presents PeerCast, a reliable and self-configurable peer to peer system for End System Multicast (ESM). Our approach has three unique features compared with existing approaches to application-level multicast systems. First, we propose a capacity-aware overlay construction technique to balance the multicast load among peers with heterogeneous capabilities. Second, we utilize the landmark signature technique to cluster peer nodes of the ESM overlay network, aiming at exploiting the network proximity of end system nodes for efficient multicast group subscription and fast dissemination of information across wide area networks. Third and most importantly, we develop a dynamic passive replication scheme to provide reliable subscription and multicast dissemination of information in an environment of inherently unreliable peers. We also present an analytical model to discuss its fault tolerance properties, and report a set of initial experiments, showing the feasibility and the effectiveness of the proposed approach.

1 Introduction

End System Multicast (ESM) is one of the practical approaches to disseminating information to a large number of receivers. Peer-to-peer (P2P) ESM is rising as a promising distributed ESM paradigm for group communication applications like audio and video conference, event and content distribution, and multi-user games. P2P end-system multicast applications can be classified as decentralized P2P or hybrid P2P systems depending on whether a central server is used for multicast group membership management and information dissemination.

Supporting End System Multicast with a decentralized peer to peer overlay network poses a number of challenges. First, a peer to peer End System Multicast usually disseminates information through an overlay network of end-system nodes interconnected by unicast links at IP layer. A critical issue for peer to peer ESM is how to reduce the traffic across the wide area overlay network and how to minimize the multicast latency experienced by end users. Second, nodes in a wide area peer to peer overlay network tend to be heterogeneous in terms of computing capacities and network bandwidth. Therefore, there is a need for an efficient end system multicast protocol that can automatically balance multicast load on end system nodes while maintaining the decentralization. Third but not the least. It is widely recognized that large scale are confronted with highly dynamic peer turnover rate [13]. For example, in both Napster and Gnutella, half of the peers participating in the system will be replaced by new peers within one hour. Thus, maintaining fault-tolerance in such a highly dynamic environment is critical to the success of a peer-to-peer ESM system.

Much effort in peer to peer ESM systems have been contributed towards addressing the first problem [4, 5, 6, 9, 11, 18]. It is widely recognized that further deployment of P2P technology for end system multicast applications other than simple file sharing demands practical solutions to the second and third problems.

With these challenges in mind, we present PeerCast, an efficient, self-configurable, and yet reliable ESM service on a top of a network of loosely coupled, weakly connected and possibly unreliable peers. Our approach has three unique features compared with existing approaches to application-level multicast.

First, we propose a capacity-aware overlay construction technique to balance the multicast load among peers with heterogeneous capabilities. Our load balancing mechanism can effectively level the workload among end-system nodes and endorses end-system nodes who contribute more resources. To the best of our knowledge, PeerCast is the first ESM system that takes into account of end-system heterogeneity.

Second, we develop an effective node clustering technique based on the landmark signature technique, which can cluster group end-system nodes by exploiting their physical network proximity for efficient multicast group subscription and fast dissemination of information across wide area networks.

Third and most importantly, we develop a dynamic passive replication scheme to provide reliable subscription and multi-

cast dissemination of information in an environment of inherently unreliable peers. Replication is a proven technique for masking component failures. However, designing replication scheme for peer to peer ESM has several specific challenges: (1) All nodes holding replicas must ensure that the replication invariant (at least a fixed number of copies exist at all time) is maintained. (2) The rate of replication and the amount of replicated data stored at each node must be kept at levels that allow for timely replication without introducing too much network overhead even when regular nodes join and leave the ESM overlay network. We develop an analytical model to discuss the fault tolerance properties of PeerCast, and report a set of initial experiments, showing the feasibility and the effectiveness of the PeerCast approach.

2 PeerCast System Overview

2.1 System Architecture

Peers in the PeerCast system are end system nodes on the Internet that execute multicast information dissemination applications. Peers act both as clients and servers in terms of their roles in serving multicast requests. Each end-system node in the overlay network is equipped with a PeerCast middleware, which is composed of two-tier substrates: *P2P network management* and *ESM Multicast Management*.

P2P Network Management Substrate. The P2P network management substrate is the lower tier substrate for P2P membership, lookups, and communication among end-system nodes. It consists of P2P membership protocol and P2P lookup protocol. With the *P2P membership protocol*, a new node can join the PeerCast system by contacting an existing peer (an entry node) in the PeerCast network. There are several bootstrapping methods to determine an entry node. Here we assume that the PeerCast service has a well-known set of bootstrapping nodes which maintain a short list of PeerCast nodes that are currently alive in the system.

In PeerCast every peer participates in end system multicast execution, and any peer can create a new multicast service of its own interest or subscribe to an existing multicast service. Each multicast service is implemented with a multicast tree structure. There is no scheduling node in the system. No peers have any global knowledge about other peers. When a new multicast service is posted by a peer, this peer first determines which peer will be the root of the new multicast tree through the *P2P lookup protocol* in the ESM Management Substrate. The decision is made by the Multicast Group Membership protocol, which takes into account several factors like peer resource diversity, load balance between peers, and overall system utilization.

ESM Management Substrate. The ESM substrate is the higher layer responsible for ESM event handling, multicast group membership, multicast payload delivery, and cache management. It consists of three protocols. The *Multicast Group Membership* protocol handles all multicast group subscription requests. An ESM source joins the ESM overlay as a peer. It first maps its service to an end system node based on an identifier matching criteria and the lookup protocol provided by the P2P network management substrate. A subscriber of an ESM service first locates the service of interest through the P2P lookup protocol. Following the multicast group membership protocol, the subscriber initializes a subscription request and includes the ESM service identifier as the parameter. The request is then forwarded through a serial of end-system nodes towards the ESM source on the P2P overlay. The subscription request terminates when it hits either the ESM source or an end-system node that is already in the multicast group. The end-system nodes encountered on the forwarding pass cooperate with each other to form an ESM subscription path from the subscriber to the ESM source. Multicast information will be delivered in the reverse direction along this path down from the ESM source. The *Multicast Information Dissemination* protocol is responsible for disseminating a multicast payload message through links among end system nodes over which the corresponding multicast tree is maintained. When some peers depart from or fail in the ESM overlay, end-system nodes use *Multicast Overlay Maintenance* protocol to ensure that the set of multicast groups to which it subscribes is re-assigned to the rest of peers, while maintaining the same objectives – exploiting network proximity and balance the load on peers.

3 Exploiting Network Proximity in PeerCast

3.1 P2P Protocol: The Basics

PeerCast peer to peer network is a Distributed Hash Table (DHT) based structured P2P network. It allows applications to register, lookup, and remove a multicast subscription using an m-bit identifier as a handle. Each subscription is mapped to a unique, effectively random m-bit identifier. Similarly, each peer in PeerCast corresponds to a set of m-bit identifiers, depending on the amount of resources donated by each peer. A peer that donates more resources is assigned to more identifiers. The number of identifiers associated to each peer is calculated by using the effective donation function provided in [8]. A peer p is described as a tuple of two attributes, denoted by $p : (\{peer\_ids\}, (peer\_props))$. $peer\_ids$ is a set of m-bit identifiers. Identifiers are generated to be uniformly distributed by using hashing functions like MD5 and SHA-1. $peer\_props$ is a composite attribute which is composed of several peer properties, including IP address of the peer, peer, resources such as connection type, CPU power and memory, and so on.

Identifiers are ordered in an m-bit identifier circle modulo 2^m, which forms a logical ring. The 2^m identifiers are organized in an increasing order in the clockwise direction. The distance between two identifiers i, j, denoted as $Dist(i, j)$,

is the shortest distance between them on the identifier circle, defined by $Dist(i,j) = min(|i-j|, 2^m - |i-j|)$. Identifier i is considered as "numerically closest" to identifier j when there exists no other identifier with a closer distance to j. Given a key of m bits, the PeerCast protocol maps it to a peer whose peer identifier is numerically closest to that key. A peer p invokes its local lookup function p.lookup(i) to locate the identifier j which is numerically closest to i.

The lookup is performed by routing the lookup queries towards their destination peers using routing information maintained at each peer. The routing information consists of a *routing table* and a *neighbor list* for each identifier possessed by a peer. The routing table is used to locate a peer that is more likely to answer the lookup query, where a neighbor list is used to locate the owner peer of a multicast note and replication peers of the subscription. The routing table is basically a table containing information about several peers in the network together with their identifiers. The neighbor list points to immediate neighbors on the identifier circle. The routing tables are used to speed up the lookup process. Initialization and maintenance of the routing tables and the neighbor lists do not require any global knowledge.

Due to the space restriction, we omit the other details about the routing information maintenance and the network proximity argument of our P2P protocol. Readers who are interested may refer to our technical report [7] and other DTH protocols [14, 17, 10, 12].

3.2 Exploiting Network Proximity in PeerCast

Motivation. Consider a typical lookup routing path in a structured P2P network as shown in Figure 1. A lookup request is initiated by a peer with identifier C83E91, and is targeted to peer with identifier DA06C5. Due the problem known as "logarithmical deterioration of routing" [12], the length of each hop increases logarithmically as the request is forwarded closer to the target peer. This problem is caused by the prefix matching requirement of routing in P2P network, and the uniformly random distribution of peer identifiers. If we envision the whole identifier space as a domain partitioned by the identifier's prefix, the kth step of forwarding the request lookup(i) is equivalent to forwarding the query into a subset of the identifier space in which identifiers share the first k digits with the parameter i. The cardinality of such subset decreases logarithmically as more hops the lookup request has been forwarded. In another word, there will be less peers residing within the network vicinity of forwarding peer. Although [12] suggested various mechanisms to reduce the length of the forwarding path, the prefix matching requirement of routing limited the benefits.

In contrast to Figure 1, Figure 2 presented an ideal routing scheme. Most of the lookup forwards are within the vicinity of the forwarding peer. The long hops are used only for necessary crossing of the Internet backbone. However, this objective could only be achieved when peers are aware of the other peers within its network vicinity and refer them in its local routing table and neighbor list. In a widely distributed P2P network, it is infeasible for a peer to learn about who are its closest peers, as the randomness distribution of the peer identifier put a limit on the qualities of the lookup path that is subject to the prefix matching limitation.

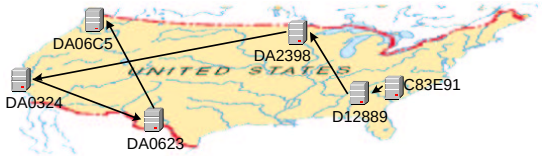

Figure 1: Logarithmical deterioration of routing in structured P2P network

Figure 2: Routing regarding network proximity in PeerCast P2P network

Landmark Signature and ESM. In PeerCast, we propose to use landmark signature based technique to cluster end system nodes of the ESM network, aiming at achieving higher percentage of lookup forwarding hops to be within each others network vicinity. One way to tackle this problem is by twisting the distribution of identifiers on the identifier circle such that peers physically closer would have numerically closer peer identifiers. A key challenge is to define an effective mechanism that can partition the end system nodes into network proximity groups, while preserving the randomness of the identifier distribution, and avoiding the problem discussed in [15].

In PeerCast, we propose a clustering scheme as the *Landmark Signature Scheme*. Landmark points are a set of end-system nodes that are randomly distributed across the Internet. We refer them as $(B_1, B_2, B_3, \ldots, B_n)$. An end-system measures its distance to the given set of landmarks points and record the result in a vector $D < d_1, d_2, d_3, \ldots, d_n >$, which we refer to as *Landmark Vector*. The intuition behind our landmark signature scheme is that end-system nodes that are close to each other will have similar landmark signatures. To simplify the incorporation of the landmark signature technique into our P2P network, we use the relative order of elements to capture the similarity of different landmark vectors. We encode the relative element order into a binary string and refer to it as a *Landmark Signature* of a peer.

Creating Landmark Signature. In PeerCast, each peer generates its landmark signature and then uses its landmark signature to generate its peer identifier upon the entry to the ESM network. Concretely, a new peer obtains a set of landmark points through the bootstrapping service when it joins the P2P network. The new peer generates its landmark signature using this set of landmark points. It also generates

its identifiers using the normal identifier generation functions such as MD5 and SHA-1. The landmark signature then substitutes a substring of each identifier at the same offset. The resulting identifiers now contain the network proximity information that can be used to identify the network vicinity of the new peer. As the new peer joins the P2P network using the standard APIs, it aligns itself along with the other peers that have similar landmark signatures. In section 3.3, we will show how this property can be used in various ways to reduce the latency of the lookup service and improve the performance of our ESM overlay networks. We refer the offset at which the landmark signature is inserted as the "Splice offset". The value of the splice offset is a system parameter that could be tuned according to the overlay population.

Clustering Peers Using Their Landmark Signatures. In PeerCast, we use the landmark signature technique to control the distribution of peer identifiers. By choosing proper value of the splice offset, say b digits, we introduce network proximity based clustering of peer identifiers while preserving the randomness of the peer identifier distribution. We envision that the leading b digits before the splice offset randomly partitions the identifier circle into a number of 2^b "buckets". Peers belong to the same network vicinity are then clustered with each "bucket" by their landmark signatures. Thus, if there are enough peers in the overlay, the multicast root peer will be surrounded by its physical network neighbors. They will be connected as the children of the multicast root peer, according to the way our multicast membership protocol works. Thus, we could reduce the link latency of the direct children of the root, because those links are more likely to be within the same network sub-domain as the root peer.

3.3 End System Multicast Scheme

The PeerCast End System Multicast management substrate is built on top of the PeerCast P2P network. It uses the APIs provided by the PeerCast P2P network management substrate to create multicast groups and subscribe to ESM services. In PeerCast, an End-System Multicast (ESM) service is established in three steps.

Step 1: Creating Multicast Group and Rendezvous Node. An ESM source (service provider) first defines the semantic information of the service that it will provide and publishes a summary of its service on an off-band channel. Potential subscribers could locate such information using the off-band channels Peers that subscribe to this ESM service will form a group, which we refer to as *multicast group*. Each multicast group in PeerCast is uniquely identified by a m-bit identifier, denoted as g_{id}. Based on the basic P2P protocol, g_{id} is mapped to an end-system node with the peer identifier that is numerically closest to g_{id}. An indirect service is then setup on this end system node. We refer to this end system node as the rendezvous point of the ESM service. The rendezvous node will forward all the ESM subscription messages to the service provider (the ESM source), who will ultimately inject the ESM data packets into the ESM overlay network through the rendezvous point.

Step 2: Managing Subscription. ESM service subscribers check those established multicast groups, and identify the services that they want to subscribe. All subscriber nodes connected themselves to the existing multicast group members. The links between the new subscriber and the existing ones will be used to forward multicast payload information and carry the other signal messages that are generated to maintaining the multicast tree.

The ESM service provider synchronizes with its rendezvous point for its direct children in the multicast group. The ESM payload is delivered in the reverse direction on the multicast tree, from the service provider down to the leaf peers. Upon receiving a multicast packet identified by a group identifier g_{id}, an end-system node invokes its local ESM interface multicast(), which will replicate this packet and forward it to all the peers appearing in the local group list identified by g_{id}.

When an end-system node decides to leave a multicast group, it will send a departure message to its parent in the end-system multicast tree. The operation p.unsubscribe(g_{id}, q) will first remove the callers node identifier q from the parent p's multicast group. The unsubscription request is forwarded in a cascade manner. If q is the last one in that multicast member list and p is no longer a member of that group, p will then delete that multicast group g_{id} from its multicast group list and call the unsubscribe of its own parent to remove itself from the multicast group.

An end-system node can participate in more than one end system multicast groups. Hence, it needs to maintain a list of multicast groups to distinguish downstream subscribers in the correspondent multicast group.

Step 3: Efficient Dissemination Using Mutlicast Groups. Finally, the multicast information is infused from the service provider and forwarded towards each subscriber through the links established and maintained by the ESM management substrate. In PeerCast, several mechanisms are employed to optimize the multicast hierarchy and maintain it against the system dynamics.

One of the unique features of our ESM maintenance protocol is the *Neighbor Lookup* technique. Using this technique, each peer initiating or forwarding the subscription request will first check its ESM overlay neighbors before it sends or forwards the request. Our landmark signature clustering scheme ensures higher likelihood that a peer can locate its physical network neighbors in its local neighbor list. Because the new subscriber could directly subscribe to its physical network neighbor, we can then take advantage of this local link and reduce the ESM traffic going across the Internet backbone. Figure 4 gives an example of how the neighbor lookup technique works. Before forwarding the subscription request to the next hop peer that satisfies the prefix matching, peer $S_{k,1}$ first check if its neighbor has already join the multicast group.

It finds peer $S_{k-1,1}$ that is already in the multicast tree. Thus, it directly subscribe to peer $S_{k-1,1}$. Similarly, peer $S_{n,1}$ is connected to its physical network neighbor $S_{n-1,1}$.

3.3.1 Maintaining ESM Multicast Overlay

To provide consistency multicast service against system dynamics such as peer departure, peer or link failure, or temporary inconsistency introduced by peer join, we need a mechanism to detect failure and restore forwarding route in the ESM hierarchy.

The status of each ESM forwarding link is maintains as a soft state on the multicast information receiver side. Each soft state is associated with a timer. A new heartbeat message or a multicast message with new sequence number resets the timer. Whenever there is no multicast message to deliver, an end-system node will send heartbeat message frequently enough to refresh its children's timers.

The timeout event indicates a broken upstream link in the multicast tree and will trigger a repair routine. The repairing is as simple as initiating a new subscription request, with the existing group id as the parameter. The P2P protocol of PeerCast guarantees that the routing information of P2P network will converge against peer failure or departure. Thus the broken link will be replaced with another one established by the new subscription request. To further minimize the repairing cost, we employ an optimistic repair scheme. The peer that detects a broken link will continue sending heartbeat messages to its children while it tries to establish a new upstream link or repair the old one. Downstream end-systems may experience temporarily service interrupt while assuming the link is still alive and wait for the further multicast message. In this manner, the impact of the broken link will be isolated from its subtree.

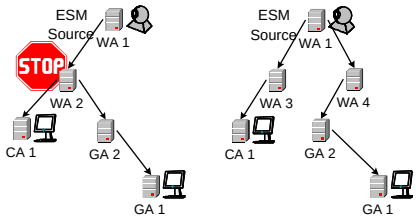

Figure 3: ESM overlay maintenance

Figure 3 shows how this mechanism works. In the original multicast tree, end-system WA2 failed due to some unknown reason. The children of WA2, i.e. CA1 and GA2, detect this event and connect to WA3 and WA4 respectively. Please note that after GA2 detects this failures, it did not stop sending message to its child GA1. Instead of forwarding the multicast information that has been interrupted, GA2 keeps sending heartbeat messages to GA1 until its multicast service is recovered. By doing so, we avoid the flashing of resubscription requests.

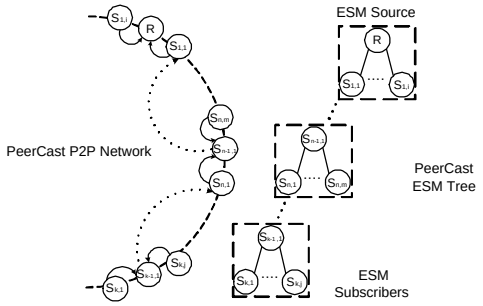

Figure 4: Improve PeerCast Overlay with Network Proximity Information

4 Load Balance in PeerCast

Balancing workloads among heterogeneous end-system nodes is vital for an ESM overlay to utilize its full capacity. In PeerCast, we tackle this problem with a technique called *Virtual Node*. Under this scheme, an end-system node joins the PeerCast ESM overlay in three steps.

First, it joins the P2P overlay with a set of identifiers generated with different random seeds. Each identifier represents a virtual end-system node that is allocated with a unit of resource. We assume that the end user can specify or estimate the resources it can contribute. We use the concept of ED (effective donation) [7] to estimate the resources that an end-system node can contribute. Each virtual node maintains its own routing table and neighbor list. An end-system node that contributes more resources will present itself at more location in the P2P identifier space, and will statistically receive more ESM subscription requests and handle more ESM forwarding workload.

In the second step, the end-system node joins a multicast group by starting the subscription process at one of its virtual nodes with an identifier numerically closest to the multicast source peer. Statistically, a more powerful end-system node should have higher probability to possess an identifier closer to the multicast source peer. In PeerCast system, the number of multicast forwarding hops between a subscriber and the multicast root is related to the numerical distance between their identifiers. Hence the closer is an identifier to the multicast source, the less delay will the ESM service experience. More importantly, in PeerCast, we design the routing information to be shared among all the virtual nodes owned by the same physical end system node. If we envision the routing information entries as the connections a peer establishes to a subset of other peers, sharing the routing information among its virtual nodes would increases the connectivity among peers and thus further improves the quality of the ESM forwarding path.

To prevent the duplicate forwarding, we design the multicast group member list as an object shared among all the virtual nodes belong to the same end-system node. Since each end-system node is identified by its IP address and port number in the multicast group list, duplicate subscription can be

removed from any multicast group list in our system. Subscription requests forwarded through an end-system node with multiple virtual identifiers may introduce multiple multicast forwarding paths in the overlay. The co-existence of these forwarding paths gives those powerful end-system nodes extra opportunities to optimize the quality of their own ESM forwarding paths. By choosing the path with shortest forwarding latency, nodes devoting more resources would be placed closer to the multicast root, and thus could receive better service. In order to measure the forwarding path latency and choose the shortest path to receive ESM payload, each end-system measures the latency of the unicast link between itself and its multicast parents. Under the assumption that the ESM forwarding latency is the major part of the link latency, the accumulated link latency on each forwarding path could be used to approximate of the latency of ESM forwarding path, and guide the end-system to decide which path to take.

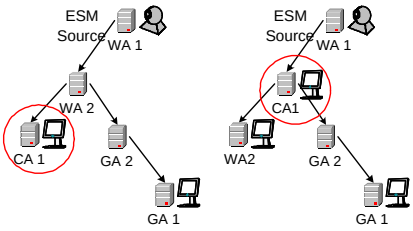

Figure 5: Bottleneck Removal

Finally, each end-system node maintains its ESM subscription status following the multicast overlay maintenance protocol and uses the *bottleneck removal* technique to optimize the ESM overlay. Each end-system node in our system periodically probes its child nodes in the ESM tree and chooses the one with most available resources as its potential replacement. Whenever a node detects that its potential replacement has more available resources than itself, our bottleneck removal protocol force it to exchange position with that potential replacement child in the ESM tree, transparent to the end-system users. Thus, nodes contributing more resources will gradually be "promoted" towards the root of the ESM tree and obtain better ESM service than the end-system nodes deep down the ESM forwarding path. Figure 5 gives an example of how this technique works. In the ESM overlay composed of end-system nodes {CA1, GA1, GA2, WA1, WA2}, WA2 probes its children CA1 and GA1. As WA2 detects that CA1 has more available resources than itself, it initiates the promotion of CA1 and switch its position to be the child of CA1.

5 Reliability in PeerCast

5.1 Departures and Failures

We identify two types of events that depart an end-system node from ESM overlay. A *proper departure* in PeerCast is a volunteer disconnection of a peer from the PeerCast overlay. During a proper departure, the PeerCast P2P protocol updates its routing information. In addition, if there is no ESM service replication mechanism employed in the system, the PeerCast application on the departing peer notify its children in the ESM overlay to initiate the re-subscribing process. Such a scenario is relatively slow and less efficient since each child of the departing peer will individually initiates a subscription request. A *failure* in PeerCast is a disconnection of a peer from the network without notifying the system. This can happen due to a network problem, computer crash, or improper program termination. Failures are assumed to be detectable (a fail-stop assumption), and are captured by the PeerCast P2P protocols neighbor list polling mechanisms. However, in order to recover the lost multicast service promptly with less re-subscription overhead, a replication mechanism is needed. Notice that once there is a replication mechanism, which will enable the continuation of the multicast service from the replicated copies, then the proper departures are very similar to failures in terms of the action that needs to be taken. This will enable the elimination of the explicit re-subscription process during departures. The main difference between a proper departure and a failure is that, a properly departing peer will explicitly notify other peers of its departure, whereas the failure is detected by the P2P protocol. In the rest of the paper, we use the term departure to mean either proper departure or failure.

5.2 Service Replication Scheme

The failure of an end-system node will interrupt the ESM payload it receives and forwards to its children. In order to recover the interrupted multicast service without explicit re-subscription, each end-system node needs to replicate the multicast group information among a selection of neighbors. The replication scheme is dynamic. As peers join and depart the ESM overlay, replications are migrated such that there are always a certain number of replica active, which is a desirable invariable that we want to maintain. The replication involves two phases. The first phase is right after the ESM group information is established on a peer. Group information replicas are installed on a selection of peers. After replicas are in place, the second phase keeps those replicas in consistency as end-system nodes subscribe to or leave the ESM group. We denote this phase as the *replica management* phase.

Given an ESM group identified by identifier g, its group information on a peer p with identifier i is replicated on a set of peers denoted as $ReplicastionList(g, i)$. We refer this set as the *replication list* of group g. The size of the replication list is r_f, where r_f is referred as the *replication factor* and is a system parameter. To localize the operation on the replication list, we demand that $r_f \leq 2 \cdot r$, which means all the replica holders $ReplicastionList(g, i)$ in are chosen from the neighbor list $NeighborList(p, i)$ of peer p.

For each ESM group g that a peer p is actively

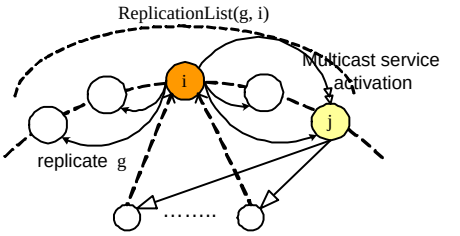

Figure 6: Multicast Service Replication with $r_f = 4$

participating, peer p will forward the replication list $ReplicationList(g, i)$ to its parent end-system node $parent(g, i)$ in group g. Once p depart from group g, its original parent $parent(g, i)$ will use $ReplicationList(g, i)$ to identify another peer q with identifier j to take over the ESM multicast forwarding work of p. q will use the group information that p put on it to carry out the ESM payload forwarding for group g. We say that q is "activated" in this scenario. Once q is activated, it will use its neighbor list $NeighborList(q, j)$ to setup the new $ReplicationList(g, j)$, and use it to replace $ReplicationList(g, i)$ on $parent(g, i)$, which is equivalent to $parent(g, j)$ now.

Our replication scheme is highly motivated by the passive replication scheme of [1]. The active participant of an ESM group acts as the 'primary server' and the peers holding replicas as the 'backup servers'. However, our scheme is difference in that the active peer could migrate its ESM tasks when it discovers a better candidate to do the job in terms of load balancing or efficiency.

5.3 Replica Management

In this section, we explain how the described dynamic replication scheme is maintained as end-system nodes subscribe or depart from the ESM system. Since the active replication scheme works for both peer departure and failure case, we use the term departure to refer to both scenarios. For the purpose of brevity, we assume that the replication factor r_f is equal to $2r$. In case that r_f is less than $2r$, our arguments still hold with some minor modifications to the description.

When a multicast group g is added to the multicast group list on a peer with identifier i, it is replicated to the peers in the $ReplicationList(g, i)$. PeerCast P2P protocol detects the later peer entering and departure event fallen within $NeighborList(p, i)$. Once such an event happens, an upcall is triggered by the P2P management protocol, and the replica management protocol will query the peers in $NeighborList(p, i)$ and update the replication list $ReplicationList(g, i)$. We describe the reaction that a peer will take under different scenarios.

Peer Departure. A peer's departure triggers the update of $2r$ neighbor list. Once a peer p with identifier i receives the upcall informing the departure of peer p', it will perform the following actions:

- For each group g that p is forwarding ESM payload, p adds p'', which is added into $NeighborList(p, i)$ by the P2P management protocol, to the replication list $ReplicationList(g, i)$.
- For each group g that p is forwarding ESM payload, p removes the departing peer p' from the replication list $ReplicationList(g, i)$.
- For each group g that p is forwarding ESM payload, p sends its group information to p''.
- For each group g that p is forwarding ESM payload, p sends the updated replication list $ReplicationList(g, i)$ to its parent peer in multicast group g.

Peer Entrance. A peer's entrance also triggers the update of $2r$ neighbor list. Once a peer p with identifier i receives the upcall informing the entrance of peer p', it will perform the following actions:

- For each group g that p is forwarding ESM payload, p adds p', to the replication list $ReplicationList(g, i)$.
- For each group g that p is forwarding ESM payload, p removes peer p'', which is removed from $NeighborList(p, i)$ due to the entrance of p', from the replication list $ReplicationList(g, i)$.
- For each group g that p is forwarding ESM payload, p sends its group information to p' as replicas.
- For each group g that p is forwarding ESM payload, p sends the updated replication list $ReplicationList(g, i)$ to its parent peer in multicast group g.

Updating Replicas. As end-systems subscribe or unsubscribed from ESM groups, their subscription or unsubscription requests will be propagated up through the ESM tree and change the group information on some peers. Once the group information of group g is changed on peer (p, i). p sends its group information to all the other peers in $ReplicationList(g, p)$.

5.4 Reliability Analysis

Assume a peer p with identifier i departs the ESM overlay at time t_d, and it takes a constant time interval Δr to recover the lost ESM service of group g , we want to know the probability that the ESM service could be properly recovered. In another word, we want to know the probability that the replica holders in $ReplicationList(g, i)$ all fail during recovering interval Δr, which we denote as $Pr_f(p, i)$. We denote this event as the *service interruption* caused by p's departure.

We assume the time before each peer departs by failing follows independent and identical exponential distribution with

parameter λ_f, and the life time of each peer in the ESM overlay follows independent and identical exponential distributions with parameter λ_d. Thus the turnover time of each peer, which is the time before each peer depart the system by failure or proper departure, also follows independent and identical distribution with parameter $\lambda = \lambda_d + \lambda_f$. The mean active time s of a peer in ESM overlay is equal to $1/\lambda$, which we refer as the service time in our later analysis. The probability that a peer departs by failure is thus $\frac{\lambda_f}{\lambda_d+\lambda_f}$, and the probability that a peer departs by proper departure is thus $\frac{\lambda_d}{\lambda_d+\lambda_f}$.

We use random variables L_1, L_2, ..., L_{r_f} to denote the amount of time that replica holders in $ReplicationList(g, i_p)$ stay active in the network after peer p's departure at time t_d. By the memorylessness property of exponential distribution, we know that L_1, L_2, ..., L_{r_f} still follow the exponential distribution with parameter λ. We thus have:

$$\begin{aligned} Pr_f(p,i) &= (\frac{\lambda_f}{\lambda_d+\lambda_f})^{r_f+1} \cdot Pr\{\text{MAX}(L_1, L_2, \ldots, L_{r_f+1}) \\ &\quad -\text{MIN}(L_1, L_2, \ldots, L_{r_f}, L_{r_f+1}) < \Delta r\} \\ &= (\frac{\lambda_f}{\lambda_d+\lambda_f})^{r_f+1} \cdot Pr\{\text{MAX}(L_1, L_2, \ldots, L_{r_f}) < \Delta r\} \\ &= (\frac{\lambda_f}{\lambda_d+\lambda_f})^{r_f+1} \cdot \prod_{i=1}^{r_f} Pr\{L_i < \Delta r\} \\ &= (\frac{\lambda_f}{\lambda_d+\lambda_f})^{r_f+1} \cdot (1-e^{-(\lambda_d+\lambda_f)\cdot\Delta r})^{r_f} \end{aligned} \quad (1)$$

p owns a set of identifiers $p.ids$ by our virtual node scheme. We assume that there is no overlapping among the replication lists of p's different identifiers, and $\forall_{i,j \in p.ids} Pr_f(p,i) = Pr_f(p,j)$. The probability that p's departure causes any service interruption can be expressed as:

$$\begin{aligned} Pr_f(p) &= 1 - \prod_{i \in p.ids} (1 - Pr_f(p,i)) \\ &= 1 - (1 - Pr_f(p,i))^{E[|p.ids|]} \\ &= 1 - (1 - (\frac{\lambda_f}{\lambda})^{r_f+1} \cdot (1-e^{-\lambda\Delta r})^{r_f})^{E[|p.ids|]} \end{aligned} \quad (2)$$

We use the turnover rate of [13] to approximate λ. As reported in [13], half of the peers participating in the P2P system will be replaced by new peers within one hour. We have $Pr\{\text{a peer depart in an hour}\} = 0.5$, which indicates $1 - e^{-\lambda \cdot 60} = 0.5$ and $\lambda = 0.0116$. The mean service time $s = 1/\lambda = 86.56$ minutes. When we setup our system as $r_f = 4$, $\Delta r = 6$ secs, and $E[|p.ids|] = 4$, we have $Pr_f(p) \simeq 7.2258e-012$. In a setup with $r_f = 2$, $\Delta r = 60secs$, and $E[|p.ids|] = 4$, we have $Pr_f(p) \simeq 5.3193e-004$, with which the resubscription of p's children will be triggered

6 Experimental Results

We have designed a simulator that implements the mechanisms explained in this paper. In the following subsections, we investigate four main subjects using results obtained from experiments carried out on our simulator. We first study effect of landmark signature technique on clustering end-system nodes by their network proximity. Then, we evaluate how the efficiency of end-system multicast tree could be improved using the network proximity information. In the third set of experiments, we study the multicast workload distribution under the virtual node scheme. Finally, we study the effect of our replication scheme on recovering multicast service under various node failure scenarios.

We used the Transit-Stub graph model from the GT-ITM topology generator [16] to generate a set of network topologies and to simulate the PeerCast overlay networks with different parameter settings. Each topology consists of 5150 routers. The link latencies are assigned values using a uniform distribution on different ranges according to the type of the link: [15ms, 25ms] for intra-transit domain links, [3ms, 7ms] for transit-stub links, and [1ms, 3ms] for intra-stub links. End-system nodes are randomly attached to the stub routers and organized into P2P network following the PeerCast P2P protocol.

We used the routing weights generated by the GT-ITM topology generator to simulate the IP unicast routing. IP multicast routes are simulated by merging the unicast routes from the source to each subscriber into a shortest path tree.

6.1 Landmark Signature and Neighbor Lookup Scheme

One of the concerns on using the landmark signature technique is whether it will incur any side-effects since it biases the node identifier distribution. In this section we study three different flavors of the PeerCast system: the one without landmark signature technique, the one with landmark signature technique only, and the one with both landmark signature technique and the neighbor lookup scheme.

We simulate a P2P network with $5*10^4$ peers. The number of peers in the multicast group varies from $1*10^4$ to $4*10^4$. We set the value of r to 8 and use 16 landmark points to minimize the inaccuracy of the landmark signature technique so that we can focus our efforts on comparing different schemes. The landmark signature is inserted into the peer identifiers at different offset after identifier digits 0, 1, and 2.

6.1.1 Delay Penalty

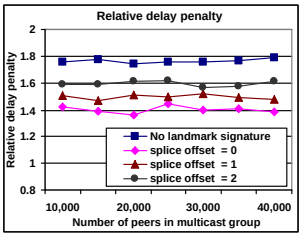

Figure 7: Relative delay penalty, using only landmark signature

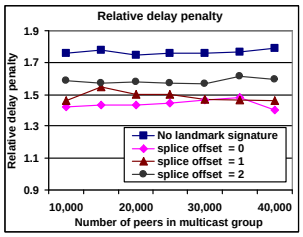

Figure 8: Relative delay penalty, using landmark signature and neighbor lookup

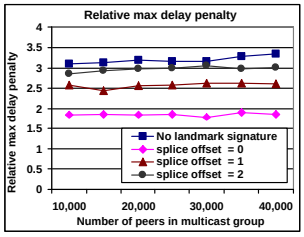

Figure 9: Max delay penalty, using only landmark signature

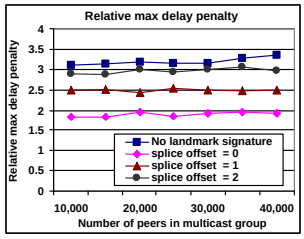

Figure 10: Max Delay Penalty, using landmark signature and neighbor lookup

We first compare the message delivery delay of IP multicast and PeerCast. ESM increases the delay of message delivery relative to IP multicast because of the multi-hop unicast message replication and forwarding. We use two metrics to evaluate the delay penalty. *Max delay penalty* is the ratio between the maximum delay using PeerCast and the maximum delay using IP multicast. *Relative delay penalty* is the ration of average PeerCast delay and the average delay using IP multicast.

The experiment results show that, using the landmark signature technique alone can reduce the delay penalty. The landmark signature technique and our end-system multicast management protocol put the multicast roots network neighbors close to it in the multicast tree. Thus, the multicast delay of any nodes that receives multicast information through those nodes will have less delay penalty.

However, increasing the value of splice offset will offset the benefit of landmark signature technique. As the randomness of peer identifier distribution increases, the P2P network with landmark signature degrades to the normal P2P network. In this case, the larger value of the splice offset puts the relative delay penalty and the max delay penalty value closer to the ones of PeerCast with no landmark signature.

6.1.2 Link Stress

Link Stress is the ratio between the number of IP messages generated by a PeerCast multicast tree and the number of messages generate by the equivalent IP multicast tree. We ignore the signal messages of PeerCast and IP multicast tree to focus on the messages that carry only the multicast content payload.

The randomness of identifiers distribution is violated by insert the landmark signature at the very beginning of the peer identifiers. The first hops of lookup requests will have to traverse the inter-networks through more hops because the request initiators physically network neighbors are now located around itself on the P2P identifier space, while the target is belong to another sub-domain of the network. On the other hand, the routing paths are less likely to converge because they are more likely originated from different sub-networks, which are now represented by identifier clusters on the different sections of the identifier circle. Hence, we observe that in Figure 11 and Figure 12, the serials with splice offset = 0 have higher link stress than the other cases. As we introduce more randomness into the identifier distribution, we force the division of peer clusters and put them into different section on the identifier overlay. Thus, the link stress drops to the same level as there is no landmark technique and no neighbor lookup.

When we use the neighbor lookup scheme to improve the efficiency of our multicast overlay, we reduce the multicast forwarding routes travels through the inter-network links by putting more forwarding hops among physical network neighbors. This mechanism reduces the link stress compared to the PeerCast system without neighbor lookup scheme.

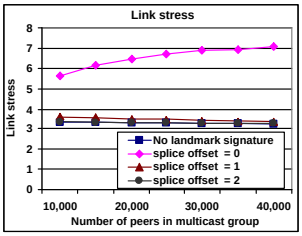

Figure 11: Link Stress, using only landmark signature

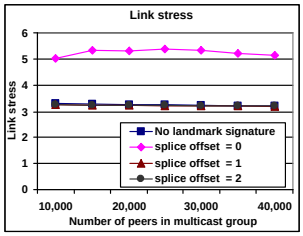

Figure 12: Link Stress, using landmark signature and neighbor lookup

6.1.3 Node Stress

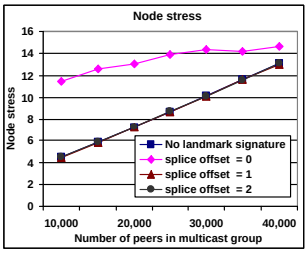

Figure 13: Node Stress, using only landmark signature

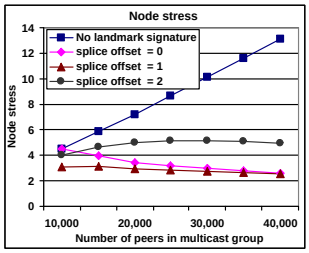

Figure 14: Node Stress, using landmark signature and neighbor lookup

End-system nodes in PeerCast handles jobs like the maintenance of multicast group membership information, and the replication and forwarding multicast messages. We use *node stress* to evaluate such extra workload on end-system nodes. The value of node stress is the average number of children that non-leave end-system nodes handle.

The way we generate landmark signature reduce the dimension of identifier space by a factor of $L!/2^{4L}$, where L is the number of landmark points. When we splice the landmark signature at the very beginning of each identifier, we reduce the number of peers that sharing the same prefix as the multicast source. Subscription requests are then more likely be forwarded through fewer peers in the overlay, and result in higher node stress. That is why the PeerCast flavor with landmark signature splice point 0 has such a higher node stress in Figure 11. When we introduce randomness into the identifier

space, we reduce the high node stress by increase the randomness of identifier distribution. Thus, we see in Figure 11 that peers with splice offset of value 1 or 2, link stresses are closer to the PeerCast scheme without landmark signature. As the number of end-system nodes in the multicast group increases, in average a peer will handle more forwarding links, thus the average node stress increases accordingly.

Using the neighbor lookup technique, a peer first trying to leverage its local network neighbors before it subscribes to the multicast service. Because the landmark signature technique rendering high probability that a peer can identify a network neighbor in its local neighbor list, we observe in Figure 12 that PeerCast overlay with both features enabled have much lower node stress in comparison to the original PeerCast scheme. As the number of peers in the multicast group increases, the chance that a peer could subscribe to its local peers increases too. The result is decreasing node stresses as the number of peers in the multicast group increases.

6.2 Balancing Load of Heterogeneous Nodes

To evaluate the effects of virtual node technique, we assign the end-system nodes in our simulation with different capacities. Due to the complexity of our experiment, we simplified our model by assuming that 20% end-system nodes possess 8 units of capacity, 30% end-system nodes have 4 units of capacity, and the rest 50% end-system nodes own single unit capacity only. We are interested in the load distribution among end-system nodes with different capacities. Also, we are interested in the relative delay penalty of end-system nodes donating different amount of resources.

We setup two kinds of ESM overlays. One set of ESM overlays are built over P2P networks with fixed number of peers $5 * 10^4$, which models shared P2P networks. Another set of ESM overlay are built over P2P networks that have the same set of end-systems as the ESM overlays. This model simulates exclusive P2P networks where only multicast service subscribers or provider participate the specific P2P network. We varies the number of end-systems in the ESM multicast overlay from 5 to $5 * 10^4$. The node stress and the relative delay penalty are recorded and grouped by the capacity of peers.

Figure 15 and Figure 16 plots the average node stress of overlay system built with virtual node promotion technique. As plotted in both Figure 15 and Figure 16, virtual node technique could match the workload to peers capacity. In Figure 15, where the multicast group members are chosen among a shared overlay, the node stresses of different peer groups present less significantly differences because the less end-system nodes are overloaded. However, as recorded in Figure 16, as the size of the multicast group grows close to the size of the overlay, the virtual node technique plays a vital role in balancing the workload. Because the basic PeerCast protocol does not distinguish the capacity of end-system node in forwarding the subscription requests, the randomness of the node identifiers causes a number of nodes with high performance be placed close to the leaves of the ESM tree, whereas some less capable nodes are placed close to the root. The node promotion technique we discussed in section 4 solve this problem by promoting nodes with more capacity to handle more forwarding workload, and moving the potential bottleneck nodes down to the leaves of the tree. As plotted in both Figure 15 and 16, this feature effectively solves this problem.

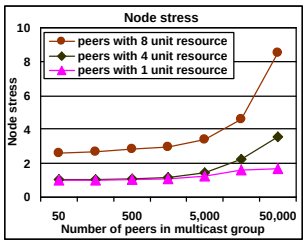

Figure 15: Average node stress, r = 8 peers number = 50,000

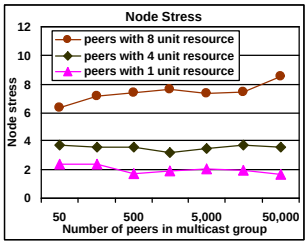

Figure 16: Average node stress, r = 8 peers number = multicast group size

One appealing feature of virtual node technique is that the end-system nodes that contribute more resources will be placed closer to the multicast source. This feature not only assigns more workload on those more powerful peers, it also awards them with better multicast service such as the lower relative delay penalty. We records the relative delay penalties of peers with different level of capacities in Figure 17 and Figure 18. We can see that the peers with more units of capacities are always with lower delay penalty.

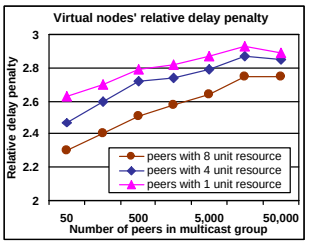

Figure 17: Relative delay penalty, r = 8 peers number = 50,000

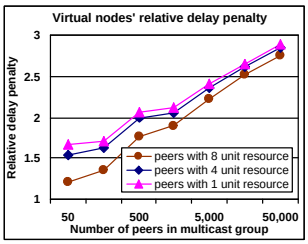

Figure 18: Relative delay penalty, r = 8 peers number = multicast group size

6.3 Replication

One of the situations that is rather crucial for the PeerCast system is the case where the peers are continuously leaving the system without any peers entering; or the peer entrance rate is too low than the peer departure rate such that the peers present in the system decreases rapidly. To observe the worst case, we setup our simulation with $4 * 10^4$ peers, among which $2 * 10^4$ participate the ESM overlay. Each peer departs the system by failing after certain amount of time. The time each peer stays in the system is taken as exponentially distributed random variable with mean *st*, which indicate the *service time* of

a peer in the overlay. It is clear that failure of peers will trigger the re-subscription process and cause the service interruption to its downstream peers in the multicast tree. However, we want to observe the behavior of our replication scheme with different r_f values and see how the ESM service in PeerCast can be recovered without going through the expensive re-subscription.

The graphs in Figures 19 and Figure 20 plot the total number of deadly failure that have occurred during the whole simulation for different mean service times (st), recovery times (Δt_r), and replication factors (r_f). These graphs show that the number of dealy failures is smaller when the replication factor is larger, the recovery time is smaller and the mean service time is longer. Note that our simulation represents a worst scenario that every peer leaves by failure and no peer enters into the system. However, with a replication factor of 3, the number of deadly failure is negligible.

There experiment shows that, although the cost of replication grows with the increasing replication factor, the dynamic replication provided by PeerCast is able to achieve reasonable reliability with moderate values for the replication factor.

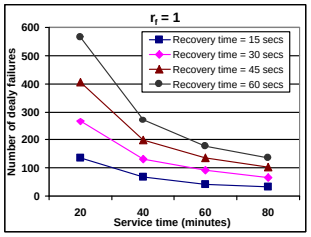

Figure 19: Deadly failure, r_f = 1

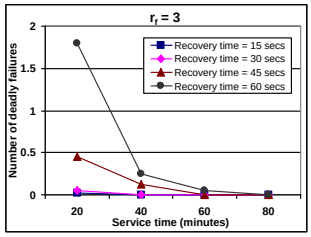

Figure 20: Deadly Failure, r_f = 3

We measure the overhead of recovering multicast service under peer failure by the number of messages exchanged after a failure is detected. A deadly failure on a peer causes the timers of its downstream peers to expire. In PeerCast, those downstream peers will re-establish their multicast services by re-subscribe themselves into the ESM overlay. And all these messages will be counted since they will increase the duration of the service interruption of those effected peers. On the other hand, if a peers service replication is activated when it fails, its children will experience less multicast service interruption, since only an fast activation message is involved to activate the service replication, in comparison to the flashing of the re-subscription requests.

We observe the number of messages exchanged under the same experiment configurations of Figure 19. We count the total number of messages generated for the replication activation and re-subscription. The results in Figure 21 conforms to the curves of Figure 19. When the number of deadly failure increases, more messages are generated for the re-subscription requests. However, most of the messages are for the replication activation, since in most of the case, the services are restored by the replication activation.

To evaluate the effect of the replication scheme on reducing the messaging overhead for multicast service restoring, we compared the replication scheme to the scenarios with no service replication. We measures multicast groups with $1 * 10^4 \sim 4 * 10^4$ peers built over P2P network with $5 * 10^4$ and $8 * 10^4$ peers. The replication scheme is setup with $r_f = 1$ and the peer service times follow exponential distribution with mean 20 minutes. This is the worst case that will generate the most number of re-subscription requests among all our experiment configurations. The experiment results are plotted in Figure 22. While the number of peers in the multicast group and the P2P network increases, the overhead of service recovering increases almost linearly. However, we observe that the number of messages involved for restoring services is far less when the service replication scheme is used. With the overhead of maintaining ONE replication, we reduce the multicast service restoring overhead by 62.3% to 73.8%. In actual implementation of PeerCast, we can piggyback the replication maintenance message to the neighbor list probing message of P2P network management protocol, and further reduce the overhead of service replication scheme.

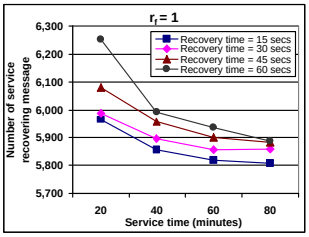

Figure 21: Number of service recovering messages under replication scheme, $r_f = 1$

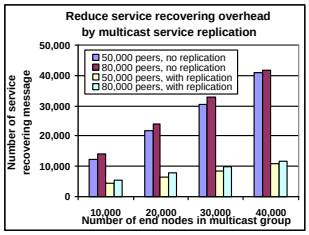

Figure 22: Number of service recovering messages, $\Delta t_r = 15$secs, $st = 20$ minutes, replication scheme has r_f = 1

7 Related Work

EMS protocols like [5, 6, 9] are developed primarily for network of relatively small size. A few end system nodes are responsible for the management functionalities such as gathering and analyzing information using certain network measurement techniques [3]. In a system with a large number of end-system nodes, such management overhead may overload those management nodes. The NICE [2] protocol builds a multicast overlay in the form of a hierarchical control topology. A top-down approach is used to forward the joining request recursively from leader nodes into the best lower layer clusters. The Overcast protocol [9] uses a distributed protocol to create a distribution tree, with the multicast source as the root. It uses end-to-end measurements to optimize bandwidth between the root and the various group members. The Narada [6] protocol applies a two-step approach to build the multicast tree. It first generates a mesh network among all members and then uses a centralized algorithm generate the multicast tree. The mesh network is dynamically adjusted in

accordance to the utilities and cost estimation to maintain the proper of the multicast tree. The approach taken by PeerCast differs from these existing researches in two aspects. First, PeerCast presents scalable solution for ESM with a heterogeneous overlay network. Second, PeerCast system offers reliability guarantee through replication.

Recent studies in Peer-to-Peer (P2P) network [10, 12, 14, 17] present a new orient to address the issue of managing large-scale multicast overlays. In Bayeux [18] system, the joining request to an existing root is forwarded to the root node first, from where a reverse message is routed back to the new member, using the Tapestry [17] protocol. The nodes on this routing path will record the new member's identity and include it into the forwarding path. The Content Addressable Network (CAN) [10] adopts landmark technique to partition the Cartesian space into equally sized bins. Each node measures its distance to well-known landmark hosts to decide the bin to join. The multicast system described in [11] exploits the structure of the CAN coordinate space and limits the forwarding of each message to only a subset of a node's neighbor. The stretch of the forwarding path is decided by the number of dimensions of the Cartesian's space. And the unbalanced node distributed caused by the landmark partitioning technique may overload a node easily. The PeerCast basic protocol is highly inspired by Scribe [4]. However, our landmark signature scheme captures the network proximity information more precisely. In addition, we use techniques such as neighbor lookup to further improve the end-system multicast overlay. Another distinct feature of PeerCast is its ability to provide scalable and reliable solution for ESM with a heterogeneous overlay network.

8 Conclusion

We have presented PeerCast, a reliable and self-configurable peer to peer system for application-level multicast. Our approach has three unique features compared to existing approaches to application-level multicast systems.

First, we propose a capacity-aware overlay construction technique based on the concept of virtual peer identifiers to balance the multicast load among peers with heterogeneous capabilities.

Second, we utilize the landmark signature technique to cluster peer nodes of the ESM overlay network, aiming at exploiting the network proximity of end system nodes for efficient multicast group subscription and fast dissemination of information across wide area networks.

Third and most importantly, we develop a dynamic passive replication scheme to provide reliable subscription and multicast dissemination of information in an environment of inherently unreliable peers. An analytical model is presented to discuss its fault tolerance properties. We evaluate PeerCast using simulations of large scale networks. The experimental results indicate that PeerCast can provide efficient multicast services over large-scale network of heterogeneous end-system nodes, with reasonable link stress and good load balance.

References

[1] P. Alsberg and J. Day. A principle for resilient sharing of distributed resources. In *Proceeding of ICSE*. IEEE Computer Society Press., 1976.

[2] S. Banerjee, B. Bhattacharjee, and C. Kommareddy. Scalable application layer multicast. In *Proc. of ACM SIGCOMM*, 2002.

[3] S. Banerjee, C. Kommareddy, K. Kar, B. Bhattacharjee, and S. Khuller. Construction of an efficient overlay multicast infrastructure for real-time applications. In *Proceedings of INFOCOM*, 2003.

[4] M. Castro, P. Druschel, A. Kermarrec, and A. Rowstron. SCRIBE: A large-scale and decentralized application-level multicast infrastructure. *IEEE Journal on Selected Areas in communications (JSAC)*, 2002.

[5] Y. Chawathe. *Scattercast: An Architecture for Internet Broadcast Distribution as an Infrastructure Service*. PhD thesis, University of California, Berkeley, 2000.

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

[7] B. Gedik and L. Liu. Peercq: A scalable and self-configurable peer-to-peer information monitoring system. Technical Report GIT-CC-02-32, Georgia Institute of Technology, Feburary 2002.

[8] B. Gedik and L. Liu. Peercq: A decentralized and self-conguring peer-to-peer information monitoring system. In *Proc. of ICDCS*, 2003.

[9] J. Jannotti, D. K. Gifford, K. L. Johnson, M. F. Kaashoek, and J. W. O. Jr. Overcast: Reliable multicasting with an overlay network. In *Proceedings of OSDI*, 2000.

[10] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A scalable content addressable network. In *Proc. of ACM SIGCOMM*, 2001.

[11] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker. Application-level multicast using content-addressable networks. *Lecture Notes in Computer Science*, 2233, 2001.

[12] A. Rowstron and P. Druschel. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. *Lecture Notes in Computer Science*, 2218, 2001.

[13] S. Saroiu, P. K. Gummadi, and S. D. Gribble. A measurement study of peer-to-peer file sharing systems. In *Proceeding of MMCN' 02*, January.

[14] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan. Chord: A scalable Peer-To-Peer lookup service for internet applications. In *Proc. of ACM SIGCOMM*, 2001.

[15] Z. Xu, C. Tang, and Z. Zhang. Building topology-aware overlays using global soft-state. In *Proc. of ICDCS*, 2003.

[16] E. W. Zegura, K. L. Calvert, and S. Bhattacharjee. How to model an internetwork. In *IEEE Infocom*, volume 2, pages 594–602. IEEE, March 1996.

[17] B. Zhao, J. Kubiatowicz, and A. Joseph. Tapestry: An infrastructure for fault-tolerant wide-area location and routing. Technical report, U. C. Berkeley, 2002.

[18] S. Zhuang, B. Zhao, A. Joseph, R. Katz, and J. Kubiatowicz. Bayeux: An architecture for scalable and fault-tolerant widearea data dissemination. In *Proc. NOSSDAV*, 2001.