Reliable Peer-to-peer End System Multicasting through Replication

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

Abstract

A key challenge in peer-to-peer computing system is to provide decentralized and yet reliable service on top of a network of loosely coupled, weakly connected and possibly unreliable peers. This paper presents an effective dynamic passive replication scheme designed to provide reliable multicast service in PeerCast, an efficient and self-configurable peer-to-peer End System Multicast (ESM) system. We first describe the design of a distributed replication scheme, which enables reliable subscription and multicast dissemination of information in an environment of inherently unreliable peers. Then we 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 provide group communication functions for applications like event and content distribution, audio and video conference, and multi-user games. Peer-to-peer (P2P) ESM has emerged as a promising distributed ESM paradigm. A P2P ESM system uses the functions provided by the P2P protocols and organizes end-system nodes into ESM overlays. The unicast links interconnecting end-system nodes carry ESM control messages and group communication payloads.

A few issues have to be addressed in supporting reliable end-system multicasting with such a decentralized environment as a P2P network.

First, It is widely recognized that large scale peer-to-peer systems presents highly dynamic peer turnover rate. As reported in [16], half of the peers participating in the system will be replaced by new peers within one hour in both Napster and Gnutella. Because ESM systems rely on end-system nodes to replicate and forward ESM payloads, the failure or departure of end-system nodes would cause the loss of subscription information and the interruption of multicast services. Thus, maintaining fault-tolerance in such a highly dynamic environment is critical to the success of a peer-to-peer ESM system.

Second, a peer-to-peer ESM system usually disseminates information through an overlay network of end-system nodes interconnected by unicast links. A critical issue for peer-to-peer ESM is to improve the efficiency of the system in term of reducing the traffic across the wide area overlay network and minimizing the multicast latency experienced by end users. Recent efforts in peer-to-peer ESM systems have been contributed towards addressing the second issue [4, 5, 6, 11, 14, 22]. It is widely recognized that further deployment of P2P technology for applications like end-system multicast demands practical solutions to the first issue.

To address both issues, we design an passive replication scheme for PeerCast [20], an efficient and self-configurable ESM system, to provide reliable ESM service on top of a network of loosely coupled, weakly connected and possibly unreliable peers. Our approach has two features that distinguish it from the existing approaches to application-level multicast.First, we develop a dynamic passive replication scheme to provide reliable subscription and multicast 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 replication scheme of PeerCast. Second, we develop an effective node clustering technique based on the landmark signature technique, which can cluster end-system nodes by exploiting their physical network proximity for fast multicast group subscription and efficient dissemination of information across wide area networks.

2 PeerCast System Overview

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 a PeerCast overlay network is equipped with a PeerCast middleware, which is composed of two-tier substrates: *P2P Network Management* and *End System Multicast Management*.

2.1 Peer-to-peer Network Management Protocol

The P2P network management substrate is the lower tier substrate for P2P membership management, lookups, and communication among end-system nodes. It consists of *P2P membership protocol* and *P2P lookup protocol*.

PeerCast system uses the P2P membership protocol to organize the loosely coupled and widely distributed end-system nodes into a P2P network that carries the multicast service. PeerCast peer-to-peer network is a Distributed Hash Table (DHT) based structured

P2P network. 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 and are generated to be uniformly distributed by using hashing functions like MD5 and SHA-1. Each identifier is composed of $\lceil m/b \rceil$ digits with m bits each. $peer\_props$ is a composite attribute which is composed of several peer properties, including IP address of the peer, resources such as connection type, CPU power and memory, and so on.

Identifiers are ordered on an m-bit identifier circle modulo 2^m, in a clockwise increasing order. The distance between two identifiers i and j is the smallest module-2^m numerical difference between them. Identifier i is considered as "numerically closest" to identifier j when there exists no other identifier having a closer distance to j than i. Given an m-bit key, the PeerCast protocol maps it to a peer whose peer identifier is numerically closest to that key.

A peer p invokes its local function p.lookup(i) to locate the identifier j that is numerically closest to i. The lookup is performed by routing the lookup request hop-by-hop towards its destination peer using locally maintained routing information. Each hop, the lookup request is forwarded to a peer sharing at least one more identifier digit with i. In a P2P system composed of N peers, the forwarding is of $O(\log_{2^b} N)$ hops.

Each identifier possessed by a peer is associated with a *routing table* and a *neighbor list* . The routing table is used to locate a peer that is more likely to answer the lookup query. It contains information about several peers in the network together with their identifiers. A neighbor list is used to locate the owner peer of a multicast service and the replication peers of the multicast subscription information. The neighbor list points to immediate neighbors on the identifier circle. 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 locality argument of our P2P protocol. Readers who are interested may refer to [8].

Network Proximity Awareness in PeerCast. A unique feature of PeerCast P2P management protocol is that it takes into consideration of the network proximity of end-system nodes, when organizing them into ESM overlays. This feature ensures the efficiency of the multicast services built over the P2P network, and distinguishes PeerCast from the other existing ESM scheme.

Our basic P2P network shares with Pastry [15] of the same problem known as "logarithmical deterioration of routing". The length of each lookup forwarding hop increases logarithmically as a request is forwarded closer to its target peer, as shown in the example of Figure 1.

In PeerCast, we propose a scheme named "Landmark Signature Scheme" to tackle this problem. A set of randomly distributed end-system nodes are chosen as the *landmark points*. The distances of an end-system node to these landmark points are recorded into a vector named as *landmark vector*. The intuition behind our scheme is that physical network neighbors will have similar landmark vectors. We use this similarity information to twist the numerical distribution of peer identifiers such that peers physically closer will have numerically closer identifiers. Concretely, a new peer obtains a set of landmark points through the bootstrapping service when it joins the P2P network. Using this set of landmark points, the new peer generates its *landmark signature* by encoding the relative order of landmark vector elements into a binary string. The landmark signature is then inserted into its identifiers at a certain offset called *splice offset*. As the new peer joins our P2P network, it aligns itself along with the other peers that have similar landmark signatures.

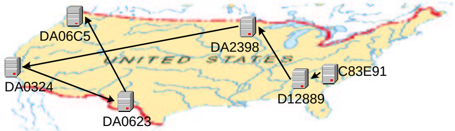

Figure 1. Logarithmical deterioration of routing in structured P2P network

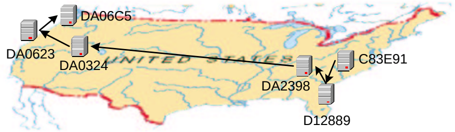

Figure 2. Routing regarding network proximity in PeerCast P2P network

Using this scheme, PeerCast system can bound more lookup forwarding hops to be within each others network vicinity, and reduce the number of long distance hops, as depicted in Figure 2.

2.2 End System Multicast Management Substrate

The ESM management substrate is the higher layer of PeerCast middleware, responsible for ESM event handling, multicast group membership management, multicast payload delivery, and cache management. It is built on top of the PeerCast P2P network management substrate and uses its APIs to carry out ESM management functions. It consists of three protocols. The *Multicast Group Membership Management* protocol handles all multicast group creation and subscription requests, and adds new subscribers into the multicast tree. The *Multicast Information Dissemination* protocol is responsible for disseminating multicast payloads through unicast links among end-system nodes. When some peers depart or fail in the ESM overlay, end-system nodes use *Multicast Overlay Maintenance* protocol to re-assign the interrupted multicast services to the other peers, while maintaining the same objectives – exploiting network proximity and balance the load on peers.

In PeerCast every peer participates in ESM service, and any peer can create a new multicast service of its own interest or subscribe to an existing one. There is no scheduling node in the system. No peers have any global knowledge about other peers. PeerCast organizes multicast service subscriber into multicast trees following the ESM management protocols, taking into account factors like peer resource diversity, load balance among peers, and overall system utilization. In PeerCast, the establishment of an ESM multicast service involves the following steps.

Creating Multicast Group and Rendezvous Node. An ESM service provider first defines the semantic information of its service and publishes a summary on an off-band channel. Potential subscribers could locate such information using the off-band channel. Each multicast group in PeerCast is uniquely identified by a m-bit identifier, denoted as g. Using the PeerCast P2P protocol, g is mapped to a peer with an identifier that is numerically closest to g. An indirect service is then setup on this end-system node.

We refer to this end-system node as the *rendezvous node* of the ESM service. The rendezvous node re-directs subscribers to the service provider (the ESM source), who will actually inject the ESM payload into the multicast group.

Managing Subscription. Peers that subscribe to an ESM service will form a group, which we refer as the *multicast group*. Subscribers check those established multicast groups using the off-band channels, and identify the services that they want to subscribe. Through the rendezvous node, they learn the identifier of the ESM source. An end-system node joins a multicast group by starting the subscription process at one of its virtual nodes closest to the multicast source. The subscription request is handled in a way similar to the lookup request in PeerCast. It will be forwarded until it reaches the multicast source or a peer that is already in the multicast group. The reverse path will be used to carry multicast payload and other signal messages for multicast tree maintenance.

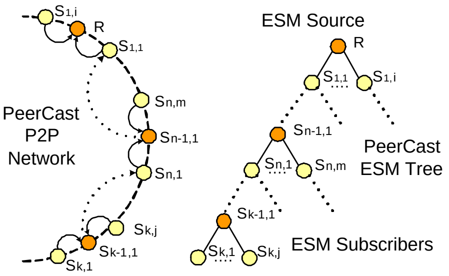

Figure 3. Improve PeerCast with Landmark Signature and Neighbor Lookup schemes

Efficient Dissemination using Multicast Groups. One unique feature of PeerCast is the *Neighbor Lookup* technique. Using this technique, each peer initiating or forwarding a subscription request will first check and try to subscribe to its P2P network neighbors before sending or forwarding the request. Our landmark signature clustering scheme ensures that a peer can reside close to its physical network neighbors in P2P network with high probability. The neighbor lookup scheme thus let the new subscriber directly subscribe to its physical network neighbor, if it is already in the multicast group. PeerCast system can then take advantage of this property and optimize the multicast tree in various ways. Figure 3 gives an example of how the neighbor lookup scheme works. Peer $\mathsf{S}_{\mathsf{k},1}$ first check if its P2P network neighbors have already joined the multicast group, before it forwards its subscription request to the next hop peer. It finds that peer $\mathsf{S}_{\mathsf{k}-1,1}$ is already in the multicast tree. It then directly subscribes to peer $\mathsf{S}_{\mathsf{k}-1,1}$ and terminate the subscription. Similarly, peer $\mathsf{S}_{\mathsf{k},\mathsf{j}}$ subscribes to $\mathsf{S}_{\mathsf{k}-1,1}$, and both $\mathsf{S}_{\mathsf{n},1}$ and $\mathsf{S}_{\mathsf{n},\mathsf{m}}$ are connected to their physical network neighbor $\mathsf{S}_{\mathsf{n}-1,1}$.

Due to the space restriction, we omit the other details about the ESM overlay maintenance and optimization. Readers who are interested may refer to our technical report [20].

3 Reliability in PeerCast

3.1 Reliability Considerations

End-system multicast services are built upon the overlay network composed of widely-distributed and loosely coupled end-systems. The network infrastructure and end-systems are subject to service interruptions caused by perturbations like unintentional faults or malicious attacks. A reliable ESM system should deal with both kind of perturbations. We discussed the security issues of ESM systems in [20]. In this paper, we focus our research efforts on designing reliable ESM system against non-malicious failures.

Failure Resiliency in PeerCast. Failure resiliency is the system's capability to tolerant unintentional faults and non-malicious failures. Maintaining uninterrupted multicast service in highly dynamic environments like P2P network is critical to the reliability of an ESM system. To design such a system, we considered the following situations that may cause the interruption to ESM services.

- When a peer p departs the network abnormally, say the user terminates the application before p handoff all its workload, the ESM services to peers downstream to p will be interrupted.
- When a peer p fails unexpectedly, it will stop provide ESM services to its downstream peers. And thus they will experience service interruption.
- Even when a peer departs the system normally, if the handoff takes longer time than the ESM overlay needs to recover the multicast service, the service of the leaving peer's downstream peers will be interrupted.
- When a peer p fails, the service replica can not be activated soon enough such that the service of its downstream peers will be interrupted.

We did not consider handling the failure of multicast sources because in the case such as video conference or online live broadcast, the failure of multicast sources can hardly be compensated. We assume that the ESM source is reliable and focus our efforts on building the reliable ESM overlay network.

Departures and Failures. We identify two types of events that depart a peer 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. The leaving peer notifies the other peers to actively take over the multicast services that it was handling. 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 a lost multicast service promptly with less overhead, a replication mechanism is needed. In both cases, multicast service can be restored by letting the peers whose services are interrupted to re-subscribe. This is the approach adopted by [4], and is also implemented in PeerCast as the "plan B" for service recovery.

Notice that once there is a replication mechanism, which enables the continuation of the multicast service from the service replica, the proper departures are very similar to failures in terms of the action that needs to be taken. This will eliminate the explicit re-subscriptions during peer 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.

3.2 Service Replication Scheme

The failure of an end-system node will interrupt the ESM services it receives from its parents and forwards to its children. To recover the interrupted multicast service without explicit re-subscribing, each end-system node in PeerCast replicates the multicast service information among a selection of neighbors. The replication scheme is dynamic. As peers join and depart the ESM overlay, replicas are migrated such that there are always a certain number of updated replica exist. This property 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. Replicas of the ESM group information are installed on a selection of peers. After replicas are in place, the second phase keeps those replicas in consistency as end-system nodes join or leave the ESM group. We denote this phase as the *replica management* phase.

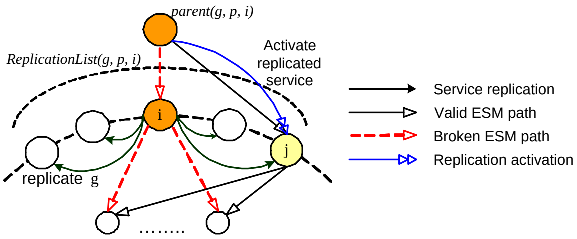

Figure 4. Multicast Service Replication with $r_f = 4$

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 $ReplicationList(g, p, i)$. We refer this set as the *replication list* of group g on peer (p, i). The size of the replication list is r_f, which is referred as the *replication factor* and is a tunable system parameter. To localize the operations on the replication list, we demand that $r_f \leq 2 * r$, which means that all the replica holders in $ReplicationList(g, p, i)$ are chosen from the neighbor list $NeighborList(p, i)$ of peer (p, i).

For each ESM group g that a peer p is actively participating, peer p will forward the replication list $ReplicationList(g, p, i)$ to its parent peer $parent(g, p, i)$ in group g. Once p departs from group g, its parent peer $parent(g, p, i)$ will use $ReplicationList(g, p, i)$ to identify another peer q with identifier j to take over the ESM multicast forwarding works of p. q will use the group information that p installed 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 $ReplicationList(g, q, j)$, and use it to replace $ReplicationList(g, p, i)$ on $parent(g, p, i)$, which is equivalent to $parent(g, q, j)$ 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.

3.3 Replica Management

In this section, we explain how the described dynamic replication scheme is maintained as end-system nodes join or depart from the ESM system. Since the active replication scheme works for both peer departure and failure cases, 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 p with identifier i, it is replicated to the peers in the $ReplicationList(g, p, i)$. 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 $ReplicationList(g, p, i)$. 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 $ReplicationList(g, p, i)$.
- For each group g that p is forwarding ESM payload, p removes the departing peer p' from the replication list $ReplicationList(g, p, i)$.
- 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 $ReplicationList(g, p, i)$ to its parent peer $parent(g, p, i)$ 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 $ReplicationList(g, p, i)$.
- 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 $ReplicationList(g, p, i)$.
- 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 $ReplicationList(g, p, i)$ to its parent peer $parent(g, p, i)$ in multicast group g.

Updating Replicas. As end-systems subscribe or unsubscribed from ESM groups, their subscription or unsubscription requests will be propagated up in 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 $ReplicationList(g, p, i)$.

Replica Management Overhead. Assuming in average a peer participates k multicast groups, we can summarize the replica management overhead as:

- Average storage cost for the replicas stored per peer $\sim k * r_f$
- Average update cost for replicas stored per peer $\sim k * r_f$
- Average number of new replications required for entrance/departure per peer $\simeq k$.

3.4 Replica Selection Policy

In this section we describe the details of the replica activation policy of PeerCast. We consider two factors, namely peer load factor and replication distance factor, in evaluating the suitableness of a replica holder to be activated. We define each of these factors as follows:

Let p_f denotes a peer that fails, and p_r denotes a replica holder of p_f for multicast group g.

Peer load factor is denoted as $PLF(p_r)$. It is a measure of a peer p_r's willingness to accept one more multicast forwarding workload considering its current load. It is defined as follows:

$$PLF(p_r) = \begin{cases} 1 & \text{if } p_r.load \leq tresh{*}\text{MAX\_LOAD} \\ 1 - \frac{p_r.load}{\text{MAX\_LOAD}} & \text{if } p_r.load > tresh{*}\text{MAX\_LOAD} \end{cases}$$

Replication distance factor is denoted as $RDF(p_f, p_r)$. It is a measure of the network proximity of the peer p_r to the failed peer p_f. RDF is defined as follows:

$$RDF(p_f, p_r) = \frac{1}{ping\_time(p_f.IP, p_r.IP)}$$

Let $UtilityF(p_f, p_r, g)$ denote the utility function, which returns a utility value for activate the service replica of peer p_f and group g on peer p_r. It is calculated based on the two measures given above:

$$UtilityF(p_f, p_r, g) = PLF(p_r) + \alpha * RDF(p_f, p_r))$$

Note that we give more importance to the peer load factor PLF. For instance, the service replica on a peer that is very close to the failed peer will not be activated if the replica holder is heavily loaded. α is used as a constant to adjust the importance of replication distance factor with respect to peer load factor. For a lightly-loaded ESM overlay, we want to have a larger value of α since the probability that peers get overloaded is lower, and a more efficient ESM overlay is more desirable. In a heavily-loaded ESM environment, we may want to have lower value of α, to guarantee the feasibility of the multicast plan first.

3.5 Reliability Analysis

Given our replication scheme, a multicast service will be interrupted on a peer only when all its replica holder fail in a short time interval, not letting the dynamic replica management algorithm to finish its execution. We call this time interval the *recovery time*, denoted by Δt_r. We call the event of having all peers contained in a replication list fail within the interval Δt_r, a *deadly failure*.

Assume a peer p with identifier i departs the ESM overlay at time t_d, we want to know the probability that the ESM service of group g could be properly recovered within Δt_r. In another word, we want to know the probability that the replica holders in $ReplicationList(g, p, i)$ all fail during recovering interval Δt_r, which we denote as $Pr_f(p, i)$.

We assume the life time of each peer before it fails follows independent and identical exponential distribution with parameter λ_f, and the life time of each peer before its proper departure 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 exponential distribution with parameter $\lambda = \lambda_d + \lambda_f$. The mean active time st of a peer in ESM overlay is equal to $1/\lambda$, which we refer as its *service time* in our later analysis. The probability that a peer departs by failure is $\frac{\lambda_f}{\lambda_d+\lambda_f}$, and the probability that a peer departs properly is $\frac{\lambda_d}{\lambda_d+\lambda_f}$.

We use random variables $L_1, L_2, \ldots, L_{r_f}$ to denote the amount of time that replica holders in $ReplicationList(g, p, i)$ 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, $L_2, \ldots, L_{r_f}$ still follow the exponential distribution with parameter λ. We thus have:

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

p owns a set of identifiers $p.ids$ by our virtual node scheme [20]. Assuming there is no overlapping among the replication lists of p's different identifiers, i.e. $\forall_{i,j \in p.ids} Pr_f(p, i) = Pr_f(p, j)$, we can express the probability with which p's departure causes any service interruption as:

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

We use the turnover rate of [16] to approximate λ. As reported in [16], half of the peers in the P2P system will be replaced by new peers within one hour. We have $Pr\{\text{a peer departs in an hour}\} = 0.5$, which indicates $1 - e^{-\lambda \cdot 60} = 0.5$ and the mean service time $st = 1/\lambda = 86.56$ minutes. When we setup our system as $r_f = 4$, $\Delta t_r = 6$ secs, $E[|p.ids|] = 4$, and all peers depart by failure, we have $Pr_f(p) \simeq 7.2258 * 10^{-12}$.

4 Experimental Results

We have designed a simulator that implements the mechanisms explained in this paper. In the following subsections, we investigate two main subjects using results obtained from experiments carried out on our simulator. We first study the effects of our replication scheme on recovering multicast service under various node failure scenarios. Next, we evaluate how the efficiency of ESM overlays could be improved using the network proximity information and the neighbor lookup scheme.

We used the GT-ITM package [19] to generate a set of network topologies for our simulation. 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, leading from the multicast source to each subscriber, into a shortest path tree.

4.1 Reliability

The Role of Network Proximity. Most of the replication management messages are exchanged among peers within each other's neighbor list. If the multicast service carrier peers and their replica holders are physical network neighbors, peers can update replica, detect failures, and restore services faster, while incurring less communication overhead.

This section examines the precision that landmark signature technique can achieve in clustering end-system nodes by their network proximity. The metrics we use is the percentage of peers that have physical network neighbors in their local P2P neighbor lists.

We simulate the P2P networks with $1 * 10^4$ to $9 * 10^4$ peers, and set the neighbor list size parameter r to be 4, 8, 12, 16, and choose 1 as the splice offset value. We use the experimental result to guide our designation and implementation of PeerCast.

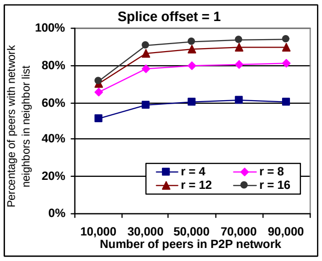

Figure 5. Network proximity clustering precision

Figure 5 shows the results of our simulation. We observe three facts. First, larger value of r increases the chances that peers can find physical network neighbors in the local neighbor list. Second, the landmark signature scheme can capture the network proximity information with satisfying precision. As many as 94% of all the peers possess one or more network neighbors in their local neighbor list, when r is set to 16. Third, larger peer population can increase the precision of clustering, as more peers are from the same network sub-domains.

Failure Resilience. 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 much lower 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 $2 * 10^4$ peers 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 deadly failure of peers will trigger the re-subscription process and cause the service interruption to its downstream peers. 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 with replica activation, instead of the expensive re-subscription.

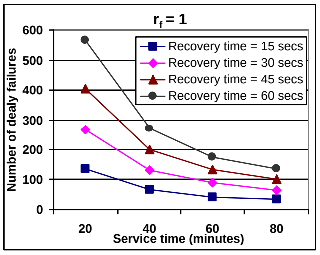

Figure 6. Deadly failure, r_f = 1

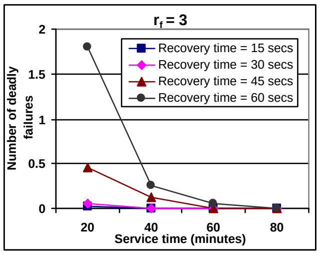

Figure 7. Deadly Failure, r_f = 3

The graphs in Figures 6 and Figure 7 plot the total number of deadly failures 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 deadly 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.

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

Service Recovering Overhead. We measure the overhead of service recovering by the total number of messages exchanged to restore the interrupted service. A deadly failure of a peer causes its downstream peers to re-subscribe to the interrupted multicast services. On the other hand, if a peer's service replica is activated when it fails, only one message is used to report the failure and one fast activation message is involved to activate the service replica.

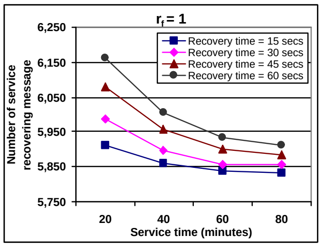

Figure 8. Number of service recovering messages under replication scheme

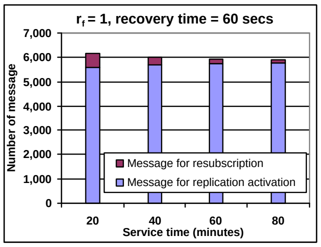

Figure 9. Number of service recovering messages under replication scheme

We observe the number of messages exchanged under the same experiment configurations of Figure 6. We count the total number of messages generated for both replica activation and service re-subscription. The results in Figure 8 conforms to the curves of Figure 6. When the number of deadly failure increases, more messages are

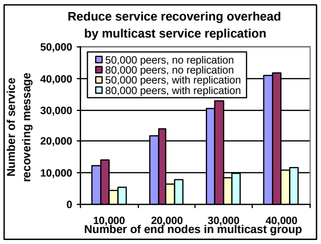

Figure 10. Number of service recovering messages, $\Delta t_r = 15$secs, $st = 20$ minutes, replication scheme has r_f = 1

generated for the re-subscription requests. However, as plotted in Figure 9, most of the messages are for the replica activation, since most of the interrupted services are restored by the replica activation.

To evaluate the effect of the replication scheme on reducing the service recovery overhead, we compared the number of messages incurred by the replication scheme with the number of messages involved when there is 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. The experiment results are plotted in Figure 10. The overhead of service recovering increases almost linearly, as the number of peers in the multicast group and the P2P network increases. However, we observe that when the service replication scheme is used, much fewer messages are generated. With the overhead of maintaining ONE service replica, we reduce the messaging overhead by 62.3% to 73.8%.

4.2 End System Multicast Efficiency

We want to to study the efficiency of PeerCast and the effects of the landmark signature technique and the neighbor lookup technique. In this section we compare two flavors of PeerCast overlays. The basic PeerCast system does not implement the aforementioned techniques, while the enhanced PeerCast is equipped with all of them. We notice that our basic PeerCast scheme is very similar to Scribe [4], and has similar performance as well.

We simulate a set of P2P networks with fixed number of peers as $5 * 10^4$, which model P2P networks shared by multiple ESM services. 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. The splice offset is set to 1, a value that allow us to maintain the randomness of identifier distribution and exploit the network proximity of end-system nodes.

Delay Penalty. 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 message replication and unicast forwarding. We use *relative delay penalty* to evaluate this penalty. It is defined as ratio of average PeerCast delay and the average delay using IP multicast.

The landmark signature technique and our ESM management protocol put the multicast root's network neighbors close to it in the multicast tree. And the neighbor lookup scheme reduce the last hop delay on the other end of the multicast forwarding path. The result is the multicast paths envisioned in Figure 2. As plotted in Figure 11, the landmark signature technique and the neighbor lookup scheme together can reduce the relative delay penalty by about 15%.

Node Stress. End-system nodes in PeerCast handles jobs like the multicast group maintenance, and the multicast message replicating and forwarding. 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 each non-leave end-system node handles.

Using the neighbor lookup technique, a peer first trying to leverage its physical network neighbors before subscribing to remote peers. Because our landmark signature technique gives peers better chance to find network neighbors in their P2P neighbor list, the multicasting workloads are thus handled by more peers. As presented in Figure 12, The enhanced PeerCast ESM overlays have much lower node stress compared to the basic PeerCast scheme. As the number of peers in the multicast group increases, a peer's chance to subscribe to its network neighbor increases too. The result is the decreasing node stresses against increasing peer number in the multicast group. On the contrary, the basic PeerCast overlays have to follow the prefix matching to build the multicast tree, and the ESM workloads are distributed only on peers sharing prefixes with the multicast source. Hence, compared to the enhanced PeerCast overlays, a smaller portion of peers have to handle the same amount of ESM workload. This explains the increasing node stress of the basic PeerCast overlays when we increase the number of peers in the multicast group.

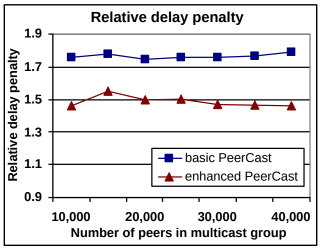

Figure 11. Relative delay penalty

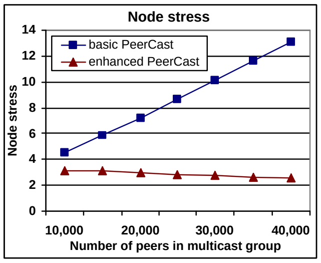

Figure 12. Node stress

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

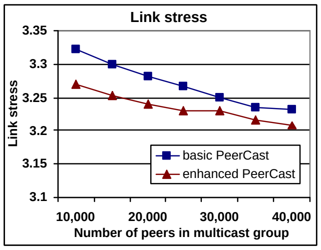

Figure 13. Link stress

The landmark signature technique renders high precision of clustering peers by their network proximity. The neighbor lookup scheme thus can take advantage of this property and put more forwarding hops within local networks and reduce the number of multicast forwarding messages traveling through the inter-network links. Together, these two techniques reduce the link stress since shorter forwarding path usually incurs fewer IP packets.

As the size of multicast group grows, more peers from the same local network domain participate the multicast group. Because the link stress over the last-hop end-systems' access links has almost constant value, the increasing number of the last-hop access links offsets the increasing link stress over the core network. The result is the decreasing link stresses as shown in Figure 13.

5 Related Work

EMS protocols like [5, 6, 11] are developed primarily for relatively small networks. A few nodes are responsible for the management functionalities such as gathering and analyzing network information [3] and maintain the multicast overlay structure [6]. The NICE [2] protocol builds a multicast overlay into a hierarchical control topology. A top-down approach is used to forward joining requests recursively from leader nodes into the most suitable lower layer clusters. The Overcast protocol [11] creates a distribution tree rooted at the multicast source. It uses end-to-end measurement results to optimize multicast path between the root and the multicast group members. The Narada [6] system generates a mesh network containing all the subscribers and uses a centralized algorithm to calculate and maintain the multicast tree.

Recent studies in Peer-to-Peer (P2P) network [13, 15, 18, 21] present a new orient to address the issues of managing large-scale multicast overlays. In Bayeux [22] system, the joining request is forwarded to the root node first, from where a message is routed reversely to the new member. The nodes on the routing path form the multicast forwarding path for the new member. The multicast system described in [14] 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 PeerCast basic protocol is highly inspired by Scribe [4]. Scribe builds multicast trees using the locality properties of Pastry [15]. Pastry has the problem known as "logarithmically deterioration of routing", and its locality properties is based on the triangle inequality which may not hold in Internet topology.

The approach taken by PeerCast differs from these existing researches in two aspects. First, PeerCast system offers reliability guarantee through replication. This ensures the availability of multicast service in high dynamical environments like peer-to-peer network. Second, PeerCast presents scalable solution for end-system multicast with a heterogeneous overlay network. We use the landmark signature and neighbor lookup scheme to build efficient multicast overlay in a decentralized way.

There exist two different classes of well-known replication techniques in the distributed systems literature, i.e. active replication and passive replication. In active replication [12, 17], each request is processed by all replicas. In passive replication (i.e. primary-backup) [1, 9], one replica processes the request, and sends updates to the other replicas. Active replication ensures a fast reaction to failures, whereas passive reaction usually has a slower reaction to failures. On the other hand, active replication uses more resources than passive replication. The latter property of the passive replication is the main motivation in selecting a variation of primary-backup approach in PeerCast.

Traditional research on replication in the area of Internet information services [10, 7] focus on minimizing the response time by directing client requests to best available server replicas by considering load on the servers and their proximity to clients. The replica selection problem in PeerCast is somewhat related to the server selection problem in Internet information services domain. Although the setup is quite different, PeerCast activates service replica by considering some similar metrics but for the purpose of better system utilization and load balance.

6 Conclusion

We have presented a dynamic passive replication scheme for PeerCast, an efficient and self-configurable peer-to-peer system for application-level multicast, 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 multicast services over large-scale network of end-system nodes, with reasonable efficiency.

References

[1] P. Alsberg and J. Day. A principle for resilient sharing of distributed resources. In *Proceeding of ICSE*, 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] Z. Fei, S. Bhattacharjee, E. W. Zegura, and M. H. Ammar. A novel server selection technique for improving the response time of a replicated service. In *Proc. of IEEE INFOCOM*, 1998.

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

[9] R. Guerraoui and A. Schiper. Software-based replication for fault tolerance. *IEEE Computer*, 30(4):68–74, 1997.

[10] J. D. Guyton and M. F. Schwartz. Locating nearby copies of replicated internet servers. In *Proc. of ACM SIGCOMM*, 1995.

[11] 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.

[12] L. Lamport. The implementation of reliable distributed multiprocess systems. *Computer Networks*, (2):95–114, 1978.

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

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

[15] 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.

[16] 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.

[17] F. Schneider. *Replication Management Using the State-machine Approach*. Addison-Wesley, second edition, 1993.

[18] 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.

[19] 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.

[20] J. Zhang, L. Liu, C. Pu, and M. Ammar. Reliable end system multicasting with a heterogeneous overlay network. Technical Report GIT-CERCS-04-19, CERCS, Georgia Institute of Technology, April 2004.

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

[22] 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.