Improving Stability for Peer-to-Peer Multicast Overlays by Active Measurements

Ye Tian [*], Di Wu, Kam-Wing Ng

Department of Computer Science and Engineering
The Chinese University of Hong Kong, Shatin, N.T., Hong Kong

Abstract

The instability of the tree-like multicast overlay caused by nodes' abrupt departures is considered as one of the major problems for application level multicast systems. In this paper, we present a protocol for improving the overlay's stability by actively estimating the nodes' lifetime model, and combining the nodes' lifetime information with the overlay's structural properties. We use the shifted Pareto distribution to model the nodes' lifetimes in designing our protocol. To support this model, we have measured the residual lifetimes of the nodes in a popular IPTV system named PPLive [21], and have formally analyzed the relationships between the distribution of the nodes' lifetimes, ages and their residual lifetimes under the shifted Pareto distribution model. We evaluate the overlay construction strategies, which are essential in improving the overlay's stability in our protocol, by comparing them with a number of other strategies in simulation. The experimental results indicate that our proposed protocol could improve the overlay's stability considerably, with informative but not necessarily accurate lifetime model estimation, and with limited overhead imposed on the network as well as negligible sacrifice regarding the end-to-end service latencies for the nodes on the overlay.

Key words: Peer-to-peer network, multicast overlay, stability

1 Introduction

With the increasing deployment of broadband techniques, media multicast is becoming more and more popular in today's Internet. However, due to the insufficient support from

[*] Corresponding author. Tel.: +852 3163 4253.
Email addresses: ytian@cse.cuhk.edu.hk (Ye Tian), dwu@cse.cuhk.edu.hk (Di Wu), kwng@cse.cuhk.edu.hk (Kam-Wing Ng).

the infrastructure, IP-multicast techniques can not be widely deployed. Researchers thus propose application-level overlay networks for media multicast, in which unicast channels are built among ordinary end-systems to transfer streaming data and users distribute the media from the source node in a cooperative fashion. Currently there are two categories of overlays in the context of application-level media multicast: the tree-like overlay and the mesh-like overlay.

The tree-like overlay is initially proposed as an imitation of the router-level multicast topology, in which each node connects to a parent node and receives the media streaming. For media multicast, the content needs only to be pushed from the parent node towards the children nodes, which is very simple to implement. Representative systems of the tree-like multicast overlays could be found in [8], [9], [10], [11] and [12], and multiple tree overlays could be found in [13] and [14]. The principle limitation of the tree-like overlay focuses on its instability: since it is the ordinary end-systems that compose the overlay network, one node's abrupt departure or abnormality will reduce the quality of service for all the nodes that receive the streaming data from it directly or indirectly. On the other hand, the mesh-like overlay is recently proposed in the context of content distribution and media multicast to cope with the dynamical network environments. The idea is that a node chooses multiple peering nodes for data exchanging, and consequently the nodes form a self-organized mesh-like overlay which is optimized dynamically. Although the mesh-like protocol is more resilient under abrupt failures, it has some limitations. First, it is complex. The nodes are required to be intelligent in choosing their peering partners. Second, unlike the tree-like overlays, in which data is pushed from parent nodes to their children nodes, in the mesh-like overlays, nodes pull data from their peering partners. Thus, it is required that a node must exchange its content information with its peering nodes frequently, which impose a burden on the system, especially in time-critical applications such as the live media multicast. Representative mesh-like systems include [15], [16], [17], [18], and [19].

In this paper, we aim to improve the stability of the tree-like overlays, which is considered as the major problem for such an overlay. Generally, for a tree-like multicast system, it must handle the following two situations:

- Nodes joining: When a new node joins the overlay, usually it will contact a well known bootstrap node for some available nodes on the overlay, and will choose one among them as its parent node for streaming. This bootstrap node could be the source node or some other management nodes publicly declared with their addresses or URLs.
- Nodes departing: When a node leaves, all of its children nodes will lose their connections and several disconnected sub-trees will be formed, each is rooted at a disconnected children node. Basically, all the nodes in a sub-tree will have a discontinuity of the streaming service until the disconnected sub-tree is reconnected on the overlay. When a node becomes disconnected due to parent node departure, it will look for a new parent node to rejoin the overlay. Disconnected nodes may connect to leaf nodes, or they may replace some other nodes on the overlay by preempting them.

Obviously, to handle the above situations, a parent selection mechanism is required for the new node or the disconnected node to select a proper parent node, and the disconnected node may need a preemption mechanism to preempt a proper node and replace it.

In this paper, we propose an overlay construction protocol to improve the overlay's stability, especially, we extend the RanSub protocol [24] for active estimation of the nodes' lifetime model, and present one parent selection strategy and two preemption strategies, which are essential for improving the overlay's stability in our protocol. In our "prediction" parent selection strategy, a new node estimates the probability of an ancestor failure for all the potential parent nodes, and connects to the one with the best path reliability. For the node preemption, we present two strategies: "preemption by descendant" and "preemption by service". In both strategies, a disconnected node will rejoin the overlay at a certain position according to its ability in providing the multicast service for the overlay, and the two strategies differ in their estimations of the services provided by candidate nodes. In designing our protocol, the shifted Pareto lifetime model is used to predict the reliability of the nodes. To verify the applicability of this model, we have measured the nodes' residual lifetimes in a popular P2P based IPTV multicast system named PPLive [21], and show that the shifted Pareto distribution is accurate in modeling the nodes' lifetimes. We further show theoretically that the age information of the nodes could be exploited efficiently to derive the lifetime model of the nodes in a P2P multicast system. Finally, to evaluate our solution, an event driven simulator is developed and we evaluate a number of strategy combinations using the purely synthetic traces and the half synthetic traces derived from the PPLive measurement. The simulation results indicate that the strategy combination of "prediction + preemption by service" achieves the best performance regarding the overlay stability; the combination of "prediction + preemption by descendant" also improves the stability of the overlay considerably, and both are obviously better than other strategy combinations, such as the "minimum-depth + preemption by degree" combination preferred in [20]. Moreover, we show that our protocol only relies on informative but not necessarily accurate lifetime model estimations, and could work inexpensively with limited sized random subsets of nodes and with very few preemption operations. We also study the end-to-end service delay for the nodes on the overlay formed by running different strategy combinations and demonstrate that our strategies preserve the good performance regarding the multicast service latency.

1.1 Related Work

To improve the overlay stability, a possible solution is to consider the individual node's reliability, as it is widely reported that the nodes' lifetimes in P2P applications exhibit a heavy-tailed property, which means the older nodes are more reliable than the younger ones. In other P2P systems (e.g. DHTs, Gnutella), it is reported that exploiting the nodes' age information helps to build more efficient and robust systems [26]; and in [27], it is also reported that by estimating the nodes' lifetimes, the P2P system could be made more robust under the failures of some important nodes. However, previous studies [4]

[20] also show that when constructing a multicast overlay, the nodes' overlay structural properties such as the depth or the out-degree should be considered over the nodes' age information: in [4], the authors point out that the parent selection strategy of choosing a node with the minimum-depth outperforms the strategy of choosing a node with the longest age regarding the overlay stability; while the authors of [20] report that for the preemption strategy, considering the nodes' outgoing bandwidths (a higher bandwidth node replacing a lower bandwidth one) leads to better overlay stability than considering the nodes' ages (an older node replacing a younger one). Our work differs from these previous works in that in our approach, the nodes' lifetime or age information as well as the structural properties are exploited to build the multicast overlay, which leads to a better performance compared with solutions only considering one factor.

Another work which proposes to improve the tree-like multicast overlay stability by combining the age information with the overlay structural properties could be found in [29] [30]. In their solution, the overlay is proactively adjusted periodically. However, as pointed out by [33], such a proactive adjusting mechanism might be exploited by malicious users to launch a denial of service attack, where the overlay is triggered to make frequent unnecessary adjustments based on wrong judgements. Unlike the proactive approach, our overlay adjustment operations are purely passive and are only executed on the events of node joining and failure, thus making the multicast overlay more adaptive to the dynamics of the nodes, and more robust under the denial of service attack.

The remainder of this paper is organized as follows: in Section 2, we present our measurement results on the PPLive multicast system; in Section 3, the shifted Pareto lifetime model, which serves as the theoretical foundation of our solution, is discussed, and we propose our overlay construction protocol; we describe the simulation methodology, the strategies for comparison, as well as the metrics in performance evaluation in Section 4; the experimental results are presented and discussed in Section 5; finally we conclude this paper in Section 6.

2 PPLive Measurement

In this section, we describe our measurement study for the PPLive system [21], which is a popular P2P based IPTV application with over $100,000$ simultaneous users. We are concerned with two properties of the nodes in the system, their lifetimes and their outgoing bandwidths. We choose PPLive as the target system in our measurement study for two reasons: first, PPLive is a very successful P2P based IPTV system with many users; and the other reason is that there are very few measurement studies on this system. Actually to our knowledge, there is only one work [31] which concentrates on the traffic analysis on PPLive. In this section, we first introduce the PPLive system and our measuring methodology, and then we present the measurement results.

2.1 Methodology

PPLive is a P2P based multicast application which is very successful and widely used by users from all over the world. According to very limited information released on its website, PPLive forms two kinds of application-level overlays during the session: one overlay is formed for the node membership and channel information management, and the other is formed for media streaming. PPLive uses a two-layered supernode architecture in which a node could be an ordinary node or a super node (called searching node in PPLive) to construct the management overlay; while for each media channel, a mesh-like overlay is formed for streaming the media content. In our study, we are interested in two properties of the nodes on the PPLive streaming overlay: 1) their lifetimes; and 2) their outgoing bandwidths. The first is highly related to the stability of the overlay and the second is important for the overlay's service capacity.

To obtain the nodes' lifetimes, we setup a PPLive installation and join in the streaming overlay of a popular TV program. Our simple idea is to actively probe all the nodes which have been encountered by our measuring node until they are offline. A measuring program is developed for this purpose. As a node in PPLive may explore new peering nodes by setting up new connections and abandon connections which are not good enough, we decrease the incoming bandwidth and block the incoming connections of our measuring computer, in this way we force the PPLive installation to explore more potential peering nodes for better streaming service, and speed up our measurement job. We must point out that actually we are measuring a node's residual lifetime, which is defined as the period between the moment it is encountered by our PPLive installation to the moment it quits the PPLive streaming overlay. We will have a theoretical discussion on the relationship between the distribution of the nodes' lifetimes and their residual lifetimes in Section 3. Another issue is that in PPLive, a mesh-like overlay is formed for data streaming, which is different from the tree-like overlays we aim to improve. However, in Section 3.1 we will show that the characteristics of nodes' lifetimes are universal in P2P networks, and are almost independent of the overlays' topologies.

We have also estimated the nodes' outgoing bandwidths. The method used here is similar to one of the methods described in [4]. We lookup a node's IP address in the Regional Internet Registries at [5], [6] or [7], and estimate the access technique the user with the address might use. We assume all the addresses in the education domains are Ethernet accessed, and for the addresses of the commercial ISPs, we visit the ISP's website and assume that the user uses the most advanced access technique provided by the ISP. Obviously our estimation is an optimistic one as we always assume the best possibility. However, we believe it reflects the real-world situation as long as we could differentiate the addresses in education networks from the addresses of home users with commercial ISPs, since the former is usually well connected with Ethernet, while the home users usually use DSL or Cable modem techniques with limited outgoing bandwidths. Fortunately, all the addresses in education networks are well recorded by the Regional Internet Registries.

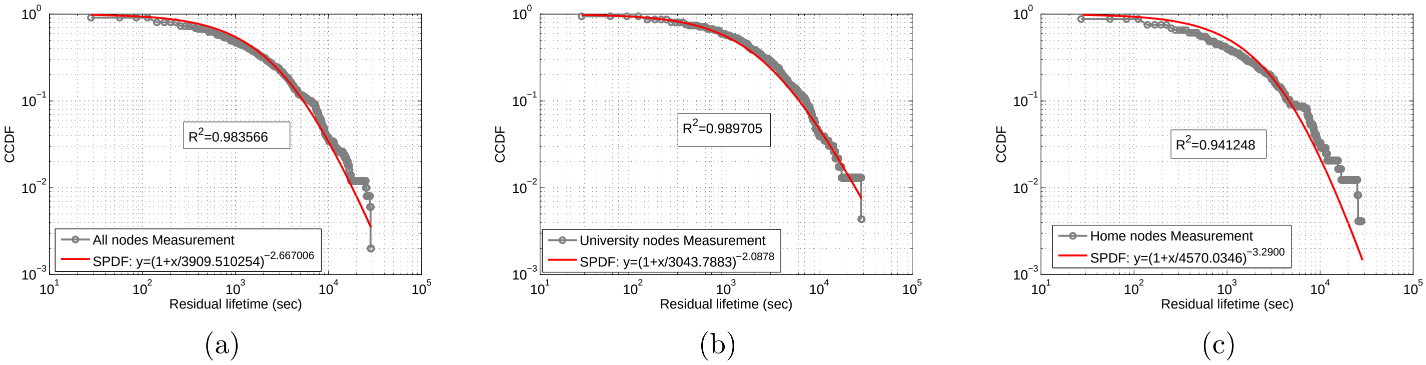


Fig. 1. CCDF of the residual lifetimes for (a) all the nodes, (b) university nodes, and (c) home nodes

Table 1
Outgoing bandwidth estimation

Access technique & Outgoing bandwidth	Num. of nodes(%)
EDU(Ethernet): Above 1 Mbps	230 (46%)
DSL, Wireless: 100Kbps-500Kbps	191 (38.2%)
Cable Modems: 600Kbps-1Mbps	52 (10.4%)
Unknown	27 (5.4%)

2.2 Results

We have traced 500 nodes for their residual lifetimes, with the longest record of $28,744$ seconds (nearly 8 hours), and the average of $2,345$ seconds (nearly 40 minutes). We also find that 45 nodes are found offline immediately after they are encountered by our PPLive installation, which means their residual lifetimes are shorter than 2 minutes, which is our probing interval.

We plot the distribution of the residual lifetimes we have measured in Fig. 1(a). The x-axis is the length of the residual lifetime in seconds, and the y-axis is the Complementary Cumulative Distribution function (CCDF). Note that we use logarithmic scale in both axes. We plot all the 500 records on the figure, and by using the method which will be introduced in Section 3 with all the 500 records, we derive the shifted Pareto distribution function (SPDF) [22] which is close to the statistical result. We could see that the curve of the shifted Pareto distribution function matches the statistical trace very well, with a goodness-of-fit (R^2) as 0.983583. We also estimate the function of Exponential distribution, and find that it does not match our measurement results well, with a goodness-of-fit as $R^2 = -7.756703$. We do not plot it on the figure for briefness. Our measurement result indicates that the residual lifetimes of nodes on the PPLive overlay are heavy-tailed, as reported in other works such as [1], [2], [3] and [4]. Furthermore, it is observed that the distribution of the nodes' residual lifetimes could be modeled by the shifted Pareto distribution very well.

We roughly estimate the outgoing bandwidths for the nodes and report the results in Table 1. Basically we could divide the nodes recorded into two categories, the nodes located in university domains with Ethernet connectivity and the nodes with commercial ISP addresses accessing the Internet using DSL or Cable modem techniques. We call the first as university nodes and the second as home nodes. Among all the nodes we have observed in our measurement, there are 46% university nodes and 48.6% home nodes, which shows that PPLive users are heterogeneous regarding the bandwidth. One natural question is whether or not users with different access techniques behave differently regarding their residual lifetimes. We plot the CCDF of the residual lifetimes for the university nodes and the home nodes in Fig. 1 (b) and (c) respectively, and we also derive and plot the shifted Pareto distribution functions for the two statistical traces separately on the figures. We find that the two traces are almost identical, and both could be matched by the shifted Pareto distribution model very well. Interestingly, we find that the average residual lifetime for the home nodes, which is $2,798$ seconds, is longer than the one of the university nodes, which is $1,995$ seconds, although the latter has better bandwidth connectivity. The reason may lie in the fact that users behind the home nodes usually have more leisure time than users behind the university nodes. More importantly, our finding suggests that the length of a node's lifetime is more related to the users themselves rather than technical factors such as the bandwidth as long as they could obtain satisfactory services.

3 Protocol Design

3.1 Node Lifetime Model

In our measurement study in Section 2, we find that the nodes' residual lifetimes are heavy-tailed, and the CDF could be modeled as a shifted Pareto distribution. We argue that this property is also valid for the nodes' lifetimes, regardless of their bandwidth conditions and the overlay topology, based on the following observations: first, our measurement study shows that both the university nodes and the home nodes exhibit almost identical characteristics in their residual lifetimes, indicating that lifetime is a property independent of the nodes' bandwidths; second, previous studies show that the lifetimes of nodes in different P2P applications are heavy-tailed [2] [4] and could be modeled with a Pareto distribution [1] [3] [22]. These applications include Napster, Gnutella and Akamai multicast. In other words, Pareto lifetime is a universal property in P2P networks. Before the description of our protocol design for improving the stability of the tree-like multicast overlay, we first discuss some features of the Pareto lifetime model which serves as the theoretical foundation for our protocol.

To allow for very short lifetime, we use the shifted Pareto distribution to present the

CDF of the nodes' lifetimes in a P2P overlay. In this model, for a particular node n, if its lifetime is denoted as l, then the probability that l is shorter than x is expressed as $F(x) = \Pr[l < x] = 1 - (1 + x/\beta)^{-\alpha}$. Suppose we begin to monitor all the online nodes on an overlay from a particular random moment, and trace all the nodes until they depart the system, then the period between the moment we start to monitor to the moment a node departs is called its residual lifetime. If the system has evolved for a long time and reaches a stable state, the distribution of the nodes' residual lifetimes is independent of the moment we start to monitor. Let r be used to denote the residual lifetime of node n, we have the following result.

Theorem 1 *In a stable P2P overlay, if the CDF of the nodes' lifetimes follows the shifted Pareto distribution as* $F(x) = 1 - (1 + x/\beta)^{-\alpha}$, $\alpha > 2$, *then the CDF of their residual lifetimes is given by:*

$$F_R(x) = \Pr[r < x] = 1 - (1 + x/\beta)^{-(a-1)}$$

PROOF. This result is proved in [22].□

This result reveals the formal relationship between the nodes' lifetimes and their residual lifetimes: for a Pareto lifetime model, the residual lifetime also follows the Pareto distribution, but appears more heavy-tailed. This result also conforms to the well-known knowledge that heavy-tailed distributions exhibit "memory", which means that users who have stayed in the system for some time are more likely to remain stayed for longer sessions than the newly arrived users.

Although it is well known that nodes' lifetimes in P2P overlays follow the Pareto distribution in CDF, it is difficult to estimate the lifetime model efficiently. Usually people measure a node's lifetime by probing it periodically, as we have done in our PPLive measurement. However, probing a large amount of nodes during their whole lifetimes is a tedious job. If there are some failure reporting mechanisms deployed, in which a node's departure is reported by other nodes such as the neighbors, then probing is not required for the lifetime measurement. However, even with this mechanism, it is still a time-consuming work as the measuring program must wait for all the nodes being traced to depart, and it takes a very long time when the lifetime is heavy-tailed, with a few nodes being online quite long. If we wish to exploit the property of the Pareto lifetime in the protocol design, an efficient and inexpensive lifetime model estimation method is necessary. Unfortunately, even with the failure reporting mechanism, directly measuring the whole lifetime or the residual lifetime is still impractical. On the other hand, since it is easy to know a node's age (the period between its joining the overlay to the moment it is queried for age), deriving the lifetime model from the distribution of the current ages of the nodes is desirable. Suppose at a particular moment, for node n, its age is denoted as a, then we have the following result regarding the distribution of the nodes' ages.

Theorem 2 *In a stable P2P overlay, if the CDF of the nodes' lifetimes follows the shifted*

Pareto distribution as $F(x) = 1 - (1 + x/\beta)^{-\alpha}$, $\alpha > 2$, then the CDF of their ages is given by:

$$F_A(x) = \Pr[a < x] = 1 - (1 + x/\beta)^{-(\alpha-1)}$$

PROOF. According to [32], $F_A(x) = \frac{1}{E[l]} \int_0^x (1 - F(z))dz$, as $F(x) = 1 - (1 + x/\beta)^{-\alpha}$, we have $F_A(x) = 1 - (1 + x/\beta)^{-(a-1)}$, thus the theorem is proved.□

It is very interesting to find that the nodes' ages and their residual lifetimes follow the same distribution, and the important practical meaning behind is that we can simply query the nodes' current ages and derive their lifetime model, which is much more efficient. We propose the following simple heuristics for this purpose.

Suppose the measuring application has obtained the ages from s nodes, denoted as $\{x_1, x_2, \ldots, x_s\}$ in ascending order. Obviously the average age could be calculated as $r_1 = \left(\sum_{i=1}^{s} x_i\right)/s$. Since the curve of Pareto distribution will appear as a straight line in log-log scales, for each pair of records, say x_i and x_j $(x_i \neq x_j)$, the log-log slope rate could be calculated as $\frac{\log i - \log j}{\log x_i - \log x_j}$. The average slope rate from the s records thus could be calculated as $r_2 = \left(\sum_{\forall i,j,x_i \neq x_j} \frac{\log i/j}{\log x_i - \log x_j}\right) / \left(\sum_{\forall i,j,x_i \neq x_j} 1\right)$. If we use $F_A(x) = 1 - (1 + x/\beta)^{-\alpha+1}$ to model the CDF of the nodes' ages, then the average age is $\frac{\beta}{\alpha-2}$, and the slope rate is $(\alpha - 1)$. Solving the two equations,

$$\begin{cases} \frac{\beta}{\alpha-2} = r_1 \\ \alpha - 1 = r_2 \end{cases}$$

we can obtain the nodes' lifetime model with α and β. Note that with this method, the cost is very small: for obtaining r_1 and r_2, the computational cost is $O(s)$ and $O(s^2)$ respectively, which only depends on the size of the sampling set. Another concern is the accuracy of the parameters derived from a small sampling set, as we only use limited information compared with the ages of the whole population. However, we can see from the simulation results in Section 5 that even with inaccurate estimated parameters, our protocol can improve the overlay stability intensively and there is no obvious performance gap between the protocol using less accurate estimation and the one using more accurate estimation, as long as most nodes can make the right decision with the informative but inaccurate parameters.

3.2 The Protocol

Random sampling techniques [23] are important in P2P overlays for locating services and improving performance. In a tree-like multicast overlay, we suppose each node keeps a random subset of the nodes on the overlay for two purposes: 1) when a new node joins, it could contact the bootstrap node for a random subset of nodes on the overlay and choose one eligible (with spare outgoing bandwidth) as its parent; 2) when a node's current parent node departs, this node can connect to one node in its random subset for rejoining the overlay. For serving the second purpose, a desired property of a node's random sampling set is that none of the nodes in it should be the node's descendant. In this work, we assume that the end-to-end bandwidth between two nodes will not change over time, thus a node will not look for a new parent node until its current parent departs. We believe this is reasonable as incidents influencing the end-to-end bandwidth such as link failures are happening less frequently compared with the joining/departing events of the nodes on the multicast overlay. We use RanSub [24], which is a scalable approach for distributing the changing, uniform random subsets of global information to all nodes on a tree-like overlay, as our basic protocol, and extend it for estimating the lifetime model. However, any other protocols which provide a random subset service (such as [23] and [25]) could also serve as the basic protocol. We introduce RanSub briefly in the followings, the detailed description could be found in [24].

RanSub distributes random subsets of participating nodes throughout the tree using a bottom-to-top procedure called **collect** round and a top-to-bottom procedure named **distribute** round periodically. The **collect** messages start at the leaves and propagate up the tree, leaving state at each node along the path to the root. During the **collect** round, each node reports a random subset of its descendant nodes to its parent node, as well as the number of its descendant nodes. After the completion of the **collect** round, the **distribute** messages start at the root and travel down the tree. During the **distribute** round, each node will receive a fixed-sized random subset of the nodes which does not include any of its descendant nodes from its parent node, as well as the size of the overlay tree. Meanwhile, this node will generate a fixed-sized random subset for each children node of itself, which contains no descendant nodes of that children node. Clearly, for each **collect/distribute** round, a node just needs to communicate with its parent node and its children nodes, thus the overhead is $O(1)$. Moveover, as the information of random subsets of the nodes is not changing very dynamically, it is not necessary for an overlay to run the protocol frequently, thus it will not impose substantial overhead on the multicast overlay.

We extend the original RanSub protocol by carrying the age information on the messages for lifetime model estimation. In our protocol, a node reports a random subset of its descendant nodes with their ages, and obtains random subset of non-descendant nodes with their ages. Moreover, some information will be propagated during the **distribute** round of the protocol. First of all, one node will be responsible for deriving the lifetime model of the nodes currently on the overlay from its random subset, using the method we have suggested in Section 3.1. In our design, this task is conducted by the source node of the tree, but it could also be performed at any other positions. The parameters of the Pareto lifetime model α and β are distributed to all the nodes on the overlay as a part of the **distribute** message, and will be sent to any new joining nodes by the bootstrap nodes. With the lifetime model, each node will estimate its individual reliability as the hazard rate according to its age. The hazard rate is estimated as follows: Suppose the lifetime model for nodes on the overlay is modeled with the shifted Pareto distribution as $F(x) = 1 - (1 + x/\beta)^{-\alpha}$, then for a node at age x, according to the definition [32], its individual hazard rate is

$$h(x) = \frac{f(x)}{1 - F(x)} = \frac{\alpha}{\beta + x}$$

Here $f(x)$ is the PDF of the nodes' lifetimes. With the individual node's hazard rate, each node estimates the reliability of the streaming path from the source node to itself, and pass it to its children nodes in the **distribute** messages. The methodology for obtaining the path reliability will be discussed in the following subsection.

3.2.1 Parent Selection Algorithm

First we consider the parent selection algorithm. When a new node joins, it will look for a node as its parent which could provide a stable service. Since the lifetime is heavy-tailed, the longer a node lives, the longer it will be expected to survive. So, selecting the oldest node is a good choice. On the other hand, on a tree-like multicast overlay, the streaming is pushed from the source node down to the leaf nodes. For a single node, the departure of any ancestor node will cause a discontinuity of the streaming service, and the shorter the depth of a node, the less likely it will suffer an ancestor departure because of its fewer ancestor nodes. So, for a new node, choosing a parent node with the minimum depth is also reasonable.

In our approach, each node calculates the reliability of the path from the source node to itself as the accumulative hazard rate that any node on the path may fail. Suppose for node n_s with age x_s, there are $s - 1$ nodes between the source node to itself (but not including it), denoted as $n_1, n_2, \ldots, n_{s-1}$, at the age of $x_1, x_2, \ldots, x_{s-1}$ respectively, then the accumulative hazard rate for the nodes on the path $\{n_1, n_2, \ldots, n_{s-1}\}$ is

$$H(n_s; x_1, x_2, \ldots, x_{s-1}) = \frac{\sum_{i=1}^{s-1} \frac{\partial(1-S(x_1,x_2,\ldots,x_{s-1}))}{\partial x_i}}{S(x_1, x_2, \ldots, x_{s-1})} = h(x_1) + h(x_2) + ... + h(x_{s-1}) \quad (1)$$

here $S(x_1, x_2, \ldots, x_{s-1})$ is the survive probability of the nodes on the path $\{n_1, n_2, \ldots, n_{s-1}\}$ as $\prod_{i=1}^{s-1}(1 + x_i/\beta)^{-\alpha}$. We can see that the accumulative hazard rate is the sum of the hazard rates for all the nodes on the path, which conforms to the meaning of the hazard rate concept and could be interpreted as the rate that any node on the path $\{n_1, n_2, \ldots, n_{s-1}\}$ fails instantaneously.

The path reliability could be easily obtained with the **distribute** message of RanSub: for the node n_s, it receives a value of the accumulative hazard rate as $\sum_{i=1}^{s-1} h(x_i)$ from its parent, and calculates the new accumulative hazard rate by adding the hazard rate of itself as $\sum_{i=1}^{s} h(x_i)$, and sends the new accumulative hazard rate to its children nodes.

In this way, the reliabilities of all the paths on the overlay are updated periodically with the extended RanSub protocol. In our "prediction" parent selection strategy, when a new node joins, it contacts the bootstrap node for a random subset of the overlay nodes, and queries for all the nodes in it which could serve as its parent for their path reliabilities in the accumulative hazard rates, then it simply choose the one with the lowest value to connect. Following is the complete parent selecting algorithm using the "prediction" strategy described in pseudo-code:

Algorithm 1 *NodeJoin(n_{join})*

```
1. Get a random subset of the overlay nodes from the bootstrap node;
2. If none of the nodes in the subset could support one more child, repeat Step 1;
3. For each node $n_i$ eligible in the subset, query for $n_i$'s accumulative hazard rate $H(n_i)$
   calculated by Equation (1);
4. Choose the node with the smallest accumulate hazard rate as the parent to connect;
5. Return.
```

3.2.2 Preemption Algorithm

We also consider the preemption situation in which a disconnected node, due to its parent departure, replaces another node on the overlay. Unlike the new nodes, the disconnected nodes are not joining as a single node, but as a disconnected sub-tree with more than one nodes. Moreover, the disconnected nodes have been on the overlay for some time, which indicates better reliability than the new joining nodes with a age of zero. For the preemption algorithm, the disconnected nodes should be relocated at a proper position on the overlay rather than at the leaves. Usually people try to build a multicast tree as compact as possible by putting a node with larger bandwidth at a higher position, so, for the preemption strategy, a node with larger out degree preempting a node with smaller out degree is a good choice. On the other hand, as in the parent selection problem, a node with longer age is expected to be more stable than a node with shorter age, thus an older node preempting a younger one is also a reasonable choice.

In our first preemption strategy called "preemption by descendant", we believe that people should consider the importance of a node in providing service on the overlay. As all the descendant nodes of a disconnected node receive streaming from it directly or indirectly, the more descendant a disconnected node has, the more important it is on the overlay. So, in the "preemption by descendant" strategy, a node with more descendant nodes preempts a node with fewer descendants.

For the second preemption strategy called "preemption by service" which is more elaborate, each node estimates the amount of the service it is supposed to provide to its descendants. Suppose a node n at age x, its hazard rate is $h(x) = \alpha/(\beta + x)$, then, the length of time it is expected to be online in future could be approximated as $1/h(x) = (\beta + x)/\alpha$; if the node n has m descendant nodes, then the expected amount of service it will provide in future is

$$S(n;x,m) = \frac{m}{h(x)} = \frac{m(\beta + x)}{\alpha} \tag{2}$$

Since each node knows the number of its descendant nodes during the **collect** round of the extended RanSub protocol, and could calculate its hazard rate according to its age and the lifetime model during the **distribute** round, the estimated amount of service for each node could be easily obtained by the node itself. In the "preemption by service" strategy, a node with a larger amount of estimated service preempts a node with a smaller amount of estimated service.

Now we describe the whole preemption algorithm: First, a disconnected node chooses a node with the best path reliability it knows regardless of whether or not it could support a new child; then it will try to preempt one of the node's current child following one of the strategies on the condition that there is no spare degree; if the preemptions fail on all the children nodes, it will try the second best potential parent node until it finds one which it could connect as a children node. If the disconnected node can not find a parent node by preemption, it will select a leaf node randomly as its parent. On the other hand, the node being preempted will look for its new parent node by executing the same preemption algorithm. Following is the complete preemption algorithm described in pseudo-code.

Algorithm 2 *NodePrempt($n_{disconnect}$)*

```
Rank nodes in $n_{disconnect}$'s random subset as an ascending list according to their accumulative hazard rates calculated by Equation (1);
For the $i$th node $n_i$ of the ascending list from the beginning
    If $n_i$ can support one more child
        $n_{disconnect}$ connects to $n_i$ as parent;
    Else
        If $n_{disconnect}$'s service amount $S(n_{disconnect})$ calculated by Equation (2) is larger than the one of a $n_i$'s children node $n_{ic}$ or //preemption by service
        If $n_{disconnect}$'s number of descendants $m$ is larger than the one of a $n_i$'s children node $n_{ic}$ //preemption by descendant
            $n_{disconnect}$ connects to $n_i$ as parent;
            NodePrempt($n_{ic}$);
            Return;
$n_{disconnect}$ randomly chooses one node in its subset, and traces along the tree to a leaf node, then connects to this leaf node as its parent;
Return.
```

Finally, our protocol relies on each node to report its individual information such as the age or the descendant number during the parent selection or node preemption. Nodes may have incentive to lie with these information. To guard against the dishonest behaviors, some anti-cheating mechanism is required. Currently there are some works on this problem, such as the witness system in [26] or the referee system in [29], both can be integrated into our protocol without much modification. However, this issue is beyond the scope of this paper, and we assume that nodes are honest in our study.

4 Simulation Setup

We evaluate our protocol by simulation. We compare the "prediction", "preemption by descendant" and "preemption by service" strategies which are essential in our protocol with a number of alternative parent selection and node preemption strategies, and an event-driven simulator is developed for the comparison. We use the synthetic traces of the nodes' joining and leaving events as well as the half synthetic traces derived from our PPLive measuring results in simulation to study the stability of the overlays.

For the events of nodes joining and departing, we use synthetic traces as well as half-synthetic traces in our simulation. In the synthetic traces, we assume that nodes are joining the overlay following a Poisson process, and their lifetimes follow the shifted Pareto lifetime model. Specifically, we assume that the nodes' entry rate is $\lambda = N(\alpha - 1)/\beta$, here α and β are the parameters of the shifted Pareto lifetime model, and N is the stable population of the nodes on the overlay, which is $1,000$ in our simulation. We also make half synthetic traces based on the PPLive measurement results. Since we have 500 nodes' residual lifetimes, and according to Theorem 1 and Theorem 2, the distribution of the ages and the residual lifetimes are identical, we can simply use one random record as a node's age and use another random record for the node's residual lifetime, thus making up the whole lifetime.

For each node in the trace, a random outgoing degree is assigned according to the measurement results in Table 1. Suppose the media's playback rate is 500 Kbps, then for a university node, its out degree is above 2, and for a home node with Cable modem, the out degree ranges from 1 to 2, and a DSL home node will have the out degree of 0 or 1. For the unknown nodes, the out degree is randomly assigned. However, we allow a maximum out degree of 20, which is 10 Mbps in the outgoing bandwidth. We build an initial overlay with $1,000$ nodes to start the simulation. The initial overlay is built randomly but compact, which means all the outgoing degrees of the non leaf nodes are occupied. In our simulation trace, there are more than $10,000$ new nodes joining and departing during the simulation, which changes the initial overlay greatly, so we believe that the initial overlay influences the simulation results trivially.

We compare the strategies presented in Section 3 with a bunch of alternative strategies in the simulation. Those strategies are either already engaged in real world systems or have been systematically studied in previous works. We will show that our algorithms outperform them at a reasonable cost. Following we describe the strategies under evaluation in details. We assume that a random subset service such as [23], [24], and [25] is available for the nodes on the overlay.

For the parent selection strategy, we consider the following options:

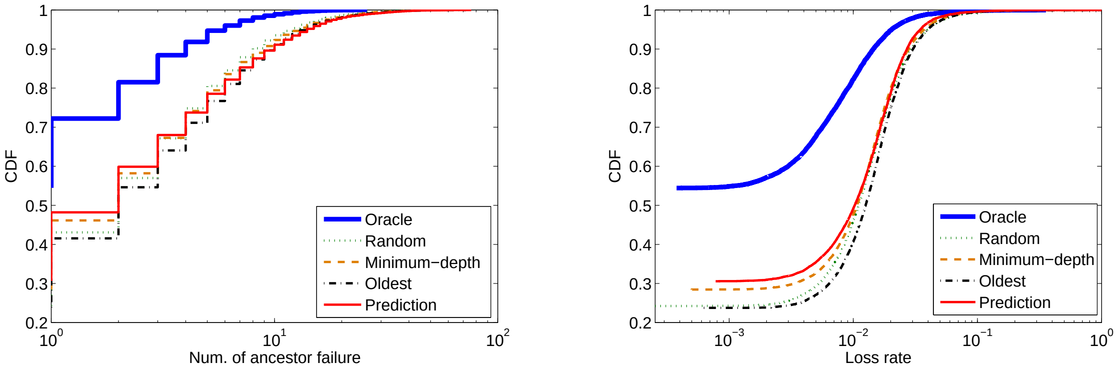
- Oracle: In this strategy, a new joining node will only choose a node which will depart later than itself to connect to as the parent. This strategy is impractical in real world systems as the node has no idea of other nodes' departure times.
- Random: In this strategy, a new joining node just chooses a node in its random uniform subset randomly which has spare outgoing degree as its parent.
- Minimum-depth: In this strategy, a new joining node chooses an available node with the minimum depth on the overlay tree, as smaller depth indicates fewer intermediate nodes from the source node to the new node itself.
- Oldest: In this strategy, a new joining node chooses an available node with the longest age to connect. As is indicated by the heavy-tailed lifetime, older nodes are more reliable than the younger ones.
- Prediction: This is the strategy we have discussed in Section 3 and implemented in Algorithm 1, in which a new joining node chooses an available node with best path reliability.

When a node departs the overlay, all its children nodes will look for a new parent with a preemption strategy. We consider the following options:

- No preemption: In this strategy, a disconnected node simply rejoins the overlay by connecting to a leaf node which has the minimum depth chosen from its random subset.
- Preemption by degree: In this strategy, a node with larger outgoing degree preempts a node with smaller degree. The implication behind is that placing nodes with larger out degree at higher positions on the overlay helps to build a compact multicast tree.
- Preemption by age: In this strategy, a node with longer age preempts a node with shorter age, as the older nodes tend to be more reliable than the younger ones.
- Preemption by descendants: This is the strategy we have discussed in Section 3 and implemented in Algorithm 2, in which a node with more descendants preempts a node with fewer descendants.
- Preemption by service: This is the strategy we have discussed in Section 3 and implemented in Algorithm 2, in which a node with more estimated service preempts a node with less estimated service.

We implement these strategies in our simulator and make a detailed comparison among the different strategy combinations. The results are presented in Section 5.

Finally, for evaluating the overlay stability, we consider the following two metrics: 1) the times a node suffers ancestor departures during its lifetime; and 2) the loss rate caused by ancestor departures. When a node departs, all of its descendant nodes will suffer an ancestor failure, and a discontinuity in the streaming service will be caused. As in [20], we assume each incident of an ancestor failure will lead to a length l of discontinuity in the streaming service, and a node's loss rate is the ratio of the total length of discontinuity divided by the total time of the streaming service it is supposed



(a) (b)

Fig. 2. CDF of the (a) number of ancestor failures and (b) loss rate under different parent selection strategies

to receive. In our simulation, we set $l = 5$ seconds, as this is a typical length required for a disconnected node to find a new parent and have a good performance after a slow start phase of a congestion-control protocol (e.g., TCP or TFRC). One question remaining is how we handle the preemptions, as they could also be viewed as some kind of failures for the nodes being replaced. However, in our simulation, we do not count preemptions into the ancestor departure events and the loss rate, as for most cases, we believe preemption can be conducted very gracefully: a node will not cut off the streaming to the preempted children node until the preempted node receives a full rate service from its new parent.

Besides the stability, we also study the overhead of our protocols and the service latency of the overlays formed by running different strategies. For the overhead, we investigate the number of the preemptions caused by a single disconnection; and for the service latency, we deploy the overlay upon a topology generated by the GT-ITM transit-stub model [28], and study the overlay stretch under different strategy combinations.

5 Evaluation Results

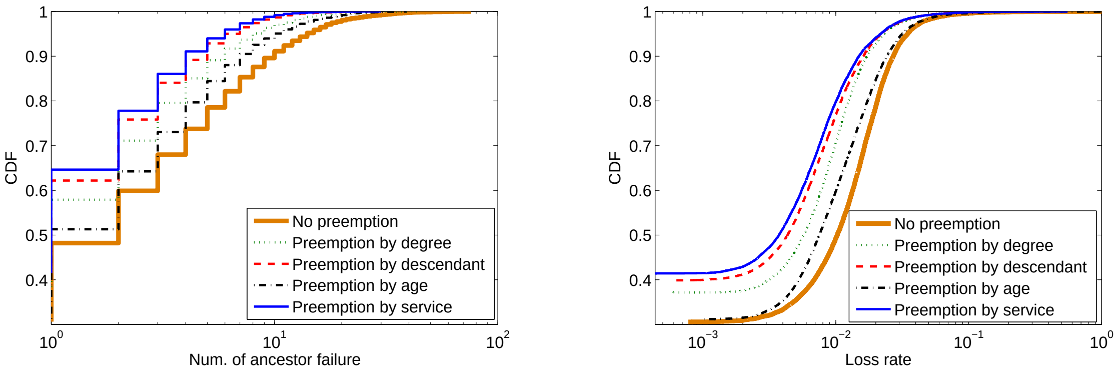
For the default simulation settings, we use the lifetime model derived from our PPLive measurement result (i.e. $\alpha = 3.667$, $\beta = 3909.51$). We start the experiment on an initial overlay with $1,000$ nodes, whose ages and residual lifetimes follow the shifted Pareto distribution in Theorem 1 and Theorem 2. The size of the random subset is set as 50, and we simulate a period of multicast service for $20,000$ seconds, during which more than $10,000$ nodes have joined and have a lifetime following the Pareto lifetime model.

5.1 Parent Selection Strategies

We first evaluate the parent selection strategies described in Section 4. For fairness, we use the “no preemption” strategy to handle all the disconnected nodes caused by parent departures. When a node departs the overlay, its descendant nodes will encounter an ancestor failure incident, and a discontinuity penalty l will be deducted. At the end of a simulation run, we can obtain the total numbers of ancestor departures and the loss rates for all the nodes in the trace. We plot the CDF of the ancestor departure events and the loss rates in Fig. 2(a) and Fig. 2(b) respectively. From the figures we can see that the “minimum-depth” strategy outperforms the “oldest” strategy and the “random” strategy, while the latter two strategies have almost the same performance in preserving the overlay’s stability. Our observation is different from the result in [4], due to the reason that in [4], all the descendants of a disconnected node reselect their parents, while in our preemption strategy, only the children of the disconnected node find their new parents. We believe the “no preemption” strategy contributes the major part of the instability observed and masks the difference between the “random” strategy and the “oldest” strategy. However, we could still find that by engaging the “prediction” strategy, the stability outperforms all the other practical strategies evaluated. Note here we have also observed ancestor failures for the “oracle” strategy, which should not exist since in this strategy a node will only choose to connect to a parent which will depart later than itself. However, in the simulation, sometimes it is hard for a node with very long lifetime to find such an available leaf node satisfying the requirement within its limited sized random subset, in this case, we use the “prediction” strategy instead. Finally, we find that actually all these overlays are not very stable, regardless of what kind of parent selection strategy is engaged. For example, even with the best “prediction” strategy, on average a node will experience 3.65 incidents of ancestor failures during its lifetime. We believe the instability is caused by the naive “no preemption” strategy, in which all the disconnected sub-trees rejoin the overlay at the leaves. We show in the next section that a proper preemption strategy could improve the overlay’s stability intensively.

5.2 Preemption Strategies

We evaluate the preemption strategies described in Section 4 in this experiment. In our simulation, we use the “prediction” strategy to handle the new joining nodes and the rejoining nodes, which means when a new joining node or a rejoining node looks for its parent node, it will choose to connect to the one with the best path reliability in its random subset. In the following part of this paper, when we mention a strategy combination, such as “prediction + preemption by descendant”, we mean that all the new joining nodes and the rejoining nodes use the same algorithm to look for their parents. However, when the preemption strategy is “no preemption”, there is no such a parent searching process, as a disconnected node just picks an available leaf node as its new parent. We plot the experimental results in the CDF of the number of the ancestor failures and the



(a)

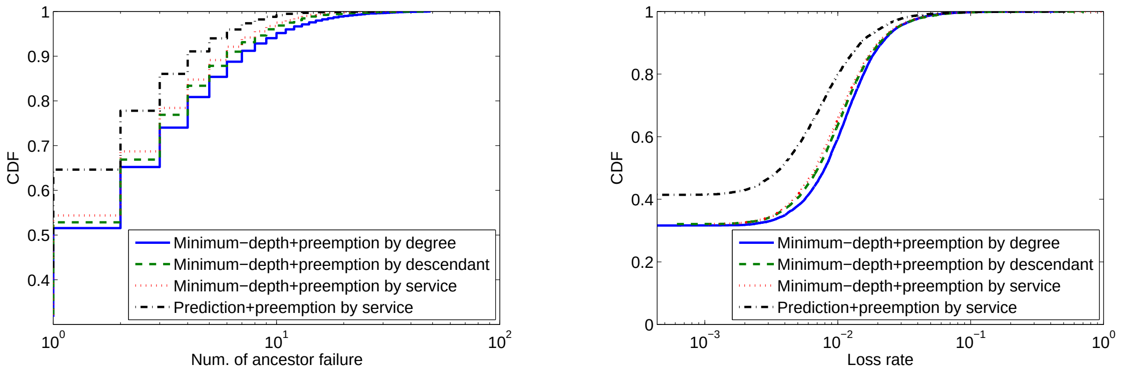
(b)

Fig. 3. CDF of the (a) number of ancestor failures and (b) loss rate under different preemption strategies

loss rates in Fig. 3(a) and Fig. 3(b) respectively. Note that the curves marked as "no preemption" are just the curves of the "prediction" strategy in Fig. 2, which gives the best stability for the parent selection strategies. From the simulation results, we can see that applying any kind of preemption strategy will improve the overlay stability considerably, and among these strategies, "preemption by degree" outperforms "preemption by age", which conforms to the observation in [20]. However, we find that both the strategy of "preemption by descendant" and the strategy of "preemption by service" are better than all the other strategies, and "preemption by descendant" performs slightly worse than "preemption by service". We conclude that proper combination of parent selection strategy and preemption strategy can improve the stability of the multicast overlay greatly, as with the combination of "prediction + preemption by service", a node will experience 1.58 incidents of ancestor failures during its lifetime on average.

5.3 What is the Best Combination?

As we have evaluated the parent selection algorithms and the preemption algorithms separately in the previous sections, one natural question is: what is the best combination? We do observe very good stability with the combinations of "prediction + preemption by descendant" and "prediction + preemption by service", so, a more concrete question is: Is the strategy of "preemption by descendant" as good as the strategy of "preemption by service", regardless of the parent selection strategy engaged? Our answer to this question is "almost yes". The detailed reason is: in the "preemption by service" strategy, the stability of the disconnected node (as $\frac{1}{h(x)}$) and the importance of the node (as m) is combined, while in the "preemption by descendant" strategy, only the importance of the disconnected node is considered. However, the reason that we do observe similar performance in Fig. 3 is that when the parent selection strategy of "prediction" is used by new joining and rejoining nodes, these nodes will choose a node with a stable path to connect as a parent. The implication behind is that a stable node will have more



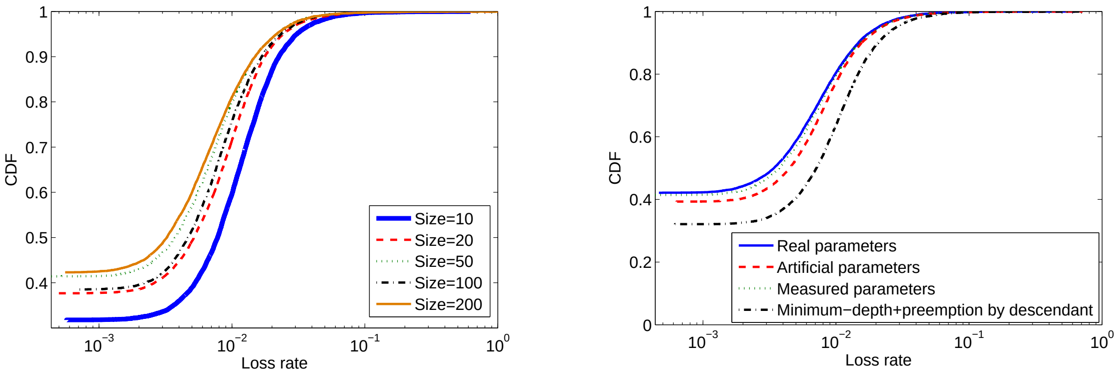
(a) (b)

Fig. 4. CDF of the (a) number of ancestor failures and (b) loss rate under several strategy combinations

descendants than an unstable one, in this case, the “preemption by descendant” strategy implicitly compares the node’s stability and the node’s descendants as well. And this implication will also be held when other parent selection strategy is engaged. To support our point, we simulate the system with the strategy combinations of “minimum-depth + preemption by descendant” and “minimum-depth + preemption by service” with the same setting as in the previous experiments, and plot the results in Fig. 4. The simulation result shows that the “preemption by descendant” works slightly worse than the “preemption by service” strategy, but their performance are very close to each other. We also plot the results of the combinations “prediction + preemption by service” and “minimum-depth + preemption by degree” in the figure, and have observed that the combination “prediction + preemption by service” obviously outperforms the best combination of “minimum-depth + preemption by descendant” among the combinations in which no active lifetime model estimation is engaged, and the combination “minimum-depth + preemption by descendant” outperforms the combination “minimum-depth + preemption by degree”. Finally, we conclude that the best strategy combination is “prediction + preemption by service”, while the combination “prediction + preemption by descendant” is also very good at preserving the overlay’s stability, and both combinations actively estimate the lifetime model.

5.4 Is Accurate Prediction Necessary?

In the previous sections, we have identified that the strategy combination of “prediction + preemption by service” as the best way to improve the overlay’s stability. As in both the “prediction” strategy and the “preemption by service” strategy, a node must estimate its hazard rate based on its age x (as $h(x)$), with the lifetime model parameters α and β derived by the source node, it is natural to ask whether or not an accurate estimation of the lifetime model is important for these strategies. During the experiments of the previous sections, we always set the random subset size as 50, which means the source



(a) (b)

Fig. 5. CDF of loss rate (a) with different random subset sizes and (b) with different lifetime model parameter sources

node derives α and β from the ages of 50 random nodes on the overlay. In the following experiment, we study the overlay's stability with different random subset sizes. In Fig. 5(a), the CDFs of the loss rate are plotted under the strategy combination of "prediction + preemption by service" with the random subset size ranging from 10 to 200 nodes. Intuitively, larger random subset leads to a more accurate estimation, and consequently gives a better stability performance for the overlay under evaluation. However, from the experimental results we can see that although a very small random subset size (such as 10) does lead to bad stability for the overlay, when the random subset size exceeds 20, the improvement caused by a larger random subset size is trivial. Note that there is another reason for the improvement on stability, as a larger random subset of nodes means more choices for parent selection and preemption, thus leads to better performance, and we believe that the main reason of the improvement on the overlay stability lies in the second factor. Overall, we show that our protocol could have good performance without introducing a large overhead caused by maintaining large sized random subsets, and we believe that accurate estimation of the lifetime model is unnecessary.

To further support this argument, we perform another experiment in which the lifetime model's parameter α and β are pre-configured instead of being measured during the execution of the protocol, which means the nodes will estimate the hazard rate $h(x)$ based on static parameters. Actually we set the artificial parameters as $\alpha' = \alpha+1$, and $\beta' = 2\beta$. We show our results in Fig. 5(b), three situations are evaluated: the strategy combination of "prediction + preemption by service" with the pre-configured artificial parameters, with the measured parameters during the protocol execution and with the real parameters used in generating the trace, we also plot the result of "minimum-depth + preemption by descendant" combination for comparison. To our surprise, overall there is no great difference regarding the overlay's stability, when lifetime model parameters from three different sources are engaged. Our findings indicate that as long as we integrate the ages of the nodes with their depths or descendants in the parent selection algorithm or preemption algorithm in proper ways, the overlay's stability gets improved, but accurate estimation of the lifetime model parameters is not necessary; as for most cases, the protocol only

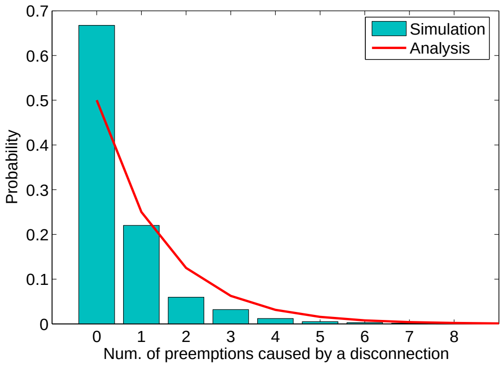


Fig. 6. Distribution of the number of preemptions caused by a single disconnected node

needs to decide the proper parent to connect to or the proper node to preempt among only 50 candidates, and can make the right choice with lifetime models of very roughly estimated parameters. Another important meaning for this experiment is: the strategy combination of "prediction + preemption by service" can work very well with some pre-configured lifetime model parameters obtained from previous measurement work, even under the condition that the pre-configured parameters are not very accurate, as we can see from the figure that with the artificial parameters, the combination of "prediction + preemption by service" still obviously outperforms the "minimum-depth + preemption by descendant" combination, which is the best strategy combination without active lifetime model estimation. Thus, we conclude that our protocol could be more light-weighted with pre-configured parameters, without estimating the lifetime model at the source node.

5.5 How Many Preemptions will be Engaged?

In our protocol, we use a preemption algorithm to handle nodes failures, which means a disconnected node will find a suitable parent and tries to replace one of its children nodes if necessary, and the node being replaced will follow the same preemption algorithm until the last replaced node rejoins the overlay as a leaf node. In our simulations, we assume the preemptions are carried out gracefully, which will not cause service discontinuity for the descendant nodes of the preempted node. However, if a departure incident causes too many preemptions, the system's performance will surely be degraded, due to the following reasons. First, the preemption operation requires information exchanging such as the estimated service amounts, the hazard rates and the decision notifications among the associated nodes, all these operations will result in some overhead. Second, although we assume the preemption is gracefully conducted, it will certainly influence the quality of the streaming service for the descendants of the parent node: because during preemption, the parent will support the preempted children node and the preempting children node simultaneously until the preempted one receives a full rate streaming from its new parent, the burden is generally beyond the parent node's regular capacity.

Because of the above concerns, it is necessary to investigate how many preemptions a disconnected node will cause in the preemption strategies, as too frequent preemption operations will degrade the overlay's performance. We study the problem with two approaches: first we analyze mathematically the probabilities of $1, 2, 3...$ preemptions caused by one disconnected node; then we count the numbers of the preemptions caused by one disconnected node during the simulation and study the distribution statistically. Fortunately, both the analytical studies and the simulations indicate that with our preemption strategy, very few preemptions are caused by a single disconnected node, thus preemption will not impose a heavy burden on the overlay or influence the quality of the streaming service severely.

We analyze the problem as follows. When a node has been disconnected due to its parent failure, it will try to rejoin the overlay with the nodes in its random subset. For each trial, there will be three possible results: 1) the disconnected node can not choose it as the new parent because it does not have enough bandwidth and the disconnected node can not preempt any of its current children nodes; 2) the disconnected node can choose it as the new parent by preempting one of its children; 3) the disconnected node can choose it as the new parent without preemption, which means the node has spare bandwidth for a new children node. Suppose the tree is balanced and compact, with only the leaf nodes having spare bandwidth, the degree is d and the height is h, then the probability that a node has spare bandwidth is $\frac{d^h}{2d^h-1}$, as there are d^h leaf nodes among the total $2d^h - 1$ nodes. If we sort all the nodes according to some criteria, such as degree, age, number of descendants or amount of future service, then for a particular node, the probability that a random node is ahead of it in this sorting is $1/2$. For the preemption strategy, the probability of the first result now could be derived as $\Pr[r_1] = \frac{d^h}{2d^h-1}(\frac{1}{2})^d$; the probability of the second result could be derived as $\Pr[r_2] = \frac{d^h-1}{2d^h-1}(1 - (\frac{1}{2})^d)$; and the probability of the third result is $\Pr[r_3] = \frac{d^h-1}{2d^h-1}$. Obviously, only the second result will cause one more preemption. If we let $d^h - 1 \approx d^h$, then the probability that s preemptions are caused by a disconnected node is $P(s) = (\frac{1}{2})^{s+1}(1 - \frac{1}{2^d})^s(1 + \frac{1}{2^d})$, which could be approximated as $P(s) = (\frac{1}{2})^{s+1}$, when $\frac{1}{2^d}$ is very small (the average out degree d is 6.2 in our simulation). Simply speaking, the probability of having one more preemption caused by a single disconnected node is decreasing exponentially. We also count the numbers of preemptions caused by nodes' departures in the simulation, and find that the longest preemption chain is 8, and the average number of preemptions caused by a disconnected node is 0.534, which is very small. The statistical result from simulation as well as the analytical one are presented in Fig. 6. We can see that actually the chance of having one more preemption decreases even faster than the exponential curve, and there are nearly 70% of the disconnected nodes incurring no preemption at all.

5.6 Latency and Overlay Stretch

Although we aim to improve the stability for the multicast overlay, it is also important to examine the end-to-end service latency for the nodes on the overlay, as large delays

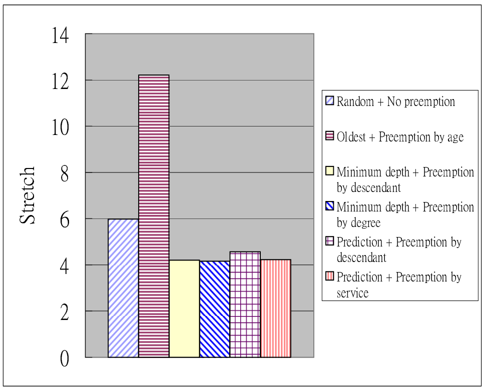
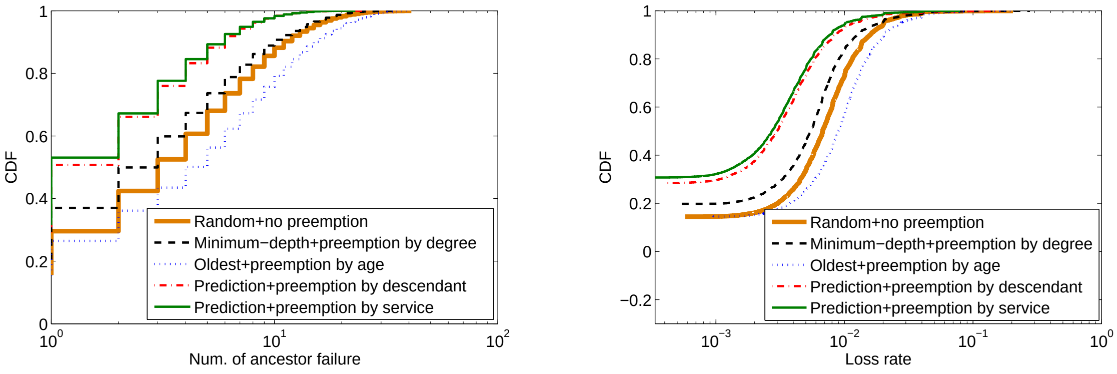


Fig. 7. Stretches for the overlays formed by running a number of strategy combinations



(a)

(b)

Fig. 8. CDF of the (a) number of ancestor failures and (b) loss rate under several strategy combinations using the half synthetic trace

are intolerable for the media streaming service, especially for live media contents. In this experiment, we simulate an underlying network topology composed of more than $6,000$ nodes, and randomly deploy the overlay formed by different strategy combinations upon the topology to study the overlay latency. We use the GT-ITM transit-stub model [28] to generate the underlying network topology. Link delays between two transit nodes, one transit node and one stub node, and two stub nodes are chosen uniformly between $[15, 25]$ ms, $[5, 10]$ ms and $[2, 4]$ ms respectively. To measure the latency, we use the overlay stretch as our metrics, which is defined as the ratio of the sum of the service delays for all the nodes on the overlay divided by the sum of the delays if these nodes receive the streaming service from the source with direct unicast channels in the underlying network. We study the representative combinations of “random + no preemption”, “oldest + preemption by age”, “minimum-depth + preemption by descendant”, “minimum-depth + preemption by degree”, “prediction + preemption by descendant” and “prediction + preemption by service”, and present the results in Fig. 7. We can see that the strategy combination of “oldest + preemption by age” has the worst performance, because both of its strategies only consider the lifetime properties of the nodes, ignoring the overlay structural information. The combination of “random + no preemption” performs the second worst due to the reason of randomness at the joining of the nodes and the lack of preemptions. The combination "minimum depth + preemption by degree" has the lowest stretch, since in its strategies, the only concern is to decrease the depth of the overlay tree. For the other combinations in which both the nodes' ages and the overlay structure are considered, their stretches are all very close to the stretch achieved by the "minimum depth + preemption by degree" combination. Thus, we conclude that although the preferred strategy combinations of "prediction + preemption by descendant" and "prediction + preemption by service" do not deliberately reduce the overlay's depth, their performances on the service latency are still very good, without much difference compared with the combination of "minimum depth + preemption by degree", which achieves the lowest overlay stretch.

5.7 Simulation with Half Synthetic Trace

For the last experiment, we evaluate the strategy combinations using the residual lifetime data we have obtained from the PPLive measurement in Section 2. As we have only obtained 500 nodes' residual lifetimes during the experiment, and a full trace with more than $10,000$ nodes is required, we use the following method to generate the half synthetic trace: for a node, we uniformly select two random residual lifetime records from the PPLive measurement result and use the sum as its lifetime, as it is indicated by Theorem 1 and Theorem 2 that the distributions of the residual lifetime and the age are identical. Besides the lifetime generation, we follow the same procedure mentioned in section 4 to generate the trace with an initial overlay of $1,000$ nodes, and more than $10,000$ new nodes joining during the simulation. We study the stability of the overlays formed by running the strategy combinations of "random + no preemption", "oldest + preemption by age", "minimum-depth + preemption by degree", "prediction + preemption by descendant", and "prediction + preemption by service", and plot the CDFs of the ancestor failure events and the loss rates in Fig. 8. The experimental results conform to the results presented in Fig. 2, Fig. 3 and Fig. 4, and again we observe that the combinations of "prediction + preemption by descendant" and "prediction + preemption by service" outperform the other strategy combinations. However, by comparing the figures in Fig. 3 and Fig. 8, we find that the performance improvement made by the combinations "prediction + preemption by descendant" and "prediction + preemption by service" are less prominent compared with the results obtained using the full synthetic trace. We explain this with the fact that in our half synthetic trace, there will be less records of nodes with very short lifetime compared with the trace generated by the shifted Pareto lifetime model in the previous experiments, as a node's lifetime is the sum of two random residual lifetimes in the PPLive measurement. So, with the absence of these extremely unstable nodes, the relative performances of the strategy combinations as "prediction + preemption by descendant" and "prediction + preemption by service" are degraded somewhat.

6 Conclusion

In this paper, we aim to improve the stability of tree-like multicast overlays. Our basic idea is to exploit and integrate the nodes' lifetime information, which is reported to be heavy-tailed in many works, as well as the overlay's structural properties. To have an accurate understanding for the lifetimes of the nodes on a multicast overlay, we study the PPLive system, which is a popular IPTV system, and find that the shifted Pareto distribution model could be used to describe the residual lifetimes of the nodes very accurately, and the nodes present the same properties on their residual lifetimes regardless of their bandwidths. We also discuss the relationships between the nodes' ages, lifetimes and residual lifetimes formally and use the results as the theoretical foundation for designing our protocol. Based on the practical measurement and the theoretical analysis, we develop an overlay construction protocol. Concretely, we extend the RanSub protocol for actively estimating the lifetime model, and present a set of algorithms for parent selection and node preemption, in which the strategies of "prediction", "preemption by descendant" and "preemption by service" are proposed.

We conduct a set of experiments based on simulations to examine the performance of the overlay under our protocol, and compare it with the performances of the overlays under a number of other strategies for parent selection and node preemption. We use the full synthetic traces generated by the shifted Pareto lifetime model and the half synthetic trace derived from the PPLive measurement in our simulation experiments. We find that compared with the other strategies, our parent selection strategy and preemption strategies, which are essential in our protocol, could improve the overlay's stability considerably. We also find that our solution can work with limited sized random subsets and will not incur frequent preemption operations, making the protocol light-weighted. Moreover, although in our protocol the nodes' lifetime model is actively measured, we show that it is not necessary to have an accurate estimation for the lifetime model parameters, as the overlay could have a good stability even with inaccurate but informative pre-configured parameters. Finally, we show that the overlay formed by our protocol has very good performance regarding the service latency, although it is not our design objective.

References

[1] F. E. Bustemante and Y. Qiao, Friendships that last: peer lifespan and its role in p2p protocols, in: Proc. Intl. Workshop on Web Content Caching and Distribution (WCW'03), Hawthorne, NY, October 2003.

[2] S. Saroiu, P. K. Gummadi, and S. D. Gribble, A measurement study of peer-to-peer file sharing systems, in: Proc. Multimedia Computing and Networking (MMCN'02), San Jose, CA, January 2002.

[3] K. Sripanidkulchai, B. M. Maggs, and H. Zhang, An analysis of live streaming workloads

on the internet, in: Proc. Internet Measurement Conference (IMC'04), Taormina, Italy, October 2004.

[4] K. Sripanidkulchai, A. Ganjam, B. Maggs, and H. Zhang, The feasibility of supporting large-scale live streaming applications with dynamic application end-points, in: Proc. ACM SIGCOMM'04, Portland, OR, August 2004.

[5] Asia Pacific Network Information Centre. `http://www.apnic.net`

[6] America Registry for Internet Numbers. `http://www.arin.net`

[7] RIPE Network Coordination Centre. `http://www.ripe.net`

[8] V. N. Padmanabhan, H. J. Wang, and P. A. Chou, Resilient peer-to-peer streaming, in: Proc. IEEE ICNP'03, Atlanta, GA, November 2003.

[9] Y. Chu, A. Ganjam, and et al, Early experience with an internet broadcast system based on overlay multicast, in: Proc. USENIX Annual Technical Conference (USENIX'04), Boston, MA, June 2004.

[10] X. Liao, H. Jin, Y. Liu, L. M. Ni, and D. Deng, AnySee: Peer-to-peer live streaming, in: Proc. IEEE INFOCOM'06, Barcelona, Spain, April 2006.

[11] D. A. Tran, K. A. Hua, and T. T. Do, A peer-to-peer architecture for media streaming, IEEE Journal on Selected Areas in Communication, 22 (1) (2004) 121-133.

[12] M. Castro, P. Druschel, A. M. Kermarrec, and A. Rowstron, SCRIBE: A large-scale and decentralized application level multicast infrastructure, IEEE Journal on Selected Areas in Communication, 20 (8) (2002) 1489-1499.

[13] M. Castro, P. Druschel, and et al, SplitStream: High-bandwidth multicast in a cooperative environment, in: Proc. ACM SOSP'03, Lake Bolton, New York, October 2003.

[14] V. Venkataraman, P. Francis, and J. Calandrin, Chunkyspread: Multi-tree unstructured end system multicast, in: Proc. 5th Intl. Workshop on Peer-to-Peer Systems (IPTPS'06), Santa Barbara, CA, February 2006.

[15] D. Kostic, A. Rodriguez, J. Albrecht, and A. Vahdat, Bullet: High bandwidth data dissemination using an overlay mesh, in: Proc. ACM SOSP'03, Lake Bolton, New York, October 2003.

[16] D. Kostic, R. Braud, and et al, Maintaining high bandwidth under dynamic network conditions, in: Proc. USENIX'05, Anaheim, CA, April 2005.

[17] X. Zhang, J. Liu, B. Li, and T.-S. P. Yum, CoolStreaming/DONet: A data-driven overlay network for live media streaming, in: Proc. IEEE INFOCOM'05, Miami, FL, March 2005.

[18] Bittorrent. `http://www.bittorrent.com`.

[19] R. Sherwood, R. Braud, and B. Bhattacharjee, Slurpie: A cooperative bulk data transfer protocol, in: Proc. IEEE INFOCOM'04, Hong Kong, March 2004.

[20] M. Bishop, S. Rao, and K. Sripanidkulchai, Considering priority in overlay multicast protocols under heterogeneous environments, in: Proc. IEEE INFOCOM'06, Barcelona, Spain, April 2006.

[21] PPLive. http://www.pplive.com.

[22] D. Leonard, V. Rai, and D. Loguinov, On lifetime-based node failure and stochastic resilience of decentralized peer-to-peer networks, in: Proc. ACM SIGMETRICS'05, Banff, Canada, June 2005.

[23] V. Vishnumurthy and P. Francis, On heterogeneous overlay construction and random node selection in unstructured p2p networks, in: Proc. IEEE INFOCOM'06, Barcelona, Spain, April 2006.

[24] D. Kostic, A. Rodriguez, J. Albrecht, A. Bhirud, and A. Vahdat, Using random subsets to build scalable network services, in: Proc. USENIX Symposium on Internet Technologies and Systems (USITS'03), Seattle, WA, March 2003.

[25] J. Ganesh, A. Kermarrec, and L. Massoulie, Peer-to-peer membership management for gossip-based protocols, IEEE Transactions on Computers, 52 (2) (2003) 139-149.

[26] Y. Qiao and F. E. Bustamante, Elders know best - handling churn in less structured p2p systems, in: Proc. 5th IEEE Intl. Conference on Peer-to-Peer Computing (P2P'05), Konstanz, Germany, September 2005.

[27] Y. Tian, D. Wu, and K.-W. Ng, Modeling, analysis and improvement for bittorrent-like file sharing networks, in: Proc. IEEE INFOCOM'06, Barcelona, Spain, April 2006.

[28] E. W. Zegura, K. Calvert, and S. Bhattacharjee, How to model an internetwork, in: Proc. IEEE INFOCOM '96, San Francisco, CA, March 1996.

[29] G. Tan, S. A. Jarvis, and D. P. Spooner, Improving fault resilience of overlay multicast for media streaming, in: Proc. IEEE Intl. Conference on Dependable Systems and Networks (DSN'06), Philadelphia, PA, June 2006.

[30] G. Tan, S. A. Jarvis, X. Chen, D. P. Spooner, and G. R. Nudd, Performance analysis and improvement of overlay construction for peer-to-peer live media streaming, in: Proc. IEEE Intl. Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS'05), Atlanta, GA, September 2005.

[31] X. Hei, C. Liang, J. Liang, Y. Liu, and K. W. Ross, Insight into pplive: Measurement study of a large scale p2p iptv system, in: Proc. WWW 2006 workshop of IPTV services over World Wide Web, Edinburgh, Scotlant, May 2006.

[32] S. I. Resnick, Adventures in Stochastic Processes (Birkhäuser, Boston, 2002).

[33] A. Walters, D. Zage, and C. Nita-Rotaru. Mitigating attacks against measurement-based adaptation mechanisms in unstructured multicast overlay networks, in: Proc. IEEE ICNP'06, Santa Barbara, CA, Nov. 2006.