Understanding Multiple-tree-based Overlay Multicast

György Dán and Viktória Fodor
Laboratory for Communication Networks
School of Electrical Engineering
KTH, Rotal Institute of Technology
Stockholm, Sweden
E-mail: {gyuri,vfodor}@ee.kth.se

ABSTRACT

The two main sources of impairment in overlay multicast systems are packet losses and node churn. Yet, little is known about their effects on the data distribution performance. In this paper we develop an analytical model of a large class of peer-to-peer streaming architectures based on decomposition and non-linear recurrence relations. We analyze the stability properties of these systems using fixed-point analysis. We derive bounds on the probability that nodes in the overlay receive an arbitrary packet of the stream. Based on the model, we explain the effects of the overlay's size, node heterogeneity, loss correlations and node churn on the overlay's performance. We show how and under what conditions overlays can benefit from the use of error control solutions, prioritization and taxation schemes. Our findings lead us to the definition of an overlay structure with improved stability properties. Based on our results, we identify the components that are needed to achieve good data distribution performance via overlay multicast.

Keywords

Modeling, Overlay multicast, Data distribution performance

1. INTRODUCTION

The peer-to-peer paradigm has proved to be an efficient means both for file distribution to a large population of users, and for lookup services without the need for expensive infrastructure. Peer-to-peer multicast streaming overlays could serve content providers as a cheap and efficient alternative to CDNs for distributing live media to a large number of spectators. In peer-to-peer multicast, peers are organized or organize themselves into an application layer overlay and distribute the data among themselves. The main advantages are that the multicast is easy to deploy and it reduces the load of the content provider, since the distribution cost in terms of bandwidth and processing power is shared by the nodes of the overlay.

Successful small scale deployments of multicast overlays were reported in the order of a few hundreds of peers [1]. But despite a large number of proposed architectures ([2, 3, 4, 5, 6, 7, 8] and references therein) and a number of deployed systems [8, 9, 10], large-scale peer-to-peer multicast has not widely been used. This could be due to two reasons: *feasibility*, i.e., the lack of bandwidth resources to construct an overlay, and the lack of data distribution *performance* similar to that of point-to-point streaming in terms of end-to-end delay and packet loss probability.

Nevertheless, recent results show that bandwidth resources are not an obstacle. In [11] it was shown that the bandwidth resources contributed by the peers tend to be enough to support large scale systems. Even if they are not, architectures can provide different levels of performance to peers with different bandwidth contributions [12].

Less is known about the data distribution performance, such as the packet reception probability of the participating nodes. Most of the results in the literature are based on simulations, and focus on metrics like the time between tree disconnections, the depth of the overlay, the amount of control overhead and the link stress. There is a lack of understanding of how the parameters of the overlay (e.g., the number of distribution trees, the error control solutions employed) and the environmental dynamics (e.g., the number of nodes, node churn and losses due to network failures) affect the end-to-end delays and the packet reception probability.

The goals of this paper are twofold. First, to give an understanding of how and why the above factors and the policies proposed in the literature influence the data distribution performance of overlay multicast. Second, to give a tool for system designers to evaluate the performance of their proposals, and give guidelines on how to achieve good performance.

We consider overlay multicast systems based on multiple distribution trees and the push model, such as the ones in [3, 4, 5, 6, 7, 12]. Multiple trees offer two advantages: they ensure graceful quality degradation in dynamic overlays, where peers can leave during the streaming session and they enable nodes to contribute to the overlay with fractions of the stream bandwidth. The higher the number of trees, the smaller the fractions, so that nodes' output capacities can be better utilized. With multi-path transmission, parts of the stream reach the peers through independent overlay paths. Consequently a node receives large part of the streaming data even if some of its parent peers stop forwarding.

The contributions of the paper are the following. (i) We present a model to describe the probability that a peer in the overlay possesses an arbitrary packet of the data stream. (ii) We show that node churn can be treated as a form of packet losses. (iii) Based on the model, we show how factors, such as the overlay's size, heterogeneous loss probabilities, heterogeneous input and output capacities and loss correlations influence the data distribution performance of the overlays. (iv) We explain how the parameters of the overlay, such as the number of distribution trees, the error control schemes employed, the prioritization and taxation schemes affect the performance. (v) Based on our findings we propose a tree structure that

Technical report TRITA-EE 2007:026, January 2007
School of Electrical Engineering, KTH, Royal Institute of Technology

improves the scalability of the overlay with respect to the number of nodes.

We use simulations to validate the approximations of the model. We did not perform measurements for two reasons. First, our aim is to understand the effects of the various parameters on the overlays' performance, hence we need a controlled environment to validate the assumptions of the model. Second, we are interested in the performance of large scale overlays, but we do not have content that could attract thousands of viewers and we do not have access to traces of large streaming events.

The rest of the paper is organized as follows. In Section 2 we give a description of the considered overlays. We develop the analytical model and derive asymptotic bounds in Section 3. We evaluate the effects of losses in Section 4 and show how to model node churn in Section 5. We conclude our work in Section 6.

2. SYSTEM DESCRIPTION

In this section we describe the considered general overlay structure in Section 2.1, our assumptions regarding the overlay maintenance and the data distribution in Sections 2.2 and 2.3 respectively.

2.1 Overlay structure

The overlay consists of a root node and N peer nodes. The peer nodes are organized in t distribution trees. Each peer node is member of at least one tree, and in each tree it has a different parent node from which it receives data. We say that a node that is i hops away from the root node in tree e is in layer i of tree e. We denote the maximum number of children of the root node in each tree by m, and we call it the multiplicity of the root node.

Nodes can have children in up to d of the t trees, called the fertile trees of a node. A node is sterile in all other trees, that is, where it does not have any children. d is a system parameter. If a node r has enough capacity to forward data to γ^r children then we say that the node has a total of γ^r cogs in its fertile trees and has no cogs in its sterile trees. For $d > 1$ the nodes balance their cogs between trees, i.e., a node can have up to $\lceil \gamma^r/d \rceil$ cogs in each of its fertile trees. If we denote the maximum number of layers in the trees by L, then in a well maintained tree each node is $1 \leq i < L$ hops away from the root node in its fertile trees, and $L-1 \leq i \leq L$ hops away in its sterile trees.

By setting $d = t$ one gets the minimum breadth trees described in [4], and by setting $d = 1$ one gets the minimum depth trees evaluated in [4, 3, 6, 12, 13]. For $1 < d < t$ the number of layers in the overlay is $O(logN)$ as for $d = 1$. Fig. 1 shows an overlay for $t = 2$, $m = 2$ and $d = 1$. The solid black lines show the parent-child relations between the nodes in the overlay.

2.2 Tree management

The construction and the maintenance of the trees can be done either by a distributed protocol (structured, like in [3] or unstructured, like in [5]) or by a central entity, like in [4]. The results presented in this paper do not depend on the particular algorithm used, our focus is on the performance of the overlay as a function of the overlay's structure, rather than the efficiency of the tree maintenance algorithm.

The purpose of the tree maintenance algorithm is to find eligible parents for the nodes based on the parent selection criteria (e.g., closest to the root) and the nodes' priorities. We consider three aspects of the tree maintenance algorithm. First, it influences the number of layers in the overlay and the distribution of the nodes in the layers. Second, it influences how often a node loses its parent in a tree depending on the node's priority in the tree. We call this the inter-disconnection time, and denote it by Ω. Third, it influences

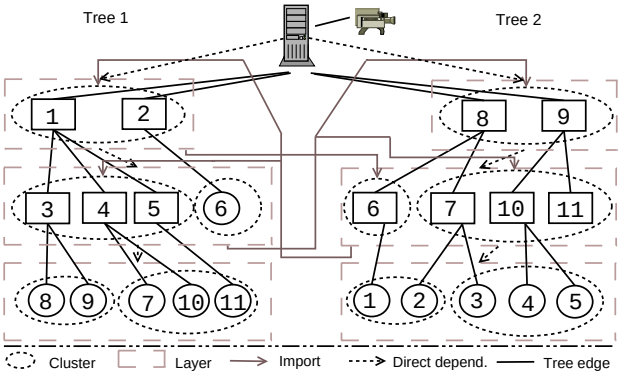

Figure 1: Overlay with $N = 11, t = 2, m = 2$ and $d = 1$ showing nodes, clusters, layers, direct dependencies and imports (see Section 3). The square indicates that the node is fertile in the tree.

how long it takes for a node to find a parent in a tree depending on its priority. We call this the reconnection time, and denote it by Ξ. The reconnection time consists of the time needed for the detection of the loss of the parent node, the time needed for searching for a new eligible parent node, and the time needed for connecting to the eligible parent. The expected value of Ξ can be up to tens of seconds depending on the tree management and the forwarding capacity in the tree [12].

2.3 Data transmission and error resilience

The root splits the data stream into t stripes, with every t^{th} packet belonging to the same stripe, and it sends the packets in round-robin to its children in the different trees. Peer nodes relay the packets upon reception to their respective child nodes. We consider two means of error resilience: retransmissions and FEC.

Retransmissions

Retransmission are widely used in streaming applications to decrease the packet loss, even though the delay constraints limit the number of retransmission attempts. Hence retransmissions cannot guarantee reliable data delivery. We assume that the maximum number of retransmission attempts is limited due to delay constraints, and we denote the limit by x.

FEC

The root uses block based FEC, e.g., Reed-Solomon codes [14], so that nodes can recover from packet losses due to network congestion and node departures. To every k packets of information c packets of redundant information are added resulting in a block length of $n = k + c$. We denote this FEC scheme by FEC(n,k). Lost packets can be reconstructed as long as no more than c packets are lost out of n packets. Once a node receives at least k packets of a block of n packets, it may recover the remaining c packets. If a packet, which should have been received in the tree where the node is fertile, is recovered, then it is sent to the respective children. Duplicate packets are discarded by the nodes. If the root would like to increase the ratio of redundancy while maintaining its bitrate unchanged, then it has to decrease the source rate. If $n \leq t$ then at most one packet of a block is distributed over the same distribution tree. Using this FEC scheme one can implement UXP, PET, or the MDC scheme considered in [4].

3. DATA DISTRIBUTION MODEL AND PERFORMANCE METRICS

The building blocks of the overlay are the individual nodes, so we start the description of the model by describing our model of a single node in Section 3.1. Using the notations introduced there we define the performance metrics we consider in Section 3.2. We define clusters of nodes in Section 3.3, and describe the model of the overlay in Section 3.4. We discuss the asymptotic behavior and the stability of the overlays in Section 3.5.

3.1 Node model

The input capacity of a node r is t^r, the number of trees the node can connect to. We denote the set of trees that node r can connect to by $\mathcal{H}^r$, $\mathcal{H}^r \subseteq \{1 \dots t\}$, $|\mathcal{H}^r| = t^r$. The number of cogs of the node in tree e is γ_e^r, its number of children is Γ_e^r.

We consider three sources of disturbances in the overlay. First, a node cannot receive data in a tree where it is not connected to a parent node. We denote the probability of being disconnected in tree e by $p_{e,p}^r$ ($e \in \mathcal{H}^r$). We assume that $p_{e,p}^r$ is independent of $p_{h,p}^r$ ($h \in \mathcal{H}^r \backslash \{e\}$). The independence assumption is reasonable if nodes do not have the same node as parent in different trees. We will show how to calculate $p_{e,p}^r$ in Section 5.1.

Second, a node might experience losses on its input link. We denote the probability that l out of j packets are lost on the input link of a node by $P_I^r(l,j)$. $P_I^r(l,j)$ can be calculated using loss models such as the Bernoulli model or the Gilbert model [15].

Third, a node might experience losses on its output link. We denote the probability that l out of j packets are lost on the output link of a node by $P_O^r(l,j)$. $P_O^r(l,j)$ can be calculated in a similar way as $P_I^r(l,j)$. Packets lost on the output link of a node cannot be received by the children of the node. We model these two loss processes separately because the correlations in the two loss processes have different effects on the performance of the overlay, as we will see later.

We incorporate retransmission in the model as a decrease of the loss probability between a node and its parents. To keep the number of parameters low, we assume that the loss probabilities between two nodes are symmetrical. Given the loss probability p on a path between two adjacent nodes, we estimate the probability of unsuccessful packet delivery after x retransmissions ($x \geq 0$) as

$$p_x = p^{x+1}(2-p)^x. \quad (1)$$

Eq. (1) is an optimistic estimate of p_x as it does not take into account the possible correlation between the loss of successive retransmissions and the effect of increased transmission rate.

3.2 Performance metrics

To measure the performance of the data distribution in the overlay we use the probability π that an arbitrary node receives or can reconstruct (i.e., possesses) an arbitrary packet. If we denote by the random variable R^r the number of packets possessed by node r in an arbitrary block of n packets, then π can be expressed as the average ratio of packets possessed in a block over all nodes, i.e., $\pi = \frac{1}{N} E[\sum_r R^r/n]$. Typically, multimedia applications require $\pi > 0.99$.

We do not consider the delay performance in this model. We assume that delay jitters can be compensated at the playout buffers of the nodes, and end-to-end delays are controlled by keeping the depth of the transmission trees low.

3.3 Simplifying assumptions

The data distribution model is based on three simplifying assumptions.

Var.	Definition
t,m	# of trees and root multiplicity respectively
n,k	FEC block length and number of data pkts respectively
$J(j)$	# of lost pkts in a block of j pkts, $P(J(j)=l) = P(l,j)$
$p_{e,p}^f$	Prob. that a node in cluster f is disconnected in tree e
$\mathcal{H}^f$	Set of trees that nodes in cluster f connect to, $\|\mathcal{H}^f\| = t^f$
$\mathcal{C}^i$	Set of clusters that forward data in layer i
Γ_e^f	Average # of children of nodes of cluster f in tree e
N^f	# of nodes in cluster f
$R_e(i)$	# of pkts successfully departing from nodes forwarding in layer i in tree e (not lost on output link)
$R_{e,a}^f$	# of pkts a node in cluster f can receive from its parent in tree e
$R_{e,r}^f$	# of pkts a node in cluster f receives from its parent in tree e
R_e^f	# of tree e pkts possessed by a node in cluster f in tree e
$R_{e,d}^f$	# of pkts that depart from a node in cluster f in tree e
$\pi(i)$	Packet possession probability of nodes fertile in layer i
π	Packet possession probability of an arbitrary node

Table 1: List of notations used in the model.

Decomposition: We decompose the overlay into t nearly independent trees [16]. Each tree can be modeled as a Bayesian network, since each tree is a directed acyclic graph. The vertices of the Bayesian network are the packet possession probabilities, and the vertices belonging to one Bayesian network depend on one vertex of the same network and of some vertices of the other networks. We call the dependency within the tree direct dependency. The dependencies of other trees are called imports. To solve the model, we provide initial guesses for the imports and use fixed point iteration.

Independent parents: The probability that the parent of a node in tree e possesses a packet is independent of that the parent of the same node in tree h possesses a packet of the same block. This assumption is not true if nodes have the same parent in different trees. One of the main goals of multiple tree based overlays is to maintain independent paths in the different trees, i.e., different parents in every tree, which supports our assumption.

Clusters of nodes: To decrease the number of vertices of the Bayesian networks, we use clusters of nodes instead of individual nodes as vertices. Nodes belonging to a cluster forward data in the same tree(s), have their parents in the same trees in the same layers, have the same input capacities, and experience the same input and output loss probabilities. Consequently, a layer of a tree possibly consists of several clusters corresponding to sets of nodes with different characteristics. Clustering can be thought of as a form of quantization: more clusters give more accurate results but increased computation time. As nodes belonging to a cluster might have parents in different clusters (within the same layer), we assume that a layer appears to be homogeneous to nodes in the next layer. The model can be used without this assumption, at the price of increased number of clusters.

Figure 1 shows the clusters, the layers, the direct dependencies and the imports of the model for an overlay with $t = 2$ and $N = 11$. Our simulations show that the model is accurate despite the simplifying assumptions.

3.4 The cluster model

Let us consider a cluster f, in which nodes join trees $h \in \mathcal{H}^f$, $\mathcal{H}^f \subseteq \{1 \dots t\}$, $|\mathcal{H}^f| = t_f \geq 1$, and the parents of the nodes in tree h are in layer i_h^f ($h \in \mathcal{H}^f$). The key to the overlay's performance

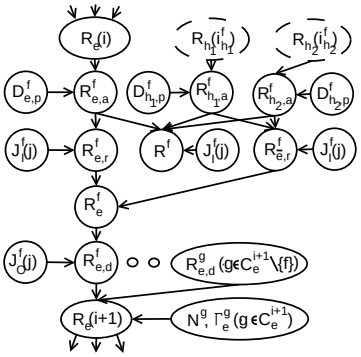

Figure 2: Bayesian network corresponding to the calculation of $R_e(i+1)$ from $R_e(i)$ through cluster $f \in \mathcal{C}_e^{i+1}$, $\mathcal{H}^f = \{e, h_1, h_2\}$. $R_{h_1}(i^f_{h_1})$ and $R_{h_2}(i^f_{h_2})$ are imports. Eqs. (2)-(9) give the relationships between the random variables.

is the probability that a node in cluster f possesses the packets in the trees where it has to forward data. Let us denote by $\mathcal{C}_e^i$ the set of clusters that forward data in tree e in layer i, and by the random variable R_e^f the number of packets possessed by a node in cluster f out of the n/t packets it should forward in tree e. In the following we show how the distribution of this random variable can be calculated. [1] We chose to give the relationship between the random variables instead of the stochastic vectors representing their distributions, as we believe that this formulation makes understanding easier. Figure 2 shows a graphical representation of the calculation of the random variables described in the following.

Let us denote by the random variable $R_e(i)$ the number of packets out of the n/t packets transmitted in tree e that successfully depart from an arbitrary node in layer i in tree e of the overlay, i.e. the packets that do not get lost on the output links of the nodes. A node in layer $i+1$ in tree e can only receive a packet from its parent if it is connected to one. Hence, given $R_e(i)$ we can express the random variable $R^f_{e,a}$, the number of packets that nodes in cluster $f \in \mathcal{C}_e^{i+1}$ (i.e., $i^f_e = i$) can receive from their parents in tree e. If we denote by $D^f_{e,p}$ a Bernoulli r.v. such that $P(D^f_{e,p} = 0) = p^f_{e,p}$, then

$$R^f_{e,a} = R_e(i^f_e) D^f_{e,p}. \tag{2}$$

Similarly, we can define the number of packets that can be received in other trees based on the imports $R_h(i^f_h)$, $h \in \mathcal{H}^f \backslash \{e\}$ and $D^f_{h,p}$. Eq. (2) is approximate if $n/t > 1$, because a parent can depart and a parent can be found during the transmission of a block. The number of packets actually received by a node depends on the loss probability on the input link of the node, so we define the random variable $R^f_{e,r}$ as the number of packets received by nodes of cluster f in tree e

$$R^f_{e,r} = R^f_{e,a} - J^f_I(R^f_{e,a}), \tag{3}$$

where $J^f_I(j)$ is the number of lost packets out of j packets on the input link, and it is a random variable with distribution $P(J^f_I(j) = l) = P^f_I(l, j)$. Similarly, we can calculate the total number of packets received in the other trees

$$R^f_{\bar{e},r} = \sum_{h \in \mathcal{H}^f \backslash \{e\}} R^f_{h,a} - J^f_I\Big(\sum_{h \in \mathcal{H}^f \backslash \{e\}} R^f_{h,a}\Big). \tag{4}$$

The relationship between the number of packets possessed in tree e, the number of packets received in tree e and the number of packets received in the other trees is

$$R^f_e = \begin{cases} n/t & \text{if} \quad R^f_{e,r} + R^f_{\bar{e},r} \geq k \\ R^f_{e,r} & \text{otherwise,} \end{cases} \tag{5}$$

due to the reconstruction of the lost packets using FEC. Now what remains is to show how $R_e(i+1)$ can be calculated. We express the random variable $R^f_{e,d}$, the number of packets out of n/t packets that do not get lost on the output link of a node of cluster f

$$R^f_{e,d} = R^f_e - J^f_O(R^f_e), \tag{6}$$

where $J^f_O(j)$ is the number of lost packets out of j packets on the output link, and is a random variable with distribution $P(J^f_O(j) = l) = P^f_O(l, j)$. Based on the $R^f_{e,d}$ for all $f \in \mathcal{C}^{i+1}$ we can express $R_e(i+1)$

$$R_e(i+1) = \frac{\sum_{f \in \mathcal{C}_e^{i+1}} R^f_{e,d} N^f \Gamma^f_e}{\sum_{f \in \mathcal{C}_e^{i+1}} N^f \Gamma^f_e}. \tag{7}$$

We start the calculation of the distributions of the above random variables by using the initial condition $P(R^{root}_e = n/t) = 1$, i.e., the root node possesses all data in all trees, and the imports $P(R_h(i)^{(0)} = 0) = 1$, $1 \leq h \leq t$. Then, in iteration l, we calculate the distribution of $R_e(i)^{(l)}$, ($1 \leq i < L$ and $1 \leq e \leq t$) using the imports from iteration $l-1$. The iteration stops when $|E[R_e(L-1)^{(l-1)}] - E[R_e(L-1)^{(l)}]| < \varepsilon$, where $\varepsilon > 0$. The iteration converges, since $E[R_e(i)^{(l)}]$ is monotonically increasing in l and $E[R_e(i)^{(l)}] \leq n/t$.

Based on the final value of $R_e(i_e)^{(l)}$, we can express the random variable R^f_r, the number of packets out of n that a node belonging to cluster f receives

$$R^f_r = \sum_{h \in \mathcal{H}^f} R^f_{h,a} - J^f_I\Big(\sum_{h \in \mathcal{H}^f} R^f_{h,a}\Big). \tag{8}$$

Finally, we define the packet possession probability π^f, as the ratio of packets in a block that a node belonging to cluster f possesses

$$\pi^f = \frac{1}{n} R^f = \frac{1}{n} E[R^f_r + \tau(R^f_r)], \tag{9}$$

where $\tau(l)$ is the number of reconstructed packets

$$\tau(l) = \begin{cases} 0 & 0 \leq l < k \\ n - l & k \leq l \leq n. \end{cases}$$

Finally, we define the packet possession probability of nodes that forward data in layer i as the weighted average of the π^f for $f \in \mathcal{C}^i$

$$\pi(i) = \frac{\sum_{f \in \mathcal{C}^i} \pi^f N^f}{\sum_{f \in \mathcal{C}^i} N^f}, \tag{10}$$

and the packet possession probability of an arbitrary node in the overlay as the weighted average of the π^f

$$\pi = \frac{\sum_f \pi^f N^f}{\sum_f N^f}. \tag{11}$$

[1] The iterative solution we outline is in fact the application of the belief propagation algorithm to a loopy Bayesian network partitioned into t trees. The marginals are the distributions of the random variables R^f_e [17].

3.5 Overlay stability

In the following we analyze the stability of a class of overlays. We observe that in all overlays proposed in the literature, nodes should be at least as close to the root in their fertile trees as they are in their sterile trees. We consider the case $n = t$, so that the random variables R_e^f are binary. We consider overlays consisting of homogeneous nodes in terms of loss probability and input capacity. We restrict ourselves to the case when nodes can receive data in every tree, thus $t^f = t$. We consider overlays with inhomogeneous incoming capacities in Section 4.10. A consequence of this assumption is that all trees are statistically identical, i.e., the $R_e(i)$, $1 \le e \le t$ are equal in distribution. We assume independent packet losses, so that losses due to node departures, on the input links and on the output links can be treated together as independent losses on the input links. If we denote the loss probability on the path between two nodes by p, then the number of lost packets in a block follows the binomial distribution

$$P(l, j) = \left(\begin{array}{c} j \\ l \end{array} \right) p^l (1-p)^{j-l}. \quad (12)$$

Overlays that fulfill the above conditions of loss independence and homogeneous capacities are not likely to be found in practice, but the results derived here give important insight into the behavior of heterogeneous overlays as we will show it in Section 4.

3.5.1 Upper bound of the packet possession probability

Using the above simplifying assumptions, from (2)-(7) and the initial condition $E[R_e^{root}] = n/t$ $(1 \le e \le t)$ it follows that $E[R_e(i)]$ is a non-increasing function of i. Hence, we can give an upper bound on $E[R^f] = P(R^f = 1)$ (R^f is a binary r.v. because $t = n$) by assuming that the parents of the nodes forwarding in a tree in layer i are in layer $i = min_{h \in \mathcal{H}} i_h^f$ in all trees. Let us denote the upper bound of the packet possession probability in layer i by $\overline{\pi}(i)$, then

$$\overline{\pi}(i+1) = \overline{\pi}(i)(1-p) + (1 - \overline{\pi}(i)(1-p)) \sum_{j=k}^{n-1} \left(\begin{array}{c} t-1 \\ j \end{array} \right) \overline{\pi}(i)^j (1-\overline{\pi}(i))^{t-1-j} \sum_{l=0}^{j-k} P(l, j). \quad (13)$$

The $\overline{\pi}(i)$ can be calculated using the initial condition $\pi(0) = 1$, and the upper bound of the packet possession probability for an overlay with L layers and $N(i)$ nodes in layer i is

$$\overline{\pi} = \frac{\sum_{i=1}^{L} \overline{\pi}(i) N(i)}{N}. \quad (14)$$

3.5.1.1 Asymptotic behavior.

Eq. (13) defines a non-linear recurrence relation for $\overline{\pi}(i)$. $\overline{\pi}(i)$ is a monotonically non-increasing function of i, $\overline{\pi}(0) = 1$ and $\overline{\pi}(i) \ge 0$, so that $lim_{i \to \infty} \overline{\pi}(i) = \overline{\pi}(\infty) \ge 0$ exists. $\overline{\pi}(\infty)$ is equal to the asymptotically stable fixed point of (13) closest to 1 if such a fixed point exists and is 0 otherwise. We see by substitution that (13) has a fixed point at $\pi = 0$ for any distribution of $P(l, j)$. In the following we are interested in the fixed points of (13) on $(0, 1]$. If there is at least one asymptotically stable fixed point on $(0, 1]$ then $\overline{\pi}(i)$ converges to that fixed point, and we say that the overlay is stable. Otherwise, $\overline{\pi}(i)$ converges to 0, and the overlay is unstable.

THEOREM 1. *For the i.i.d Bernoulli loss model the number of fixed points of (13) is* 0,1 *or* 2. *For $k = 1$ a fixed point exists iff $p < (n-1)/n$. For $k > 1$ the number of fixed points is* 0 *if $p > (n-k+1)/n$. If there are* 2 *fixed points r_1 and r_2 ($r_1 < r_2$) then r_2 is asymptotically stable and r_1 is unstable.*

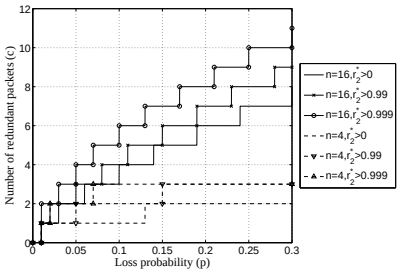

Figure 3: *c* **vs** *p* **for various objectives for the stable fixed point.**

The proof of the theorem can be found in the Appendix. A consequence of the proof is that for any p and $\varepsilon > 0$ there is an n, k pair for which r_2 exists and $r_2 > 1 - \varepsilon$. Fig. 3 shows the number of redundant packets needed in a block of packets in order to achieve various objectives for the stable fixed point r_2 as a function of the loss probability p.

3.5.2 Lower bound of the packet possession probability

We get the lower bound of the packet possession probability by assuming that the parents of a node of cluster f are in layer $i = max_{h \in \mathcal{H}^f} i_h^f$ in all trees. Let us denote the lower bound of the packet possession probability in layer i by $\underline{\pi}(i)$. If there is no FEC reconstruction, then $\underline{\pi}(i) = (1-p)^L$. Using FEC in an overlay with L layers, if $(1-p)^L > r_1$ then after successive iterations of the model $\underline{\pi}(i) = \underline{\pi}(L) = r_2$, the stable fixed point of (13). Consequently, $(1-p)^L > r_1$ is a sufficient condition for the overlay to be stable.

4. PERFORMANCE EVALUATION: PACKET LOSSES

We start the evaluation by considering the simplest case, homogeneous nodes with independent packet losses. When considering heterogeneous systems, we follow the "ceteris paribus" principle, i.e., we change one property at a time and keep all other properties equal. Doing so allows us to understand and explain the effects of different types of heterogeneity. Most figures we show are composed of two sub-figures. The one on the left shows the behavior of the overlay for a large interval of the input parameter. The one on the right is zoomed on values of π of practical interest and can show both modeling and simulation results.

4.1 Simulation methodology

We developed a packet level event-driven simulator to validate our models. We used the GT-ITM topology generator [18] to generate a transit-stub network with 10^4 nodes and average node degree 6.2. We placed each node of the overlay at random at one of the 10^4 nodes of the topology and used the one way delays given by the generator between the nodes. The delay between overlay nodes residing on the same node of the topology was set to 1 ms. We assume that the interarrival times of nodes are exponentially distributed, this assumption is supported by several measurement studies [19, 20]. We consider two distributions for the session holding times M: the log-normal distribution [19] with CDF $F_M(x) = 0.5 + 0.5 erf((ln(x) - a)/(b\sqrt{2}))$, $a = 4.93$, $b = 1.26$; and the shifted Pareto distribution [20] with CDF $F_M(x) = 1 - (1 + x/b)^{-a}$, $b = 612$, $a = 3$. In both cases the mean lifetime is $E[M] = 306s$ [19].

Tree maintenance: We assume that a distributed algorithm, such as gossip based algorithms, is used by the nodes to learn about other nodes. We do not simulate the information dissemination, but assume that it provides random knowledge of the overlay such

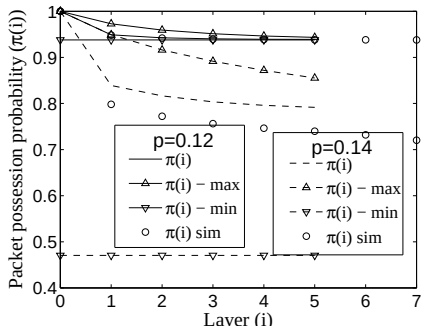

Figure 4: $\pi(i)$ **vs i.** $\overline{N} = 50000$**,** $n = t = 4$**,** $k = 3$**,** $m = 50$**, homogeneous case.**

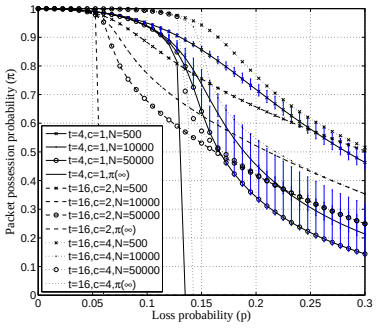

Figure 5: π **vs** i **for** $n = t$**,** $m = 50$**, homogeneous case.**

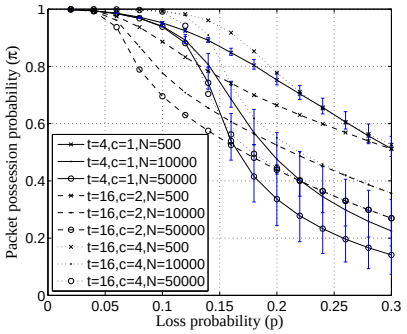

Figure 6: π **vs** i **for** $n = t$**,** $m = 50$**, homogeneous case. Simulation results.**

as in [11]. Since our focus is not on the structure of the resulting overlays, this assumption does not influence our conclusions.

When a node wants to join the overlay, it contacts the root and obtains a random list of $g = 100$ members of every tree. The root tells to the arriving node in which trees it should forward data: in the ones with the least amount of forwarding capacity. The arriving node then uses the following parent selection procedure to find a parent.

To select a parent in a tree, the node sorts the g members it is aware of into increasing order according to their distances from the root, and looks for the first node that has available capacity or has a child that can be preempted, i.e., which has lower priority. We describe the considered priority schemes below. If the node has to preempt a child, but itself has available capacity, then the preempted child can immediately become a child of the preempting node. Otherwise, the preempted child has to follow the parent selection procedure just like the child nodes of a departed node.

We specify the distributions used to simulate the reconnection time (Ξ) in Section 5.2. As opposed to [11, 13], we do not force all nodes in the subtree of a departed node to reconnect individually. We believe that forcing all nodes in a subtree to disconnect in a large overlay creates large control overhead and can lead to scalability issues.

Node priority: We consider two node preemption strategies. For simplicity we represent a node's priority as an unsigned 32 bit integer b consisting of 4 bytes b_0 (MSB) to b_3 (LSB). Higher b means higher priority. In the following we specify how these bytes are set to reflect the priority of a node, which can depend both on the tree and on the layer where it looks for a parent.

In the non-prioritized preemption strategy the only preemption is when fertile nodes preempt sterile nodes. This is necessary to push fertile nodes close to the root and sterile nodes to the last layers of the trees. b_1 is 1 if the node forwards data in the *tree* and it is 0 otherwise. We will refer to this strategy by NP.

The second preemption strategy is specific to some performance measure, such as the packet reception probability, the number of cogs of a node or the input capacity of the node. We set b_0 proportional to the performance measure of the node in the *tree*, b_1 is the forwarding capacity of the node in the *tree*, b_2 is proportional to the performance measure of the node in the *overlay*, and b_3 is the *total* forwarding capacity of the node. We will refer to this strategy by P. For example, if we want to prioritize nodes according to the packet loss probabilities they experience, we set b_0 to $\lceil 255(1-p) \rceil$.

Data distribution: We consider the streaming of a 112.8 kbps data stream. The particular choice of the bitrate does not affect the validity of our conclusions, as we express the links' capacity relative to the bitrate. The packet size is 1410 bytes. Nodes have a playout buffer capable of holding 140 packets, which corresponds to 14 s delay with the given parameters. Every node has an input and an output buffer of 80 packets each to absorb the bursts of incoming and outgoing packets. Apart from packet losses due to the overflow of the input and output buffers and due to late arriving packets, we simulate packet losses on the input and the output links of the nodes via two-state Markovian models, often referred to as the Gilbert model [21]. For given stationary loss probability p and conditional loss probability (the probability that a packet is lost given that the previous packet was lost) $p_{\omega|\omega}$ we set the parameters of the model as described in [22].

To obtain the results for a given overlay size $\overline{N}$, we start the simulation with $\overline{N}$ nodes in its steady state as described in [23]. We set $\lambda = \overline{N}/E[M]$ and let nodes join and leave the overlay for 5000 s. The purpose of this warm-up period is to introduce randomness into the trees' structure. The measurements are made after the warm-up period for 1000 s and the presented results are the averages of 10 simulation runs. The results have less than 5 percent margin of error at a 95 percent level of confidence.

4.2 Approximating the overlay's structure

Given m, t, d and the nodes' parameters in a layer, one can calculate the number of nodes and the number of clusters per layer. Without prioritization, we assume that nodes with different parameters are distributed uniformly in the layers. With prioritization, we assume that prioritized nodes are as close as possible to the root. There is a difference between the calculated and the real overlay structure due to node churn and the distributed tree maintenance, but the simulations show that the effects of these differences are negligible.

4.3 The minimum depth overlay

We start the evaluation with the minimum depth overlay as this is the one most commonly used in the literature [3, 4, 11, 6, 13, 12]. We start the evaluation with a homogeneous overlay, and in the following subsections we show how heterogeneity influences the overlay's performance. To keep the number of clusters low, when calculating the trees' structure, we assume that a node is sterile in the same layer in all trees, i.e., the penultimate or the last layer. Thus the fertile nodes in a layer of the tree belong to one of two clusters depending on the layer where they are sterile. To consider independent, homogeneous losses on the overlay links, $P_I^f(l, j)$ follows a binomial distribution with parameters j, p, and $P_O^f(0, j) = 1$ for all clusters. For all nodes $t^r = t$ and $\gamma^r = t$.

Figure 4 shows the packet possession probability as a function of the layer where nodes are fertile for two loss probabilities $p = 0.1$ and $p = 0.14$. Fertile nodes occupy more layers in the simulation

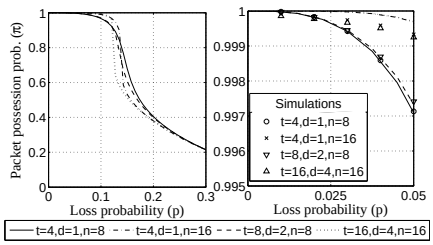

Figure 7: π **vs** p **for** $d > 1$ **and** $n > t$**,** $m = 50$**,** $\overline{N} = 10^4$**.**

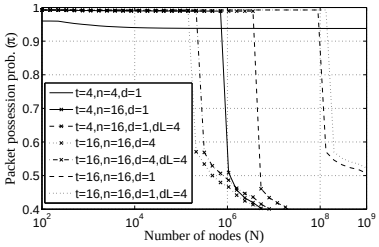

Figure 8: π **vs number of nodes at** $p = 0.10$**,** $k/n = 0.75$**,** $m = 50$**.**

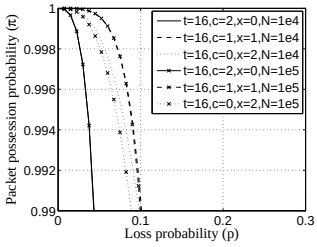

Figure 9: π **vs loss probability for FEC and retransmissions for** $t = n = 16$**.**

than they would in a well maintained tree considered for the model. The stability threshold is $p_{max} = 0.129$, i.e., for $p = 0.14$ the overlay is unstable. The upper bound of the packet possession probability given by (13) is tight in the stable state only: in the unstable state the poor reception in the last layer impacts the performance of the uppermost layer. The lower bound given in Section 3.5.2 is far below in the unstable state, which shows that FEC reconstruction improves π in the unstable state as well.

Figure 5 plots π as a function of the loss probability. Figure 6 shows simulation results for the same scenarios. The simulations verify that the decomposition approach gives accurate results even for small overlays.

The overlays are unstable where $\pi(\infty) = 0$ for the corresponding FEC parameters and number of trees. In the unstable state π drops suddenly. The drop is faster for larger overlays, hence good results obtained with a small overlay do not necessarily hold as the number of nodes increases. The results are however independent of the overlay's size in the stable state. Comparing results for different redundancy rates (c/n) shows that a higher redundancy rate results in a wider region of stability and higher values of π.

Increasing the FEC block length, in general, improves the performance of FEC. Figure 5 shows that π can be increased at a given redundancy rate by increasing the number of trees t and the block length n. Figure 7 shows that increasing n can improve π without having to increase the number of trees, as long as the overlay is stable and losses are not correlated.

4.4 Increasing the number of fertile trees

Increasing the number of trees decreases the depth of the overlay and as we have seen improves the FEC performance. At the same time it can increase the time it takes to find a parent, unless one increases the number of trees where a node can forward data [24]. Figure 7 shows π as a function of p for cases when $d > 1$. To decrease the number of clusters, we assume for the model that a node is fertile in the same layer in all trees. The simulation results in the figure show that this approximation is accurate. As shown in the figure, for the considered independent losses increasing d decreases the stability region. Consequently, to improve FEC performance it looks more favorable to increase n without increasing t and d. We will see that under node churn the contrary is true in Section 5.2.

The minimum breadth overlay. The minimum breadth overlay, in which nodes forward data in all trees, is the $d = t$ special case of $d > 1$ and has been studied earlier in the literature. The number of layers and the average number of hops between the root and the peer nodes in this overlay is $O(N)$, so that nodes have to remain in almost the same layer in all trees to avoid large delays between the data arriving in different trees. If they do so, the packet possession probability of nodes in layer i of the overlay approaches the upper bound given in (13). A detailed analysis of this overlay was presented in [22].

4.5 Overlay size

Figure 8 shows the dependence of π on the number of nodes in the overlay. For $n = 4$ the overlay is stable in the whole considered interval, for $n = 16$ it is however not. We can conclude that a stable overlay can become unstable for two reasons: increased packet losses or increased number of layers. We would like to remind the reader, that it is not the number of nodes that causes the degradation, but the number of layers needed to accommodate them. Consequently, an overlay can become unstable for lower values of N depending on the tree maintenance algorithm used.

4.6 Limiting the layer spread

Our model reveals a significant deficiency of the minimum depth overlay. The depth of the overlay influences the probability of reconstruction even in nodes close to the root in their fertile tree, since reconstruction requires packet reception in the sterile trees, where nodes are located in the last layers. Motivated by this deficiency, we propose a tree stucture in which the spread between the layers in the different trees is limited by dL. That is, if a node is fertile in layer i in a tree then it is located no deeper than layer $i + dL$ in its sterile trees. We do not discuss here how to implement this scheme, our goal is to show its possible benefits if it can be implemented. Limiting the layer spread can increase the number of layers in the overlay, but it makes FEC reconstruction more efficient. Figure 8 shows that limiting the layer spread does not decrease the performance of a stable overlay, but, as expected, the overlays with limited layer spread remain stable for larger values of N.

4.7 Retransmissions vs. FEC

It is difficult to make a fair comparison between FEC and retransmissions, as the overhead introduced by retransmissions depends on the loss probability, while the overhead of FEC is independent of it. Fig. 9 shows π as a function of the loss probability with different combinations of FEC redundancy and maximum number of retransmissions, denoted by x. We do not model the effect of increased transmission rate, the latency introduced by retransmissions, and the resulting late arrivals, hence the results shown are upper estimates of the performance. We show results for two overlay sizes. Increasing the number of nodes does not affect the performance if both FEC and retransmission is used (the corresponding curves run together), but decreases the performance if only retransmission is applied. The most efficient solution for the considered scenarios is the combined use of FEC and retransmissions. Note, however, that if losses are due to node departures, retransmissions work only if nodes can request retransmissions from backup parent nodes.

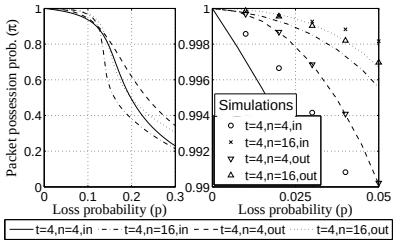

Figure 10: π **vs** p **for** $t = 4$,$k/n = 0.75$, $m = 50$, $\overline{N} = 10^4$, $p_{\omega|\omega} = 0.3$ **correlated losses on the input or on the output links.**

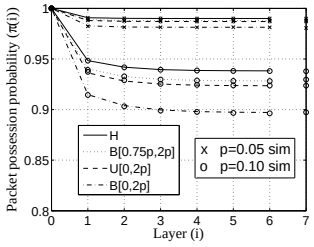

Figure 11: $\pi(i)$ **vs** i **for inhomogeneous losses.** $\overline{N} = 10^4$, $m = 50$, $t = n = 4$, $k = 3$, $d = 1$. **Model and simulations.**

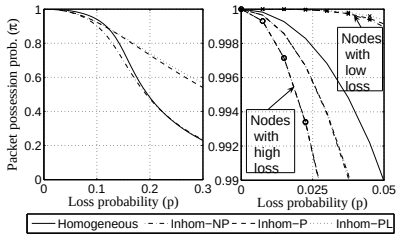

Figure 12: π **vs packet loss probability for inhomogeneous losses and prioritization.** $\overline{N} = 10^4$, $m = 50$, $t = n = 4$, $k = 3$, $d = 1$.

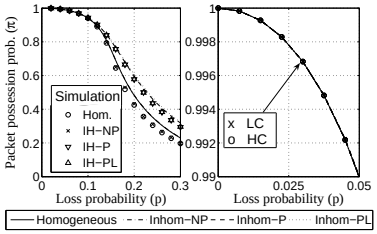

Figure 13: π **vs** i **for inhomogeneous output capacities.** $\overline{N} = 10^4$,$m = 50$,$t = n = 4$,$k = 3$

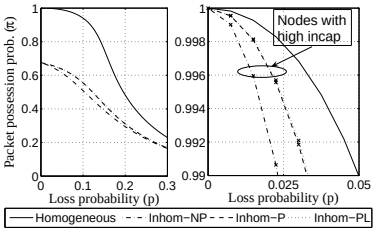

Figure 14: π **vs** i **for inhomogeneous input capacities.** $\overline{N} = 10^4$,$m = 50$,$t = n = 4$,$k = 3$

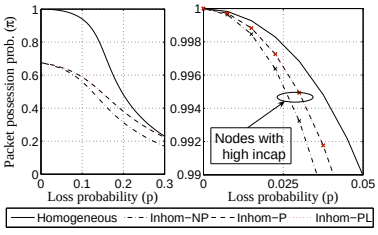

Figure 15: π **vs** i **for inhomogeneous input and output capacities.** $\overline{N} = 10^4$,$m = 50$,$t = n = 4$,$k = 3$

4.8 Correlated losses

Fig 10 plots π for correlated losses on the input links or on the output links of the nodes. We show results for a conditional loss probability of $p_{\omega|\omega} = 0.3$. Correlations on the output links of the nodes have no effect on the performance if $n = t$, since the consecutive packets will be received by different child nodes. Correlations on the input links decrease the performance compared to the case of independent losses. A longer FEC block ($n > t$) increases the packet possession probability for both kinds of correlations when the overlay is stable. Based on the model we know that for correlated on the output links and for $n > t$ the performance approaches that of $n = t$ as $p_{\omega|\omega}$ increases. Correlated losses affect the overlay's performance mostly at low loss probabilities as correlations decrease the mean number of reconstructed packets. Consequently, correlations decrease the stability region of the system. The simulations shown in the zoomed box show a good match with the model for correlated losses on the output links. There is a mismatch in the case of correlations on the input links, as packets of the same block do not necessarily arrive successively in the simulation, hence the loss correlation between packets in a block in the simulation is lower than $p_{\omega|\omega}$.

4.9 Inhomogeneous losses

Figure 11 compares the performance of an overlay with $\overline{N} = 10^4$ for four distributions of the loss probability experienced by nodes and with the Bernoulli loss model. We use homogeneous (H) losses with probability p as the reference, and compare that to the following scenarios: 80 percent of the nodes experience $0.75p$ while the rest $2p$; uniform distribution on $[0, 2p]$; 50 percent of the nodes experience 0 while the rest $2p$. We used 100 clusters per layer to approximate the uniform distribution in the model. Both the model and the simulations show that $\pi(i)$ decreases as the variance of the losses increases.

To see whether prioritization could help to alleviate the negative effects of loss inhomogeneity, Fig. 12 compares the average packet possession probability in the overlay for four cases: homogeneous losses, for inhomogeneous losses without any priority scheme (Inhom-NP), for inhomogeneous losses prioritizing nodes with low packet loss probability (Inhom-P) and for inhomogeneous losses and prioritization, also limiting the layer spread (Inhom-PL) with $dL = 2$. We consider $t = 4$, and $\overline{N} = 10^4$ of which 50 percent experience $2p$ and 50 percent experience no losses. Prioritizing nodes based on the packet losses they experience can be difficult in practice, but it is still interesting if one could improve the system by such a scheme at all. Surprisingly, prioritization does not improve π in the stable region of the system. Nevertheless, nodes with no losses experience better performance thanks to prioritization, limiting the layer spread giving slightly larger gain. In the unstable region, prioritization pays off as the decrease of π becomes much slower.

4.10 Inhomogeneous capacities

We start by showing the effects of inhomogeneous output capacities. We consider prioritization based on the output capacities of the nodes. A practical alternative would be to consider the number of children of a node [12], as that is easier to estimate, but it would not help high contributor nodes joining the overlay for the first time.

Fig 13 considers an overlay with $t = 4$, and $\overline{N} = 10^4$, of which 65 percent are low contributors with $\gamma^r = 2$ and 35 percent are high contributors with $\gamma^r = 8$. This ratio of high and low contributors is similar to that considered in [12] based on a measured trace. The figure shows a scenario with homogeneous output capacities as reference, the inhomogeneous case without priority, with priority, and also limiting the layer spread with $dL = 2$. Prioritization does not make any difference for a stable overlay, as the number of layers does not influence the performance of the overlay in the stable region. High and low contributors experience the same performance too. We note that as the number of layers decreases due to prior-

itization based on the output capacities, the stability region might increase. For the same reason, prioritization gives superior performance in the unstable state of the overlay. The simulations show a good match with the model, though for high losses the model somewhat overestimates π which is due to the difference between the number of layers in the simulation and the one we calculated with.

Next, we consider inhomogeneous input capacities for $t = 4$ and $\overline{N} = 10^4$ in Fig. 14. 65 percent of the nodes have $t^r = 2$ and the rest $t^r = 4$. Prioritization is based on the input capacities of the nodes. Prioritization does not improve the performance of the overlay in the stable state, though it proves to be beneficial in the unstable regime. Nevertheless, using prioritization, nodes with high input capacity experience significantly better performance.

As a next step, we combine the previous two scenarios in Fig. 15. The input capacity of the low contributors is $t^r = 2$, and that of the high contributors is $t^r = 4$. The results show that the effects of prioritization are similar to those in Fig. 14, i.e., prioritization can give incentives to high contributors but does not improve the overall performance in the stable state. Limiting the layer spread slightly improves the performance seen by high contributors as expected.

5. MODELING NODE CHURN

In the following section we calculate the probability that a node in cluster f does not have a parent in tree e, i.e., parameter $p^f_{e,p}$ of the data distribution model in Section 3. We first develop a general model of the ratio of disconnected parents of a node, then we show how to use it to model the effects of node departures and preemptive parent selection schemes.

5.1 Random observer model

The probability that a node is disconnected in a tree is influenced by how often it loses its parent in the tree, and for how long it has to look for a new one. These two measures are influenced by the priority of the node in the tree. Consequently, we consider a set of trees where nodes of cluster f have the same priority $\mathcal{H}^f_b$, $|\mathcal{H}^f_b| = t_b$ (e.g., for the *NP* scheme and $d = 1$ there are two sets, the trees where the nodes are sterile $\mathcal{H}^f_S$, and the tree where they are fertile $\mathcal{H}^f_F$).

For the model we assume that the distribution of the nodes' lifetimes M is exponential with parameter μ, $E[M] = 1/\mu$. Let us denote the inter-disconnection time of the nodes in the cluster by Ω_b and model it with an exponential distributed r.v. with mean $E[\Omega_b] = 1/\omega_b$. Without preemptions and if preemptions are graceful Ω_b and M are equal in distribution due to the exponential assumption. If preemptions are ungraceful, then the disconnection intensity ω_b of a node is the sum of the preemption intensity and the death intensity of the parents of the node. Let us assume that the reconnection times Ξ_b in the considered trees fit an exponential distribution with parameter ξ_b, i.e., $E[\Xi_b] = 1/\xi_b$. We will evaluate the accuracy of the exponential modeling assumptions in the following section.

The probability $p^f_{e,p}$ can be expressed as the average ratio of disconnected parents in trees $e \in \mathcal{H}^f_b$ of a node of cluster f *as seen by a random observer*. Without loss of generality we can denote the arrival time of the observer by 0.

We model the evolution of the number of disconnected parents with a continuous time discrete state space Markov process $X(h) \in S$, $S = [0 \dots t_b]$. The ratio of disconnected parents is $r_i = i/t_b$ in state i ($0 \le i \le t_b$). The transition intensities of the Markov process are

$$q_{i,i+1} = (t_b - i)\omega_b \qquad 0 \le i \le t_b - 1 \quad (15)$$

$$q_{i,i-1} = i\xi_b \qquad 1 \le i \le t_b. \quad (16)$$

The above model is an Engset system [25], and we are interested in the probability $P(X(0) = j|\mathbf{u})$ that a random observer finds an arbitrary node in state j, given that the node was started with initial state distribution $\mathbf{u} = \{u_0, ..., u_{t_b}\}$. Let us denote by h the age of the node when the random observer arrives and by $A(h)$ its distribution function, then

$$P(X(0) = j|\mathbf{u}) = \int_0^\infty \sum_{i=0}^{t_b} u_i p_{i,j}(h) dA(h). \quad (17)$$

$p_{i,j}(h)$ is given by $p_{i,j}(h) = P(X(0) = j|X(-h) = i) = e^{\mathbf{Q}h}_{\{i,j\}}$, where $\mathbf{Q}$ is the intensity matrix $\mathbf{Q} = \{q_{i,j}\}$. We use zero-based indexing for the rows and columns of the matrices. The age of an arbitrary node as seen by a random observer is the backward recurrence time of a renewal process with exponentially distributed inter-renewal times. Hence, the distribution of h is exponential with parameter μ. Based on $P(X(0) = j|\mathbf{u})$ we calculate the mean of the ratio of disconnected parents of the node as

$$E[\Delta_b|\mathbf{u}] = \sum_{j=0}^{t_b} \frac{j}{t_b} P(X(0) = j|\mathbf{u}), \quad (18)$$

In the following we give a closed form expression for initial state distribution $\mathbf{u}_i, u_k = \delta_i(k)$.

THEOREM 2. *For initial state distribution $\mathbf{u}_i$ the mean of the ratio of disconnected parents is*

$$E[\Delta_b|\mathbf{u}_i] = \frac{t_b + i\alpha_b}{t_b(\kappa_b + \alpha_b + 1)}, \quad (19)$$

where $\kappa_b = \xi_b/\omega_b$ and $\alpha_b = \mu_b/\omega_b$.

PROOF. Let us substitute (17) into (18)

$$E[\Delta|\mathbf{u}_i] = \sum_{j=0}^{t_b} \frac{j}{t_b} \left\{ \int_0^\infty p_{i,j}(h) \mu e^{-\mu h} dh \right\} \quad (20)$$

$$= \int_0^\infty \left\{ \sum_{j=0}^{t_b} \frac{j}{t_b} p_{i,j}(h) \right\} \mu e^{-\mu h} dh. \quad (21)$$

$E[\Delta|\mathbf{u}_i]$ is determined by the elements $\{p_{i,j}(h)\}$ of the i^{th} row of the matrix $P(h) = e^{-\mathbf{Q}h}$, which can be given in closed form due to the special structure of the $\mathbf{Q}$ matrix. The number of disconnected parents is governed by the differential-difference equations

$$\begin{aligned} p'_{i,0}(h) &= -t_b\omega_b p_{i,0}(h) + \xi_b p_{i,1}(t) \\ p'_{i,j}(h) &= -((t_b - j)\omega_b + j\xi_b) p_{i,j}(h) + \\ &\quad (t_b - j)\omega_b p_{i,j-1}(h) + (j+1)\xi_b p_{i,j+1}(h) \\ p'_{i,t_b}(h) &= -t_b\xi_b p_{i,t_b}(h) + \omega_b p_{i,t_b-1}(h). \end{aligned}$$

The generating function of the probabilities $\{p_{i,j}(h)\}$ is

$$P_i(z,h) = \sum_{j=0}^{t_b} p_{i,j}(h) z^j = \frac{1}{(1+\kappa)^{t_b}} (B + Az)^{t_b - i} (D + Cz)^i, \quad (22)$$

where $A = 1 - M(h)$, $B = M(h) + \kappa_b$, $C = \kappa_b M(h) + 1$, $D = \kappa_b(1 - M(h))$, and $M(h) = e^{-\omega_b(1+\kappa_b)h}$. For $\mathbf{u}_{t_b}$ and $\mathbf{u}_0$ evaluating (22) leads to the well known product form solution [25], but we are not

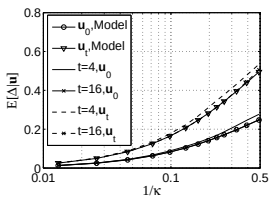

Figure 16: $E[\Delta]$ **vs** $1/\kappa$ **for log-normal lifetime and deterministic reconnection time distribution.**

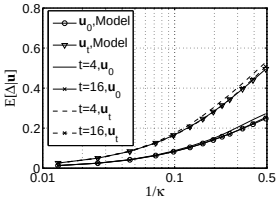

Figure 17: $E[\Delta]$ **vs** $1/\kappa$ **for Pareto lifetime and normal reconnection time distribution.**

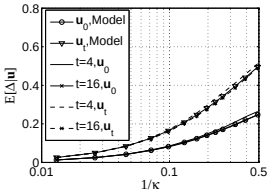

Figure 18: $E[\Delta]$ **vs** $1/\kappa$ **for Pareto lifetime and uniform reconnection time distribution.**

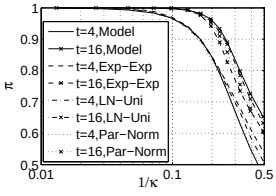

Figure 19: π **vs** $1/\kappa$ **for** $\overline{N} = 10^4$**,** $n = t$**,** $k/n = 0.75$**,** $m = 50$**,** $\mathbf{u}_0$**, the model and various lifetime distribution–reconnection time distribution pairs.**

aware of any results for the general case described here. Let us substitute (22) into the sum in (21)

$$\sum_{j=0}^{t_b} \frac{j}{t_b} p_{i,j}(h) \quad = \quad \frac{(t_b - i)(1 - M(h)) + i(\kappa_b M(h) + 1)}{t_b(1 + \kappa_b)}. \quad (23)$$

We substitute (23) into (21) and get

$$E[\Delta_b | \mathbf{u}_i] \quad = \quad \frac{t_b + i\alpha_b}{t_b(\kappa_b + \alpha_b + 1)}. \qquad \square$$

For $\alpha \to \infty$ (19) reduces to the initial state i, while for $\alpha \to 0$ it converges to the steady state solution of the mean number of jobs in an Engset system [25]. Based on (19) one can calculate the mean number of the children of a node as well, if one substitutes ω by the arrival rate of the children as seen by the node, and ξ by the departure rate of the children of the node.

5.2 Performance evaluation

We start by evaluating the sensitivity of the mean ratio of disconnected parents, $E[\Delta]$ to the node lifetime and the reconnection time distributions. We consider the scenario $E[\Xi_F] = E[\Xi_S]$, which means, the reconnection times are the same in the sterile and the fertile trees, and homogeneous input and output capacities. The scenario is not realistic, but its simplicity allows us to focus on the sensitivity of the results to the distributions. We simulated two node lifetime and three reconnection time distributions, and for each combination we considered two scenarios, corresponding to $\mathbf{u}_0$ and $\mathbf{u}_t$ with graceful preemptions ($\alpha = 1$). We set $\overline{N} = 10^4$, $m = 50$. Figs. 16-18 show that the exponential approximation is accurate, and gives a lower bound for other distributions. Using a heavy-tailed distribution the proportion of short lived nodes is high, but they have fewer children upon their departure, hence their impact is lower on $E[\Delta]$.

Next we apply the data distribution model to calculate π in the presence of node churn: for given κ we set $p^f_{e,p} = E[\Delta]$. The simulation results shown for $\mathbf{u}_0$ for the data distribution performance show a similarly good match in Fig. 19.

Increasing the block length: For packet losses due to network failures increasing the block length without increasing the number of trees does improve the performance in a stable overlay. Fig. 20 shows that in the case of node departures this is not necessarily true. For $t = 4$, $n = 16$ the performance is equal to that of $t = 4$, $n = 8$, and in fact is equal to that of $t = n = 4$. Increased block length gives however increased performance if the number of trees and the number of fertile trees increase as well, as shown in the figure for $d > 1$. The simulations were performed using the Pareto lifetime and normal reconnection time distributions and show that the approximation for $n > t$ is accurate.

Why does preemption improve the performance?

We showed in Section 4 that not even the ideal preemption strategies can significantly improve the average performance of an overlay in its stable state in the case of packet losses. Nevertheless, simulation and measurement studies [6, 12] show that preemption does improve the overlay's stability. The two are not contradictory.

Fig. 21 shows π as a function of the ratio of the mean reconnection times of nodes in their fertile trees ($E[\Xi_F]$) and in their sterile trees ($E[\Xi_S]$). For given $E[\Xi]$ we set $E[\Xi_F] + (t-1)E[\Xi_S] = E[\Xi]$ and consider two cases. The best case, graceful preemptions ($E[\Omega_S] = E[M], \alpha = 1$), and the worst case, non-graceful preemptions occurring after the departure of every fertile node ($E[\Omega_S] = (t-1)/tE[M], \alpha = 0$). The performance significantly improves as $E[\Xi_S]/E[\Xi_F]$ increases in both scenarios with a decreasing marginal gain, i.e., any preemption scheme that decreases $E[\Xi_F]$ without increasing $E[\Xi]$ is beneficial.

Finally we look at the effects of taxation and contribution aware parent allocation [12] in Fig. 22. We consider an overlay with $t = n = 8$, $k = 6$, and $\overline{N} = 10^4$. 75% of the nodes are low contributors (LC) with 4 cogs and the rest are high contributors (HC) with 16 cogs. The offered cogs are not enough for all nodes to connect to all trees. Hence, we consider four scenarios. In scenario *NP* 25% of the nodes connect to t trees, 50% of them connect to $t-1$ trees, and the rest to $t-2$ trees independent of their contribution. In scenarios *P*, $Tax-P$ and $CA-P$ nodes are prioritized based on the number of their cogs. In scenario *P* the number of trees they can join is still random as in *NP*. In scenario $Tax-P$ every node connects to $t-1$ trees (taxation). In scenario $CA-P$ HC nodes connect to t trees, 67% of LC nodes connect to $t-1$ trees, the remaining 33% connect to $t-2$ trees (contribution-aware parent allocation). We use $E[\Xi_S]/E[\Xi_F] = 11$ for all scenarios, that is, the reconnection time is shorter in the fertile trees, but prioritizing HC nodes does not decrease their reconnection times. Based on Fig. 21 a further increase of $E[\Xi_S]/E[\Xi_F]$ would not significantly influence the results. We do not model the decrease of $E[\Xi_F^{HC}]$ and $E[\Xi_S^{HC}]$, neither the possible increase of $E[\Xi_F^{LC}]$ and $E[\Xi_S^{LC}]$. The effect of such inhomogeneity is like that of decreasing the loss probability seen by HC nodes and increasing that seen by LC nodes. Hence, it is equivalent to the case of inhomogeneous losses, for which we showed earlier that prioritization does not improve the overall performance in the stable state of the system (Fig. 12).

The best average performance is achieved by the $Tax-P$ scheme, the $CA-P$ scheme performs slightly better than the *NP* scheme. $CA-P$ achieves the best performance for HC nodes, but the worst for *LC* nodes. Consequently, giving incentives to HC nodes can contradict to the goal of improving the average performance of the overlay.

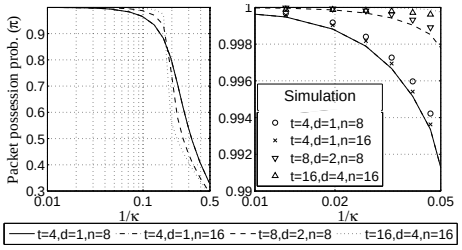

Figure 20: π vs $1/\kappa$ for $d > 1$ and $n > t$. $m = 50$, $\overline{N} = 10^4$, $k/n = 0.75$.

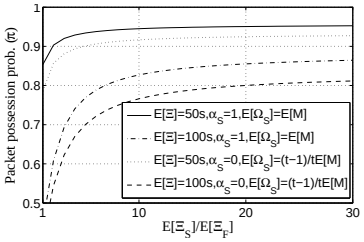

Figure 21: π vs the ratio of reconnection times for the NP preemptive scheme. $m = 50$, $\overline{N} = 10^4$.

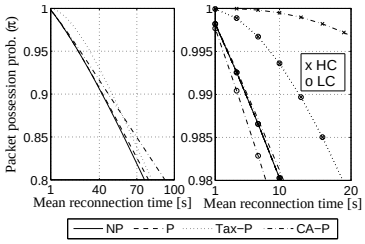

Figure 22: π vs $E[\Xi]$ for $m = 50$, $\overline{N} = 10^4$, $t = n = 8$, $k = 6$. Taxation and contribution aware parent allocation.

6. CONCLUSION AND PRACTICAL CONSEQUENCES

In this paper, we present an analytical model of the data distribution performance of multiple-tree-based overlay multicast architectures. We develop lower and upper bounds for a simple class of overlays, and show that the overlay is either in a stable or an unstable state depending on the packet loss probabilities and the size of the overlay. Our findings lead us to the definition of an overlay architecture with limited layer spread with improved stability and scalability properties. Using the model, we evaluate the effects of inhomogeneous and correlated losses, heterogeneous input and output capacities, and investigate how prioritization can improve the overlay's performance. We show that the effects of node churn are determined by the ratio of the reconnection time and parent disconnection intensity, and are similar in nature to those of packet losses. Based on our results we can draw a number of practical consequences that can serve as design guidelines for future systems.

FEC is the key to the scalability and good performance of multiple-tree-based overlay multicast. The FEC block length and the ratio of redundancy determine the performance of the overlay. Nevertheless, longer FEC codes do not necessarily improve the performance: they can make the overlay unstable if the number of trees is not increased. There is a need for an adaptive control algorithm to adjust the FEC block length and the ratio of redundancy, as node churn and the packet loss rates change dynamically.

Retransmissions and FEC are both needed to define an efficient and scalable overlay architecture. FEC gives scalability in terms of number of nodes and retransmissions decrease the ratio of redundancy needed. If the retransmission requests are limited to the parent within the tree, then retransmissions do not decrease the loss probability caused by the disconnections after node departures. Hence, in order to achieve high packet possession probability without having to introduce much redundancy, every node should maintain a list of backup parents. Backup parents can be asked occasionally to retransmit a piece of data, and should be located no deeper in the tree than the parents of the node.

Prioritization: The primary benefit of prioritization is the decrease of disturbances in the trees where a node forwards data. We show that prioritization does not necessarily improve the overall system performance, but it gives incentives to nodes with good performance.

Stability: If the overlay is stable, the number of layers does not influence the performance. The number of layers influences however the region of stability, so that the number of layers has to be kept low, e.g., by prioritizing high contributor nodes. The stability region can be increased by using shorter FEC codes, though shorter FEC codes give inferior performance in the case of stability.

Limited layer spread: It is possible to increase the stability region of large overlays by limiting the spread between the layers where nodes receive data. Limiting the layer spread also helps to decrease the effects of nodes with poor connections on the performance of high contributors. While one can argue about the fairness of this solution, it definitely gives incentives to nodes to contribute.

The model we propose can easily be extended, and can be a useful tool for future system designers. As a first step, we will incorporate the effects of per hop delay characteristics, and evaluate solutions to manage end-to-end delays in overlay multicast. It is an open question how the model can be applied to pull-based (a.k.a swarming) overlay multicast systems. We believe that there are many similarities between the two approaches, but we leave this as an area of future work.

7. REFERENCES

[1] Y. Chu, A. Ganjam, T.S.E Ng, S.G. Rao, K. Sripanidkulchai, J. Zhan, and H. Zhang, "Early experience with an Internet broadcast system based on overlay multicast," in *Proc. of USENIX*, 2004.

[2] Y. Chu, S.G. Rao, S. Seshan, and H. Zhang, "A case for end system multicast," *IEEE J. Select. Areas Commun.*, vol. 20, no. 8, 2002.

[3] M. Castro, P. Druschel, A-M. Kermarrec, A. Nandi, A. Rowstron, and A. Singh, "SplitStream: High-bandwidth multicast in a cooperative environment," in *Proc. of ACM SOSP*, 2003.

[4] V. N. Padmanabhan, H.J. Wang, and P.A Chou, "Resilient peer-to-peer streaming," in *Proc. of IEEE ICNP*, 2003, pp. 16–27.

[5] E. Setton, J. Noh, and B. Girod, "Rate-distortion optimized video peer-to-peer multicast streaming," in *Proc. of ACM APPMS*, 2005, pp. 39–48.

[6] M. Bishop, S. Rao, and K. Sripanidkulchai, "Considering priority in overlay multicast protocols under heterogeneous environments," in *Proc. of IEEE INFOCOM*, April 2006.

[7] V. Venkataraman, K. Yoshida, and P. Francis, "Chunkyspread: Heterogeneous unstructured end system multicast," in *Proc. of IEEE ICNP*, Nov. 2006.

[8] X. Zhang, J. Liu, B. Li, and T.S.P. Yum, "Coolstreaming/donet: A data-driven overlay network for peer-to-peer live media streaming," in *Proc. of IEEE INFOCOM*, 2005.

[9] "OctoShape," http://www.octoshape.com/, June 2007.

[10] "MediaZone Internet TV," http://www.mediazone.com/p2p/index.jsp, July 2006.

[11] 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. of ACM SIGCOMM*, 2004, pp. 107–120.

[12] Y-W. Sung, M. Bishop, and S. Rao, "Enabling contribution awareness in an overlay broadcasting system," in *Proc. of ACM SIGCOMM*, 2006, pp. 411–422.

[13] P.B. Godfrey, S. Shenker, and Stoica. I., "Minimizing churn in distributed systems," in *Proc. of ACM SIGCOMM*, 2006, pp. 147–158.

[14] I.S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *SIAM J. Appl. Math.*, vol. 8, no. 2, pp. 300–304, 1960.

[15] E. O. Elliott, "Estimates of error rates for codes on burst-noise channels," *Bell Syst. Tech. J.*, vol. 42, pp. 1977–1997, September 1963.

[16] Gianfranco Ciardo and Kishor S. Trivedi, "A decomposition approach for stochastic reward net models," *Performance Evaluation*, vol. 18, no. 1, pp. 37–59, 1993.

[17] J.S. Yedidia, W.T. Freeman, and Y. Weiss, *Understanding Belief Propagation and its Generalizations*, Exploring Artifical Intelligence in the New Millenium, ISBN 1-55860-811-7. 2003.

[18] Ellen W. Zegura, Ken Calvert, and S. Bhattacharjee, "How to model an internetwork," in *Proc. of IEEE INFOCOM*, March 1996, pp. 594–602.

[19] E. Veloso, V. Almeida, W. Meira, A. Bestavros, and S. Jin, "A hierarchical characterization of a live streaming media workload," in *Proc. of ACM IMC*, 2002, pp. 117–130.

[20] K. Sripanidkulchai, B. Maggs, and H. Zhang, "An analysis of live streaming workloads on the Internet," in *Proc. of ACM IMC*, 2004, pp. 41–54.

[21] E. N. Gilbert, "Capacity of a burst-noise channel," *Bell Syst. Tech. J.*, vol. 69, pp. 1253–1265, Sept. 1960.

[22] Gy. Dán, V. Fodor, and G. Karlsson, "On the stability of end-point-based multimedia streaming," in *Proc. of IFIP Networking*, May 2006, pp. 678–690.

[23] J-Y. Le Boudec and M. Vojnovic, "Perfect simulation and stationarity of a class of mobility models," in *Proc. of IEEE INFOCOM*, March 2004.

[24] Gy. Dán, V. Fodor, and I. Chatzidrossos, "On the performance of multiple-tree-based peer-to-peer live streaming," in *Proc. of IEEE INFOCOM*, May 2007.

[25] Donald Gross and Carl M. Harris, *Fundamentals of Queueing Theory*, Wiley, New York, 1998.

APPENDIX

Proof of Theorem 1: At the fixed point of the discrete dynamic system the mean number of lost packets has to equal the mean number of reconstructed packets. The mean number of packets that a node can reconstruct is given by

$$r(\pi,p,n,k) = \sum_{j=k}^{n} \binom{n}{j} \pi^j (1-\pi)^{n-j} \sum_{l=0}^{j-k} (n-j+l) \binom{j}{l} p^l (1-p)^{j-l}. \quad (24)$$

The mean number of lost packets is $n\pi p$, so that

$$n\pi p = r(\pi,p,n,k). \quad (25)$$

Our goal is to show that the number of intersections of the lines $n\pi p$ and $r(\pi,p,n,k)$ on $(0,1]$ is no more than two, i.e. there are at most two fixed points.

First, we show that $r(1,p,n,k) < np$. We substitute $\pi = 1$ into

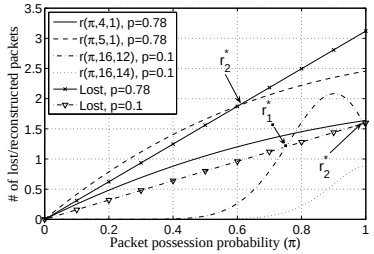

Figure 23: Number of lost and reconstructed packets vs. π for independent losses.

(25)

$$np = \sum_{l=0}^{n} lP(l,n) > \sum_{l=0}^{n-k} lP(l,n) = r(1,p,n,k) \quad (26)$$

for any loss distribution that satisfies $\sum_{l=n-k+1}^{n} P(l,n) > 0$, e.g., the Bernoulli loss model with $p > 0$.

For $k = 1$ we know that $r(\pi,p,n,1)$ is concave on $(0,1]$, as

$$\begin{aligned} r^{(1)}(\pi,p,n,1)|_{\pi=0} &= n(n-1)(1-p) > 0, \\ r^{(2)}(\pi,p,n,1)|_{\pi=0} &= -n^2(n-1)(1-p)^2 < 0, \end{aligned}$$

and the second derivative has one nonzero root at $1/(1-p) > 1$, so that there can be no inflection point on $(0,1]$. Due to the concavity on $(0,1]$, the two curves intersect in one point iff $r^{(1)}(0,p,n,1) > np$, i.e. $p < (n-1)/n$, otherwise they do not intersect.

For $1 < k < n$ we start by showing that there is a π^{**} for which $r(\pi,p,n,k)$ is convex for $0 < \pi < \pi^{**}$. We know that $r(0,p,n,k) = 0$, $r^{(1)}(0,p,n,k) = 0$, and that there is π for which $r(\pi,p,n,k) > 0$. Since $r(\pi,p,n,k)$ is a continuous function, $r^{(1)}(\pi,p,n,k) > 0$ for some $\pi > 0$ and hence $r^{(2)}(\pi,p,n,k) > 0$ as well. Thus, π^{**} exists and is the smallest positive inflection point.

Now it is enough to show that $r(\pi,p,n,k)$ has at most one inflection point on $(0,1]$, and hence it is either a convex curve or the combination of a convex and a concave curve.

For any $k > 1$ $r^{(2)}(\pi,p,n,k)$ has $n-k$ nonzero real roots, $\pi_1^{**} = \frac{1}{1-p}$ of multiplicity $n-k-1$ and $\pi_2^{**} = \frac{k-1}{n(1-p)}$. Both π_1^{**} and π_2^{**} are inflection points as $r^{(3)}(\pi_1^{**},p,n,k) > 0$ and $r^{(3)}(\pi_2^{**},p,n,k) < 0$ (i.e., the second derivatives change sign). $1/(1-p) > 1$, so that $r(\pi,p,n,k)$ has an inflection point on $(0,1]$ iff $p \le (n-k+1)/n$, and the inflection point is π_2^{**}.

If $r(\pi,p,n,k)$ has no inflection point (it is convex on $(0,1]$) then the number of intersection points is 0, because of (26) and $r(0,p,n,k) = 0$. If $r(\pi,p,n,k)$ has one inflection point then the number of fixed points can be 0, 1 or 2.

If there is 1 fixed point r_1 then $r^{(1)}(r_1,p,n,k) = np$, and the fixed point is unstable. If there are two fixed points r_1 and r_2 ($r_1 < r_2$), then r_2 is asymptotically stable ($r(\pi,p,n,k) > n\pi p$ for $\pi \in (r_1,r_2)$, and $r(\pi,p,n,k) < n\pi p$ for $\pi > r_2$). For r_1 the contrary is true, hence it is unstable. □