Stability and Performance of Overlay Multicast Systems Employing Forward Error Correction

György Dán * Viktória Fodor

Laboratory for Communication Networks, School of Electrical Engineering, KTH, Royal Institute of Technology, Stockholm, Sweden

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. Our findings lead us to the definition of an overlay structure with improved stability properties. We show how and under what conditions overlays can benefit from the use of error control solutions, prioritization and taxation schemes. Based on our results, we identify the components that are needed to achieve good data distribution performance in multi-tree-based overlay multicast.

Key words: Overlay multicast, FEC, Data distribution performance, Stability, Performance bounds

1. Introduction

The peer-to-peer paradigm has proved to be an efficient means both for file distribution, 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 commercial content distribution networks for delivering 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.

A large number of overlay multicast architectures has been proposed by the research community ([1–7] and references therein), and a number of large scale commercial deployments of peer-to-peer streaming systems were also recorded [8,9]. There is however not much analytical understanding of the data distribution performance of these systems, 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 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 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.

* Corresponding author
Email addresses: gyuri@ee.kth.se (György Dán), vfodor@ee.kth.se (Viktória Fodor).

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 [2–6,10–12]. Multiple trees offer two advantages: they ensure graceful quality degradation in dynamic overlays, in which 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. We describe the model applied to multi-tree based overlay multicast, but the modeling approach can generally be applied to multicast data distribution employing FEC in multi-hop environments. For example, in overlays that employ the pull model, data reach the nodes via spanning trees of the overlay graph, and hence we believe that some of our results can be applied to pull based overlays by adapting the packet loss model. (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 improves the scalability of the overlay with respect to the number of nodes.

The rest of the paper is organized as follows. We review related work in Section 2. In Section 3 we give a description of the considered overlays. We develop the analytical model of the overlay in Section 4 and derive asymptotic bounds on the system performance in Section 5. We describe the simulation methodology used to validate the model in Section 6. We evaluate the effects of packet losses in Section 7 and apply the model to node churn in Section 8. We conclude our work in Section 9.

2. Related work

Peer-to-peer streaming systems utilizing a single transmission tree were analyzed in [13]. The authors derived results on the number of levels as a function of the upload capacities of the peers, and evaluated the probability of blocking arriving nodes in the overlay. The first models that describe the data distribution performance of multi-tree-based overlay multicast were proposed in [14,15] and showed that these systems exhibit a phase-transition when using FEC. The model we present here generalizes the above models, and makes it possible to evaluate the effects of node heterogeneity, churn and different overlay management policies. The effect of the forwarding capacity on multi-tree-based overlays was investigated in [16] using a queuing theoretic approach, and in [17] based on a fluid model.

The effect of node dynamics on the connectivity of peers was evaluated in [18] for peer-to-peer file sharing systems. The authors derived results for the time to isolation and the probability of isolation for various node lifetime distributions. The authors in [19] proposed master equations to model the evolution of the number of neighbors of the peers in an overlay, they did not provide however any closed form solution. To the best of our knowledge, ours is the first work to propose a general framework for modeling the data distribution performance of multi-tree based peer-to-peer streaming systems employing FEC in a dynamic environment.

3. System description

In this section we describe the considered general overlay structure in Section 3.1, our assumptions regarding the overlay maintenance and the data distribution in Sections 3.2 and 3.3 respectively.

3.1. *Overlay model*

The overlay consists of a source node and N peer nodes. The peer nodes are organized in τ distribution trees, and the source is the root of all trees. Each peer node is member of at least one tree, and in each tree it has a different predecessor node (called parent) from which it receives data. We say that a node that is l hops away from the source node in tree e is in level l

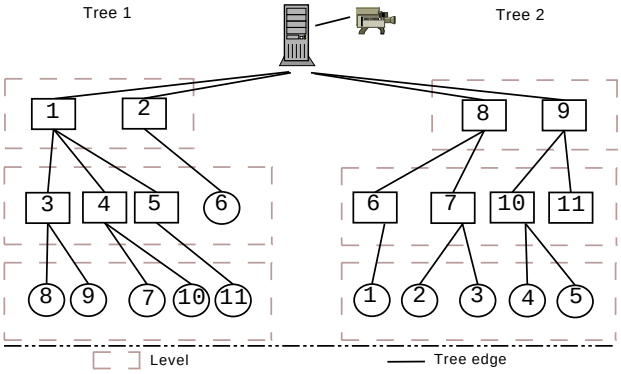

Fig. 1. Nodes, levels and parent-child relationships for an overlay with $N = 11, \tau = 2, m = 2$. The square indicates that the node forwards data in the tree.

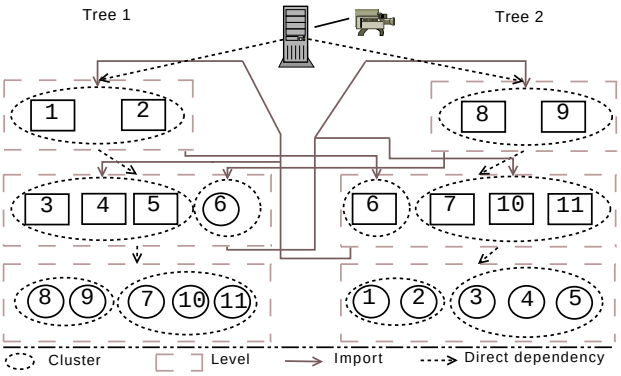

Fig. 2. Clusters, levels, direct dependencies and imports for the same overlay (see Section 4). The nodes are divided into five clusters.

of tree e. We denote the maximum outdegree (the maximum number of children) of the source node in each tree by m, and the maximum outdegree of a node r by d^r. m is limited by the ratio of the source's upload capacity and the stream's bitrate. d^r is limited by the ratio of the node's upload capacity and the stream's bitrate divided by the number of trees τ.

Nodes can split their forwarding capacity between s trees. In our model the nodes balance their forwarding capacity between the s trees, i.e., a node can have up to $\lceil d^r/s \rceil$ children in each of the s trees. One gets the minimum breadth trees described in [3] for $s = \tau$, and the minimum depth trees evaluated in [2,3,11] for $s = 1$. The case $1 < s < \tau$ was proposed in [16] to improve the overlay's stability under churn. Fig. 1 shows an overlay in which each node forwards data in one tree only ($s = 1$), and the source has two child nodes in each tree ($m = 2$). The solid black lines show the parent-child relations between the nodes in the overlay.

We introduce the notion of well-maintained overlay: the number of nodes that forward data is maximal in every level of its trees. Well-maintained overlays have the smallest depth for given N, τ and s. For instance, in a well-maintained overlay with L levels, each node is $1 \leq l \leq L$ hops away from the source node in the trees in which it forwards data, and $L-1 \leq l \leq L$ hops away in the trees in which it does not. Furthermore, for $s < \tau$ the number of levels L in the trees is $O(logN)$.

It is however not necessary for an overlay to be well maintained. Motivated by our modeling work, we propose a tree structure with *limited level spread*. In an overlay with level spread limit Δ_l a node that forwards data in level l in a tree should be located no deeper than level $l + \Delta_l$ in the other trees. We do not discuss here how to implement such a tree structure, our goal is to show its possible benefits assumed it can be implemented.

3.2. *Tree management*

The construction and the maintenance of the trees can be done either by a distributed protocol (structured, like in [2] or unstructured, like in [4]) or by a central entity, like in [3]. 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 on 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 source) and the nodes' priorities. Priorities were introduced in multi-tree-based overlays in order to allow the most important nodes to be closest to the source. In the simplest case nodes that forward data in a tree have high priority, and hence can preempt nodes that do not forward data in that tree. Such a strategy was proposed in [3] in order to push contributing nodes close to the source and non-contributing nodes to the last levels of the trees. Prioritization can also be based on more complex criteria, such as the packet reception probability of a node, the level spread of a node, the input capacity of a node or the maximum outdegree of a node (e.g., [5]).

We consider three aspects of the tree maintenance algorithm. First, it influences the number of levels in the overlay and the distribution of the nodes among the levels. 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 of node r in tree e, and model it with a random variable Ω_e^r. Third, it influences 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 model it with a random variable Ξ_e^r. 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 $E[\Xi_e^r]$ of the reconnection time can be up to tens of seconds depending on the tree management and the forwarding capacity in the tree [11].

3.3. *Data transmission and error resilience*

The source splits the data stream into τ stripes, with every τ^{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. Consequently, subsequent packets of the stream reach a node via different overlay paths.

The source uses block based FEC, e.g., Reed-Solomon codes [20], so that nodes can recover from packet losses due to network congestion and node departures. To every k consecutive 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 forwards data, is recovered, then it is sent to the respective children. Duplicate packets are discarded by the nodes. Since subsequent packets of an FEC block reach a node via different overlay paths, the packet loss process as seen by a node is close to independent, which improves the efficiency of FEC in reconstructing lost packets [21]. Using this FEC scheme one can implement unequal error protection (UXP), priority encoding transmission (PET), or the multiple description coding (MDC) scheme considered in [3]. If the source would like to increase the ratio of redundancy while maintaining its bitrate unchanged, then it has to decrease the source rate.

4. Performance metrics and data distribution model

The building blocks of the overlay are the individual nodes, so first we describe the model of a single node in Section 4.1. Using the notations introduced there we define the performance metrics we consider in Section 4.2. We define clusters of nodes in Section 4.3, and describe the model of the overlay in Section 4.4. We then turn to the modeling of node dynamics in Section 4.5, and describe how to estimate the overlay's structure in Section 4.6.

4.1. *Node model*

The input capacity of a node determines the number of trees the node can connect to. We denote the set of trees that node r can connect to by $\mathcal{H}^r$, $\mathcal{H}^r \subseteq \{1 \dots \tau\}$, $|\mathcal{H}^r| \leq \tau$. The maximum outdegree of the node in tree e is d_e^r, its number of children is w_e^r. Our model captures three sources of disturbances in the overlay.

First, a node cannot receive data in a tree if it is not connected to a parent node in that tree. We model whether a node is connected with the binary r.v. D_e^r, such that $D_e^r = 0$ corresponds to node r being disconnected in tree e. We assume that whether a node is disconnected in a tree is independent of it being disconnected in another tree, i.e., D_e^r is independent of D_h^r ($h \in \mathcal{H}^r \setminus \{e\}$). The independence assumption is reasonable if nodes do not have the same node as parent in different trees. We show how to calculate the probability $P(D_e^r = 0)$ in Section 4.5.

Second, a node might experience losses on its input link. (We refer as input link of a node to the part of the network that is shared between data arriving from all parent nodes.) We denote the probability that i out of j packets are lost on the input link of a node by $P_I^r(i,j)$. $P_I^r(i,j)$ can be calculated using loss models such as the Bernoulli model or the Gilbert model [22].

Third, a node might experience losses on its output link. (We refer as output link of a node to the part of the network that is shared between data departing to all child nodes.) We denote the probability that i out of j packets are lost on the output link of a node by $P_O^r(i,j)$. $P_O^r(i,j)$ can be calculated in a similar way as $P_I^r(i,j)$. We model these two loss processes separately because the correlations in the two loss processes will have different effects on the performance of the overlay.

4.2. *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.

4.3. *Decomposition and Clustering*

Modeling large-scale overlays on a per node basis is computationally not feasible, hence we introduce two techniques in order to make the data distribution model scalable.

Clustering of nodes: We introduce the notion of clusters of nodes. Nodes belonging to a cluster forward data in the same trees, have their parents in the same trees in the same levels, have the same input capacities, and experience the same input and output loss probabilities. Consequently, a level of a tree possibly consists of several clusters, the different clusters correspond to sets of nodes with different characteristics. We treat nodes within a cluster as stochastically identical, so that we only have to calculate the packet possession probabilities for clusters of nodes. Consequently, we can use the random variables introduced for individual nodes in Section 4.1 for clusters of nodes, e.g., we use the random variable D_e^f to model whether a node that belongs to cluster f is connected to its parent node in tree e. 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 level), we assume that a level appears to be homogeneous to nodes in the next level. The model can be used without this assumption, at the price of increased number of clusters.

Decomposition: We decompose the overlay into τ nearly independent subsystems. Such a decomposition approach was used for the solution of large stochastic Petri nets in [23]. The subsystems are the trees of the overlay: the probability that a node possesses a packet in a tree does not only depend on whether its parent node in the same tree possesses that packet, but is also dependent on the probability of some nodes possessing packets of the same block in other trees due to the use of FEC. We call the dependency within a tree *direct dependency*, such a dependency exists between a level and the clusters in the subsequent level. The dependencies of other trees are called *imports*. The dependency graph contains cycles for most overlay structures, hence, to solve the model, we provide initial guesses for the imports and use fixed point iteration.

The decomposition involves an assumption of independence: whether the parent of a node in tree e possesses a packet is independent of whether the parent of the same node in tree h possesses a packet of the same block. This assumption does not hold, for example, if nodes have the same parent in various trees. Nevertheless, 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 the independence assumption.

We illustrate clustering and decomposition in Fig. 2, which shows the clusters, the levels, the direct dependencies and the imports used in the model for the overlay shown in Fig. 1. Clustering reduces the calculation complexity of the model: instead of calculating the probability of receiving a packet for each of the 11 nodes, we only have to perform the calculations for the

Var.	Definition	Var.	Definition
N	# of nodes in the overlay	τ,m	# of trees and outdegree of the source respectively
s	# of trees in which a node can forward data	n,k	FEC block length and number of data pkts respectively
$\mathcal{H}^f$	Set of trees that nodes in cluster f connect to, $\lvert\mathcal{H}^f\rvert = \tau^f$	$\mathcal{C}^l$	Set of clusters that forward data in level l
N^f	# of nodes in cluster f	N(l)	# of nodes that forward data in level l
d^f	Maximum outdegree of nodes in cluster f	w_e^f	Average # of children of nodes of cluster f in tree e
$J(j)$	# of lost pkts in a block of j pkts, $P(J(j)=i) = P(i,j)$	$W_e(l)$	# of pkts successfully departing from nodes that forward in level l in tree e (not lost on output link)
X_e^f	# of pkts a node in cluster f can receive from its parent in tree e	Y_e^f	# of pkts a node in cluster f receives from its parent in tree e
V_e^f	# of tree e pkts possessed by a node in cluster f in tree e	Z_e^f	# of pkts that depart from a node in cluster f in tree e
$\pi(l)$	Packet possession probability of nodes that forward data in level l	π	Packet possession probability of an arbitrary node

Table 1
List of notations frequently used in the paper.

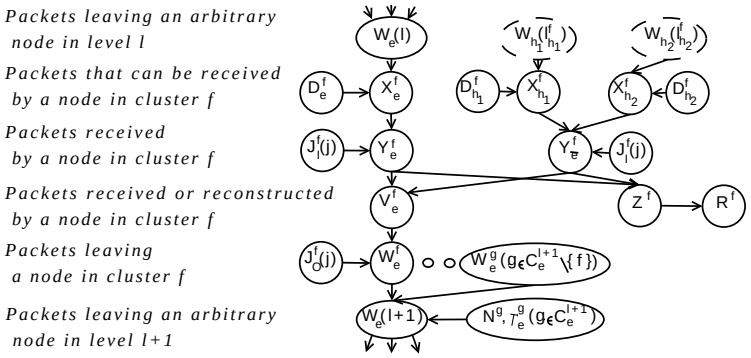

Fig. 3. Random variables and their relations used for the calculation of $W_e(l+1)$ from $W_e(l)$ through cluster $f \in \mathcal{C}_e^{l+1}$, $\mathcal{H}^f = \{e, h_1, h_2\}$. $W_{h_1}(l_{h_1}^f)$ and $W_{h_2}(l_{h_2}^f)$ are imports. Eqs. (1)-(8) give the relationships between the random variables.

5 clusters indicated in the figure. For example, nodes 3, 4 and 5 belong to the same cluster, because they receive data in Tree 1 from nodes in level 1, and in Tree 2 from nodes in level 2, and we assume for the example that they have the same input capacities and packet loss probabilities. Decomposition means that when we calculate the probability that a node receives a packet in Tree 1 then we assume that we already know the probability that it will receive some other packets in Tree 2. Once we have the new probabilities for Tree 1, we recalculate the results for Tree 2. The solution is then obtained in an iterative way.

4.4. *Data distribution model*

Let us consider a cluster f, in which nodes join trees $h \in \mathcal{H}^f$, $\mathcal{H}^f \subseteq \{1 \dots \tau\}$, $|\mathcal{H}^f| = \tau_f \geq 1$, and the parents of the nodes in tree h are in level l_h^f $(h \in \mathcal{H}^f)$. The key to the overlay's performance is the probability that a node in cluster f possesses the packets in the trees where it has to forward data. In the following we describe how to calculate this probability using the reasoning that a node possesses a packet in a tree if it receives the packet from its parent in the tree, or if it receives enough packets in order to reconstruct the packet using FEC. Let us denote by $\mathcal{C}_e^l$ the set of clusters that forward data in tree e in level l, and by the random variable V_e^f the number of packets possessed by a node in cluster f out of the n/τ packets it should forward in tree e. In the following we show how the distribution of this random variable can be calculated.

We chose to give the relationships between the random variables instead of the stochastic vectors representing their distributions, as we believe that this formulation makes understanding easier. Figure 3 shows a graphical representation of the calculation of the random variables described in the following.

Let us denote by the random variable $W_e(l)$ the number of packets out of the n/τ packets transmitted in tree e that successfully depart from an arbitrary node in level l in tree e of the overlay, i.e., the packets that do not get lost on the output links of the nodes. Note that $W_e(l)$ is related to the level of the tree and not to specific clusters, because we consider layers to appear as homogeneous towards lower levels of the overlay. Let us now consider a cluster f in level $l+1$ in tree e, i.e., $f \in \mathcal{C}_e^{l+1}$ and $l_e^f = l$. A node in cluster f can only receive a packet from its parent if it is connected to one. Hence, given $W_e(l)$ we can express the random variable X_e^f, the number of packets that nodes in cluster f can receive from their parents in tree e

$$X_e^f = W_e(l_e^f) D_e^f. \tag{1}$$

Similarly, we can define the number of packets that can be received in other trees based on the imports $W_h(l_h^f)$, $h \in \mathcal{H}^f \setminus \{e\}$ and D_h^f. Eq. (1) is approximate if $n/\tau > 1$, it assumes that D_e^f does not change during the transmission of a block of packets, even though a parent can depart and a parent can be found during the transmission of a block. The model can be extended so that this assumption does not have to be made, but this approximation works well if the time to transmit a block of packets is much shorter than the inter-disconnection and the reconnection times.

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 Y_e^f as the number of packets received by nodes of cluster f in tree e

$$Y_e^f = X_e^f - J_I^f(X_e^f), \tag{2}$$

where $J_I^f(j)$ is the number of lost packets out of j packets on the input link, and it is a random variable with distribution $P(J_I^f(j)=i)=P_I^f(i,j)$. Similarly, we can approximate the total number of packets received in the other trees

$$Y_{\bar{e}}^f = \sum_{h\in\mathcal{H}^f\setminus\{e\}} X_h^f - \sum_{h\in\mathcal{H}^f\setminus\{e\}} J_I^f(X_h^f). \tag{3}$$

If FEC(n,k) is used to recover missing packets then 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

$$V_e^f = \begin{cases} n/\tau & \text{if} \quad Y_e^f + Y_{\bar{e}}^f \geq k \\ Y_e^f & \text{otherwise.} \end{cases} \tag{4}$$

Now what remains is to show how $W_e(l+1)$ can be calculated. We express the random variable W_e^f, the number of packets out of n/τ packets that do not get lost on the output link of a node of cluster f

$$W_e^f = V_e^f - J_O^f(V_e^f), \tag{5}$$

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

$$W_e(l+1) = \frac{\sum_{f\in\mathcal{C}_e^{l+1}} W_e^f N^f w_e^f}{\sum_{f\in\mathcal{C}_e^{l+1}} N^f w_e^f}, \tag{6}$$

where N^f is the number of nodes in cluster f and w_e^f is the number of children in tree e of the nodes that belong to cluster f.

We start the calculation of the distributions of the above random variables by using the initial condition $P(V_e^{src}=n/\tau)=1$ ($1\leq e\leq\tau$), i.e., the source node possesses all data in all trees, and the imports $P(W_e(l)^{(0)}=0)=1$, $1\leq e\leq\tau$. Then, in iteration i, we calculate the distribution of $W_e(l)^{(i)}$, ($1\leq l<L$ and $1\leq e\leq\tau$) using the direct dependencies and the imports from iteration $i-1$. The iteration stops when $|E[W_e(L-1)^{(i-1)}]-E[W_e(L-1)^{(i)}]|<\varepsilon$, where $\varepsilon>0$. $E[W_e(l)^{(i)}]$ is monotonically increasing as long as (1)-(6) are monotonically increasing functions in their respective variables. Consequently, as $E[W_e(l)^{(i)}]\leq n/\tau$ the iteration converges.

The iterative solution we outlined here can be interpreted as the application of the belief propagation algorithm to a loopy Bayesian network partitioned into τ trees [24]. A Bayesian network is a graphical representation of conditional dependencies between random variables. The nodes of the graph are the random variables, in our case the V_e^f, and the arcs represent the dependencies that we have described above. The belief propagation algorithm is an iterative algorithm used to calculate the marginals of the joint distribution of the random variables represented by the nodes of the graph, i.e., in our case the distributions of the random variables V_e^f. The algorithm starts from the leaf nodes of the graph, in our case the leaf nodes are the source of the trees and the imports, and calculates the marginals in an iterative way.

Based on the final value of $W_e(l_e)^{(i)}$, we can express the random variable Z^f, the number of packets out of n that a node belonging to cluster f receives

$$Z^f = \sum_{e\in\mathcal{H}f} Y_e^f. \tag{7}$$

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

$$\pi^f = \frac{1}{n}R^f = \frac{1}{n}E[Z^f + \rho(Z^f)], \tag{8}$$

where $\rho(i)$ is the number of reconstructed packets if i packets are received in a block of n packets

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

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

$$\pi(l) = \frac{\sum_{f\in\mathcal{C}^l}\pi^f N^f}{\sum_{f\in\mathcal{C}^l} N^f}, \tag{9}$$

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

$$\pi = \frac{\sum_f \pi^f N^f}{\sum_f N^f}. \tag{10}$$

4.5. *Modeling node dynamics*

In the following section we calculate the probability that a node in cluster f is disconnected in tree e, i.e., the probability $P(D_e^f = 0)$. This probability is influenced by how often a node in cluster f loses its parent in tree e, and for how long it has to look for a new one. These two measures are influenced by the priority of the nodes of cluster f in tree e, because a node is likely to find a parent faster in a tree in which it has a high priority, and it is less often disconnected from its parent due to preemption. Consequently, we consider a set of trees $\mathcal{H}_b^f$, $|\mathcal{H}_b^f| = \tau_b$, in which the nodes of cluster f have the same priority. As an example, consider that nodes forward data in one tree only ($s = 1$), and nodes that forward data in tree e obtain a parent in tree e faster than those that do not forward data in tree e because of a prioritization scheme such as the one we explained in Section 3.2. Then for a cluster f that consists of nodes forwarding data in tree e, $\mathcal{H}^f$ consists of two sets of trees, the tree in which the nodes forward data $\mathcal{H}_F^f = \{e\}$, and the trees in which they do not forward data $\mathcal{H}_S^f = \mathcal{H}^f \setminus \mathcal{H}_F^f$. Consequently, $\tau_F = 1$ and $\tau_S = \tau - 1$, and $\mathcal{H}_b^f$ refers to one of these two sets. Since in the remainder of this section all random variables refer to the same cluster of nodes, we omit the superscript f in order to simplify the notation.

Let us denote by $\mathbf{u} = \{u_0, ..., u_{\tau_b}\}$ the stochastic vector who's i^{th} component contains the probability that a node is *not connected* to a parent in i of the τ_b trees that belong to $\mathcal{H}_b$ upon joining the overlay. The probability of a node being disconnected given the initial state distribution $\mathbf{u}$ can be expressed using the law of total probability

$$P(D_e = 0|\mathbf{u}) = \sum_{i=0}^{\tau_b} u_i P(D_e = 0|\mathbf{u}_i), \tag{11}$$

where $\mathbf{u}_i$ is the initial state distribution with exactly i disconnected parents.

In order to develop a closed form solution for $P(D_e = 0|\mathbf{u}_i)$, we assume that the distribution of the nodes' lifetimes (M), the inter-disconnection times (Ω_b) and the reconnection times (Ξ_b) can be modeled as exponential. That is, M is exponential distributed with parameter μ, $E[M] = 1/\mu$, Ω_b is exponential distributed with parameter ω_b, $E[\Omega_b] = 1/\omega_b$, and Ξ_b is exponential distributed with parameter ξ_b, $E[\Xi_b] = 1/\xi_b$. 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. We will evaluate the accuracy of the exponential modeling assumption in Section 8. Using the exponential assumptions, in the following we give a closed form expression for the probability $P(D_e = 0|\mathbf{u}_i)$.

Theorem 1 *For initial state distribution $\mathbf{u}_i$ the probability of a node being disconnected in tree $e \in \mathcal{H}_b$ is*

$$P(D_e = 0|\mathbf{u}_i) = \frac{\tau_b + i\alpha_b}{\tau_b(\kappa_b + \alpha_b + 1)}, \tag{12}$$

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

Proof We can model the evolution of the number of disconnected parents in trees $e \in \mathcal{H}_b$ of a node with a continuous time discrete state space Markov process $X(h) \in S$, $S = [0 \dots \tau_b]$. The transition intensities of the Markov process are

$$q_{i,i+1} = (\tau_b - i)\omega_b \qquad 0 \le i \le \tau_b - 1 \tag{13}$$

$$q_{i,i-1} = i\xi_b \qquad 1 \le i \le \tau_b. \tag{14}$$

The ratio of disconnected parents is $r_i = i/\tau_b$ in state i ($0 \le i \le \tau_b$) of the Markov process. The conditional probability $P(D_e = 0|\mathbf{u}_i)$ can be expressed as the average ratio of disconnected parents in trees $e \in \mathcal{H}_b$ of a node *as seen by a random observer* given $\mathbf{u}_i$. Without loss of generality we can denote the arrival time of the observer by 0,

$$P(D_e = 0|\mathbf{u}_i) = E[\Delta_b|\mathbf{u}_i] = \sum_{j=0}^{\tau_b} \frac{j}{\tau_b} P(X(0) = j|\mathbf{u}_i). \tag{15}$$

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

$$P(X(0) = j|\mathbf{u}_i) = \int_0^\infty p_{i,j}(t)dA(t). \tag{16}$$

$p_{i,j}(t)$ is given by $p_{i,j}(t) = P(X(0) = j|X(-t) = i) = e^{\mathbf{Q}t}_{\{i,j\}}$, 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 evolution of $\{p_{i,j}(t)\}$ is governed by the differential-difference equations

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

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

$$P_i(z,t) = \sum_{j=0}^{\tau_b} p_{i,j}(t)z^j = \frac{1}{(1+\kappa)^{\tau_b}}(B+Az)^{\tau_b-i}(D+Cz)^i, \tag{17}$$

where $A = 1 - M(t)$, $B = M(t) + \kappa_b$, $C = \kappa_b M(t) + 1$, $D = \kappa_b(1 - M(t))$, and $M(t) = e^{-\omega_b(1+\kappa_b)t}$.

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 t is exponential with parameter μ. Consequently, after substituting (16) into (15) we get

$$E[\Delta_b|\mathbf{u}_i] = \sum_{j=0}^{\tau_b} \frac{j}{\tau_b}\left\{\int_0^\infty p_{i,j}(t)\mu e^{-\mu t}dt\right\} = \int_0^\infty \left\{\sum_{j=0}^{\tau_b} \frac{j}{\tau_b}p_{i,j}(t)\right\}\mu e^{-\mu t}dt. \tag{18}$$

We can substitute the inverse z-transform of (17) into the sum on the right hand side of (18) to get

$$\sum_{j=0}^{\tau_b} \frac{j}{\tau_b}p_{i,j}(t) = \frac{(\tau_b - i)(1 - M(t)) + i(\kappa_b M(t) + 1)}{\tau_b(1+\kappa_b)}, \tag{19}$$

and by substituting (19) into (18) we get the theorem

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

Discussion of the result: The parameter $\kappa_b = \xi_b/\omega_b$ in (12) reflects the self-healing capability of the overlay: the higher its value, the more resilient is the overlay to node churn. Similarly, the parameter $\alpha_b = \mu_b/\omega_b$ in (12) reflects the likelihood of that a node will depart before one of its parents will be disconnected: the higher its value, the more likely that the node will depart before it gets disconnected. For $\mathbf{u}_{\tau_b}$ and $\mathbf{u}_0$ evaluating (17) leads to the well known product form solution [25], but we are not aware of any results for the general case described here. For $\alpha \to \infty$ (12) reduces to i/τ_b, 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 (12) 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.

4.6. *Overlay structure*

Important parameters of the model are the depth L of the overlay and the number of nodes per cluster N^f. These can be estimated for given overlay size N, maximum source degree m, number of trees τ, number of trees in which a node forwards data s and node parameters, such as input and output capacities.

The number of nodes that forward in level l of a well-maintained tree can be approximated by the recurrence $N_l = \sum_{r\in\mathcal{R}(l-1)} d^r/s$ with initial condition $N_1 = min(N/(\tau/s), m)$, where $\mathcal{R}(l-1)$ denotes the set of nodes that forward in level

$l-1$. Prioritization schemes affect the probability that nodes with certain properties (e.g., high output capacity) are located close to the source, and hence they influence the depth of the trees. Without prioritization, one can assume that nodes with different parameters are distributed uniformly among the levels. With prioritization, we assume that prioritized nodes are as close as possible to the source. There is a difference between the overlay structure estimated this way and the real overlay structure due to node churn and the distributed tree maintenance, but the simulations we present later show that the effects of these differences are negligible.

4.7. *Example*

Consider a well-maintained minimum depth overlay in which nodes are organized in $\tau = 3$ trees. The outdegree of the source is $m = 2$, and an FEC(3,2) code is used for error resilience. There are $N = 24$ nodes in the overlay in $L = 3$ levels. Assume that packet losses between nodes are i.i.d. with probability p, and the losses happen on the input links of the nodes, i.e., $P_I^f(i,j)$ follows a binomial distribution with parameter p. Since $n = \tau$, every node receives one packet of a block of n packets in a tree, and consequently the V_e^f are binary random variables. In each tree there are two clusters: the nodes that forward data in level 1, and the nodes that forward data in level 2.

Consider now tree e, in which the two clusters are $E1$ and $E2$. The source possesses all packets, hence we have $P(V_e^{src} = 1) = 1$, and since there are no losses on the output link of the source, we have $P(W_e(0) = 1) = 1$ from (5) and (6). Since we do not consider node churn in the example, we have $P(X_e^{E1} = 1) = P(W_e(0) = 1) = 1$ from (1). Nevertheless, packets get lost with probability p on the input links of the nodes, so that $P(Y_e^{E1} = 1) = P(X_e^{E1} = 1)(1-p)$ from (2). Nodes that forward packets in level 1 are in level 3 of the other trees, so that they receive both of the other packets in the FEC block from their parent nodes in level 2 with probability $P(Y_{\bar{e}}^{E1} = 2) = P(X_{\bar{e}}^{E1} = 2)(1-p)^2$ from (1)-(2). A node in level 1 possesses a packet if it receives it from the source, or if it receives the two other packets in the FEC block in level 3, i.e., $P(V_e^{E1} = 1) = P(Y_e^{E1} = 1) + (1 - P(Y_e^{E1} = 1))P(Y_{\bar{e}}^{E1} = 2)$ from (4). The trees are statistically identical, there is only one cluster per level in every tree, and there are no losses on the output links of the nodes, so that for tree $h \neq e$ we have $P(W_h(2) = 1) = P(W_e(2) = 1) = P(V_e^{E2} = 1)$ from (5) and (6). For the same reason we can also omit the subscripts denoting the trees, so that for the probability that a node in level 1 possesses a packet that it should forward we get

$$P(V^1 = 1) = P(V^{src} = 1)(1-p) + [1 - P(V^{src} = 1)(1-p)]P(V^2 = 1)^2(1-p)^2 = 1 - p + p(1-p)^2 P(V^2 = 1)^2. \quad (20)$$

Similarly, we can express the probability that a node in level 2 possesses a packet that it should forward, i.e., $P(V^2 = 1)$, as

$$P(V^2 = 1) = P(V^1 = 1)(1-p) + [1 - P(V^1 = 1)(1-p)]P(V^2 = 1)^2(1-p)^2. \quad (21)$$

For this simple example we do not need an iterative solution, but we can substitute (20) into (21), and solve for $P(V^1 = 1)$ and $P(V^2 = 1)$. For example, for $p = 0.1$ we get $P(V^1 = 1) = 0.9764$ and $P(V^2 = 1) = 0.9714$. We can use these results to calculate the distribution of Z^1 and Z^2, and finally π.

5. Overlay stability

In the following we analyze the stability of a class of overlays, and we establish bounds on the packet possession probability π as a function of the loss probability and the depth of the overlay. We observe that in all overlays proposed in the literature, nodes should be at least as close to the source in the trees in which they forward data as they are in the other trees. We consider the case $n = \tau$, so that the random variables V_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 $|\mathcal{H}^f| = \tau$. A consequence of this assumption is that all trees are statistically identical, i.e., the $W_e(l)$, $1 \leq e \leq \tau$ 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 i in a block of j packets follows a binomial distribution.

5.1. *Upper bound of the packet possession probability*

Using the above simplifying assumptions, from (1)-(6) and the initial condition $E[V_e^{src}] = n/\tau$ $(1 \le e \le \tau)$ it follows that $E[W_e(l)]$ is a non-increasing function of l. Hence, we can give an upper bound on $E[V_e^f] = P(V_e^f = 1)$ (V_e^f is a binary r.v. because $\tau = n$) by assuming that the parents of the nodes forwarding in a tree in level l are in level $l = min_{h \in \mathcal{H}} l_h^f$ in all trees. Since nodes are homogeneous, we only have to consider one cluster of nodes per level. Furthermore, since V_e^f is a binary random variable, (1)-(6) implies that a packet is possessed by a node if it receives it from its parent or if it receives at least k packets of the other $n-1$ packets of the block from its other parents, at most one packet from each parent. That is, if we denote the upper bound of the packet possession probability in level l by $\overline{\pi}(l)$, then

$$\overline{\pi}(l+1) = \overline{\pi}(l)(1-p) + (1-\overline{\pi}(l)(1-p)) \sum_{j=k}^{n-1} \binom{n-1}{j} \overline{\pi}(l)^j (1-\overline{\pi}(l))^{n-1-j} \sum_{i=0}^{j-k} P(i,j). \tag{22}$$

The $\overline{\pi}(l)$ can be calculated using the initial condition $\pi(0) = 1$. Similar to (10), the upper bound of the packet possession probability for an overlay with L levels and $N(l)$ nodes in level l can be calculated as

$$\overline{\pi} = \frac{\sum_{l=1}^{L} \overline{\pi}(l) N(l)}{N}. \tag{23}$$

5.2. *Asymptotic behavior*

Eq. (22) defines a non-linear recurrence relation for $\overline{\pi}(l)$, consequently we are interested in the existence of the fixed points of (22) on $(0,1]$.

Theorem 2 (Existence of fixed points) *For the i.i.d Bernoulli loss model the number of fixed points of (22) is* 0,1 *or* 2. *For* $k = 1$ *a fixed point exists and is asymptotically stable 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.*

Proof 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

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

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

$$n\pi p = r(\pi,p,n,k). \tag{25}$$

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. Fig. 4 illustrates the solution of (25) on four examples.

We start the proof by showing that $r(1,p,n,k) < np$. We substitute $\pi = 1$ into (25)

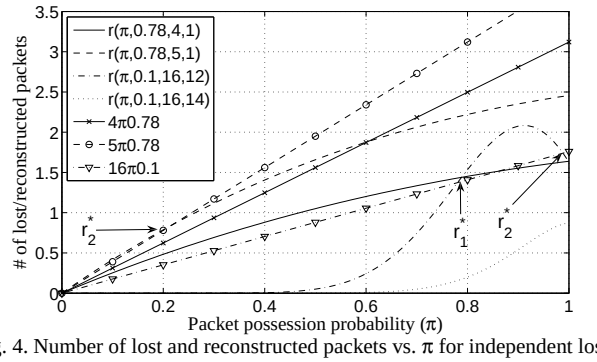

Fig. 4. Number of lost and reconstructed packets vs. π for independent losses.

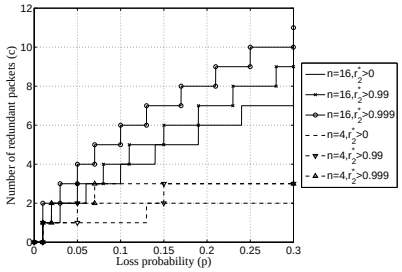

Fig. 5. c vs p for various objectives for the stable fixed point.

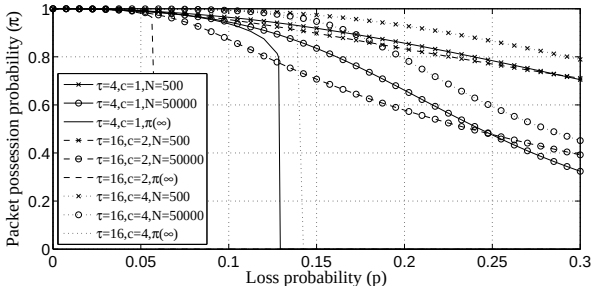

Fig. 6. Upper bound of the performance vs p for various number of trees and overlay sizes ($n = \tau$).

$$np = \sum_{i=0}^{n} iP(i,n) > \sum_{i=0}^{n-k} iP(i,n) = r(1,p,n,k) \tag{26}$$

for any loss distribution that satisfies $\sum_{i=n-k+1}^{n} P(i,n) > 0$, 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

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

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 of $r(\pi,p,n,1)$ on $(0,1]$, the two curves intersect in one point, denoted by r_2, iff $r^{(1)}(0,p,n,1) > np$, i.e., $p < (n-1)/n$. If it exists, r_2 is asymptotically stable and its domain of attraction is $(0,1]$. (E.g., the solid and the dashed lines in Fig 4.)

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.

If $\pi^{**} > 1$, that is, $r(\pi,p,n,k)$ has no inflection point on $(0,1]$, then $r(\pi,p,n,k)$ is convex on $(0,1]$, so that the number of intersection points is 0, because of (26) and $r(0,p,n,k) = 0$.

For $\pi^{**} \le 1$ it is enough to show that $r(\pi,p,n,k)$ has exactly one inflection point on $(0,1]$, and hence it is 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 then $\pi^{**} = \pi_2^{**}$. Consequently, if $p \le (n-k+1)/n$ then $r(\pi,p,n,k)$ has one inflection point on $(0,1]$ and the number of fixed points can be 0, 1 or 2. (E.g., the dotted line in Fig 4.)

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. Furthermore, the domain of attraction of r_2 is $(r_1,1]$. (E.g., the dash-dotted line in Fig 4.) □

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. 5 shows the number of redundant packets needed in a block of packets in order to achieve various objectives for the asymptotically stable fixed point r_2 as a function of the loss probability p.

If (22) has an asymptotically stable fixed point on $(0,1]$ then $\overline{\pi}(l)$ converges to that fixed point, and we say that the overlay is *potentially stable*: for the given loss probability and FEC parameters there exists an overlay structure for which a lower bound on π (the stable fixed point of (22)) can be given independent of the overlay's size. This overlay is a minimum breadth overlay ($s = \tau$) in which the nodes are in the same level in all trees. Otherwise, $\overline{\pi}(l)$ converges to 0, and the overlay is *unstable*: for the given loss probability and FEC parameters there is no overlay structure for which a lower bound on π can be given independent of the overlay's size.

Fig. 6 shows the theoretical upper bound of the packet possession probability as a function of the loss probability. The bound is obtained by combining $\overline{\pi}(l)$ from (22) with the node distribution $N(l)$ of a minimum depth overlay ($s = 1$) with N

nodes. The upper bound of the performance of an unstable overlay decreases as the overlay's size increases, while that of a potentially stable overlay is insensitive to the overlay's size. For large overlays ($N = \infty$) the upper bound is approximately equal to the stable fixed point of (22) if that exists, and is 0 otherwise.

5.3. *Sufficient condition for stability*

We call an overlay *stable* if it is potentially stable and for given overlay size, loss probability and FEC parameters the packet possession probability is no less than the asymptotically stable fixed point of (22). We can get a sufficient condition for the overlay to be stable using similar reasoning as used to obtain the necessary conditions.

Theorem 3 (Sufficient condition for stability) *For $k = 1$ if $p < (n-1)/n$ then the overlay is stable. For $k > 1$, if the number of fixed points of (22) is 2 and $(1-p)^L > r_1$ then the overlay is stable.*

Proof Let us denote the lower bound of the packet possession probability in level l by $\underline{\pi}(l)$. If there is no FEC reconstruction, then $\underline{\pi}(l) = (1-p)^L$. Using FEC, if $k = 1$ then according to Theorem 2 for $p < (n-1)/n$ there exists an asymptotically stable fixed point r_2 of (22) with domain of attraction $(0,1]$. Hence, after successive iterations of the model $\underline{\pi}(l) \geq \underline{\pi}(L) \geq r_2$. For $k > 1$, if the number of fixed points of (22) is 2 and $(1-p)^L > r_1$ then after successive iterations of the model $\underline{\pi}(l) \geq \underline{\pi}(L) \geq r_2$, the stable fixed point of (22). □

Consequently, the deeper the overlay, the smaller the range of loss probabilities for which an overlay with arbitrary structure is stable.

5.4. *Examples*

Example 1: Consider that an FEC(3,2) code is used to distribute data. We can calculate the fixed points of (22) analytically on $(0,1]$. The mean number of packets that can be reconstructed, (24), is

$$r(\pi, p, 3, 2) = 3\pi^2(1-\pi)[(1-p)^2] + \pi^3[3p(1-p)^2]. \tag{27}$$

In order to find the fixed points of (22) we solve the equation $3\pi p = r(\pi, p, 3, 2)$, which yields

$$r_1 = \frac{1-p-\sqrt{1-6p+5p^2}}{2(1-p)^2} \quad r_2 = \frac{1-p+\sqrt{1-6p+5p^2}}{2(1-p)^2}.$$

In order for two fixed points to exist we need $1-6p+5p^2 > 0$, i.e., $p < 0.2$. Consequently, for $p < 0.2$ one can construct an overlay of arbitrary size such that $\pi \geq \frac{1-p+\sqrt{1-6p+5p^2}}{2(1-p)^2}$. For $p > 0.2$ this is however not possible.

Example 2: Consider a minimum depth overlay in which nodes are organized in $\tau = 3$ trees. The outdegree of the source is $m = 2$, and an FEC(3,2) code is used for error resilience. There are $N = 24$ nodes in the overlay, so that if the overlay is well-maintained then $L = 3$, and the sufficient condition for stability is $(1-p)^3 \geq \frac{1-p-\sqrt{1-6p+5p^2}}{2(1-p)^2}$, i.e., $p \leq 0.1957$. The same condition for an overlay with $N = 10^5$, $L = 10$ would be $p \leq 0.137$.

6. Simulation methodology

Before presenting numerical results, we briefly describe the simulation environment we used to validate the results. We developed a packet level event-driven simulator to validate our model. We used the GT-ITM topology generator [26] 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 between the nodes are exponentially distributed, this assumption is supported by several measurement studies [27,28]. We consider two distributions for the session holding times M: the log-normal distribution [27] with CDF $F_M(x) = 0.5 + 0.5erf((ln(x)-a)/(b\sqrt{2}))$, $a = 4.93$, $b = 1.26$; and the shifted Pareto distribution [28] with CDF $F_M(x) = 1-(1+x/b)^{-a}$, $b = 612$, $a = 3$. In both cases the mean lifetime is $E[M] = 306s$ [27].

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 as in [29]. 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 source and obtains a random list of $g = 100$ members of every tree. The source 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 source, 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. As opposed to [29,30], 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 level where it looks for a parent.

In the non-prioritized preemption strategy the only preemption is when nodes that forward data in a tree can preempt nodes that do not forward data in that tree. This is necessary to push contributing nodes close to the source and non-contributing nodes to the last levels of the trees. b_1 is 1 if the node forwards data in the *tree* and it is 0 otherwise. This strategy was proposed in [3], and we will refer to it as NP.

The second preemption strategy is specific to some performance measure, such as the packet reception probability, the maximum outdegree 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. For example, if we want to prioritize nodes according to the packet loss probabilities they experience, we set b_0 to $\lceil 255(1-p) \rceil$. Another example for a strategy that fits into this framework is the one proposed in [5], in which prioritization is based on the maximum forwarding capacities of the peers. We will refer to this strategy by *P*.

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' capacities 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 [31]. 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 [14].

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 [32]. 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.

7. Performance evaluation: Packet loss

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. The results we present were obtained with the mathematical model presented in Section 4, we show simulation results to validate the simplifying assumptions of the model when necessary. Most figures we show are composed of two sub-figures. The sub-figure 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 analytical and simulation results.

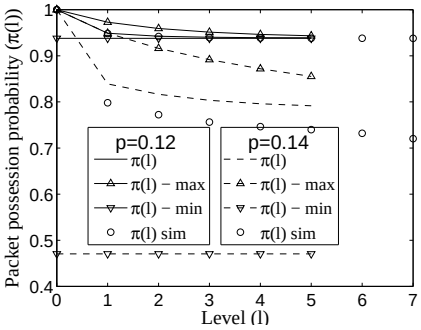

Fig. 7. $\pi(l)$ vs l. $\overline{N} = 50000$, $n = \tau = 4$, $k = 3$, $m = 50$, homogeneous case.

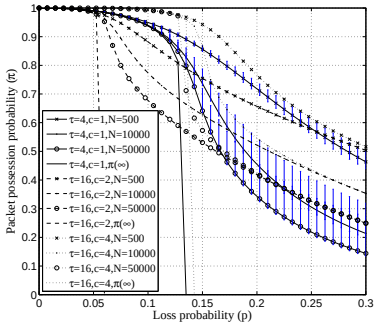

Fig. 8. π vs loss probability for $n = \tau$, $m = 50$, homogeneous case.

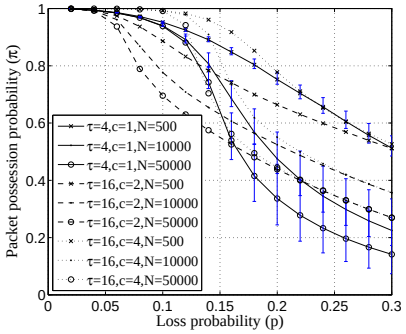

Fig. 9. π vs loss probability for $n = \tau$, $m = 50$, homogeneous case. Simulation results.

7.1. *The minimum depth overlay*

We start the evaluation with the minimum depth overlay, that is $s = 1$, as this is the most common multi-tree-based overlay structure in the literature [2,3,5,11,29,30]. We begin 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 in the same level in all trees in which it does not forward data, i.e., either the penultimate or the last level. Thus the nodes that forward data in a level of the tree belong to one of two clusters depending on the level where they are in the trees in which they do not forward data. The nodes are members of all τ trees and the outdegree of each node is $d^r = \tau$. We consider independent, homogeneous losses on the overlay links, so that $P_I^f(i, j)$ follows a binomial distribution with parameters j, p, and $P_O^f(0, j) = 1$ for all clusters.

Figure 7 shows the packet possession probability as a function of the level where nodes are in the tree in which they forward data for two loss probabilities $p = 0.1$ and $p = 0.14$. The stability threshold is $p_{max} = 0.129$ for the given FEC parameters, i.e., for $p = 0.14$ the overlay is unstable. For $p = 0.1$ the analytical results show a perfect match with the simulation results. For the unstable overlay the analytical results slightly overestimate the simulation results, because the trees are deeper in the simulations than calculated for a well-maintained tree. The upper bound of the packet possession probability given by (22) is tight for the potentially stable overlay only: in the unstable overlay the poor reception in the last level impacts the performance of the uppermost level. The lower bound given in Section 5.3 is far below in the unstable state, which shows that FEC reconstruction improves π significantly in the unstable state as well.

Figure 8 plots π as a function of the loss probability. Figure 9 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.

Figs. 8 and 10 show that increasing the FEC block length, in general, improves the performance of FEC in accordance with earlier results on FEC performance [21]. Fig. 8 shows that π can be increased at a given redundancy rate by increasing the number of trees τ and the block length n. Fig. 10 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.

7.2. *Splitting the forwarding capacity*

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 [16]. Figure 10 shows π as a function of p for cases when $s > 1$. To decrease the number of clusters, we assume for the model that a node is in the same level in the trees in which it forwards data. The simulation results in the right sub-figure show that this approximation is accurate. As shown in the figure, for the considered independent losses increasing s decreases the stability region. Consequently, to improve FEC performance it looks more favorable to increase n without increasing τ and s. We will see that under node churn the contrary is true in Section 8.

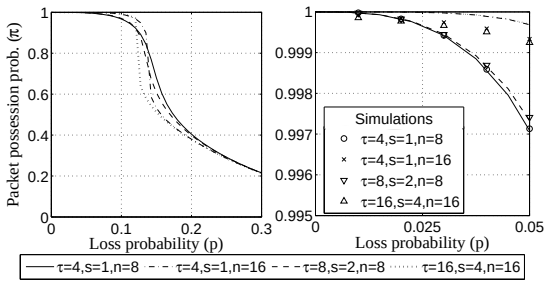

Fig. 10. π vs p for $s > 1$ and $n > \tau$, $m = 50$, $\overline{N} = 10^4$.

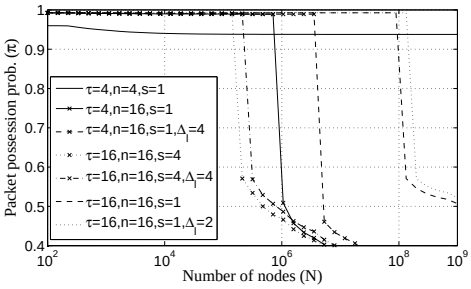

Fig. 11. π vs number of nodes at $p = 0.10$, $k/n = 0.75$, $m = 50$.

The minimum breadth overlay The minimum breadth overlay, in which nodes forward data in all trees, is the $s = \tau$ special case of $s > 1$ and has been studied earlier in the literature. The number of levels and the average number of hops between the source and the peer nodes in this overlay is $O(N)$, so that nodes have to remain in almost the same level 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 level l of the overlay approaches the upper bound given in (22). A detailed analysis of this overlay was presented in [14].

7.3. *Overlay size*

Figure 11 shows the dependence of π on the number of nodes in the overlay. We conclude that a stable overlay can become unstable for two reasons: increased packet losses and increased number of levels. It is not the number of nodes that causes the degradation, but the number of levels needed to accommodate them. Consequently, an overlay can become unstable for lower values of N if the tree maintenance algorithm cannot keep the trees close to well-maintained.

Surprisingly, for $\tau = n = 4$ the overlay is stable in the whole considered interval, for $\tau = 16$, $n = 4$ and for $\tau = n = 16$ it is however not, even though the overlay is not as deep for $\tau = n = 16$. This result seems counter-intuitive at the first sight, as in point-to-point communications longer FEC blocks are usually more efficient [21]. Nevertheless, in the case of multiple trees, FEC reconstruction close to the source requires packet reception in the trees, in which nodes are located in the last levels, and consequently are likely not to receive the packets unless FEC reconstruction works close to the source. Thus, a longer FEC code leads to higher possession probability if the system is stable, the region of stability is however smaller.

7.4. *Limiting the level 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 source in the tree in which they forward, since reconstruction requires packet reception in the other trees, in which nodes are located in the last levels. Motivated by this deficiency, we consider how our proposal to limit the level spread influences the overlay's performance. Limiting the level spread can of course increase the number of levels in the overlay, but it makes FEC reconstruction more efficient. Figure 11 shows that limiting the level spread does not decrease the performance of a stable overlay, but, as expected, the overlays with limited level spread remain stable for larger values of N.

7.5. *Sensitivity to correlated losses*

One of the major detriments of FEC is its poor performance when losses are correlated. In order to evaluate how loss correlations affect the performance of overlay multicast employing FEC we show π for correlated losses on the input links or on the output links of the nodes in Fig. 12. We used the Gilbert model with a conditional loss probability of $p_{\omega|\omega} = 0.3$ to calculate $P_I^f(i,j)$ and $P_O^f(i,j)$, respectively. Correlations on the output links of the nodes have no effect on the performance if $n = \tau$, since the consecutive packets will be received by different child nodes. Correlations on the input links decrease however the performance compared to the case of independent losses for $n = \tau$. A longer FEC block ($n > \tau$) increases the packet possession probability for both kinds of correlations when the overlay is stable. Based on the model we know that for correlated losses on the output links and for $n > \tau$ the performance approaches that of $n = \tau$ as $p_{\omega|\omega}$ increases. Correlated losses affect the overlay's performance mostly at low loss probabilities as correlations decrease the mean number

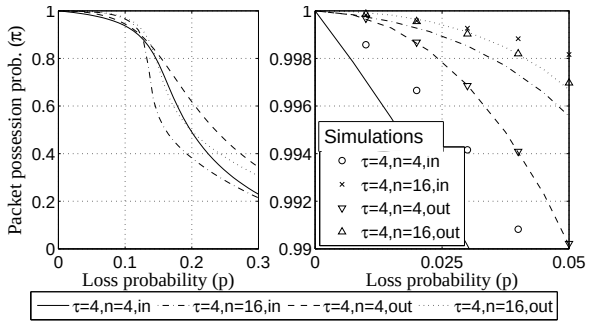

Fig. 12. π vs p for $\tau = 4, k/n = 0.75$, $m = 50$, $\overline{N} = 10^4$, $p_{\omega|\omega} = 0.3$ correlated losses on the input or on the output links.

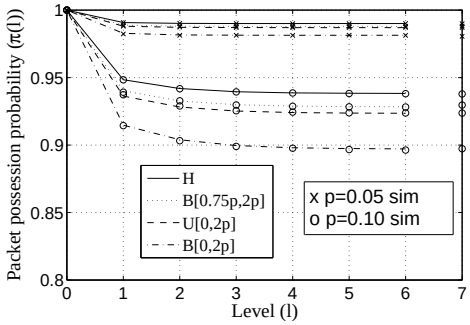

Fig. 13. $\pi(l)$ vs l for inhomogeneous losses. $\overline{N} = 10^4$, $m = 50$, $\tau = n = 4$, $k = 3$, $s = 1$. Model and simulations.

of reconstructed packets. Consequently, correlations decrease the system's region of stability and its region of potential stability. The simulations shown in the right sub-figure 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 (and in real systems), hence the loss correlation between packets in a block in the simulation is lower than $p_{\omega|\omega}$.

7.6. *Inhomogeneous losses*

Figure 13 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 level 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. 14 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 level spread (Inhom-PL) with $\Delta_l = 2$. We consider $\tau = 4$, 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 due to prioritization, limiting the level spread giving slightly larger gain. In the unstable region, prioritization pays off as the decrease of π becomes much slower.

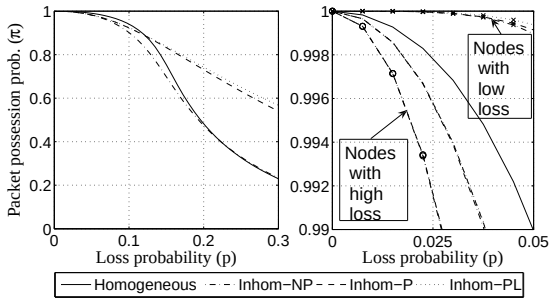

Fig. 14. π vs packet loss probability for inhomogeneous losses and prioritization. $\overline{N} = 10^4$, $m = 50$, $\tau = n = 4$, $k = 3$, $s = 1$.

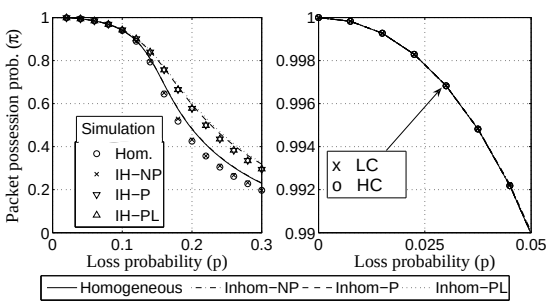

Fig. 15. π vs p for inhomogeneous output capacities. $\overline{N} = 10^4, m = 50, \tau = n = 4, k = 3$

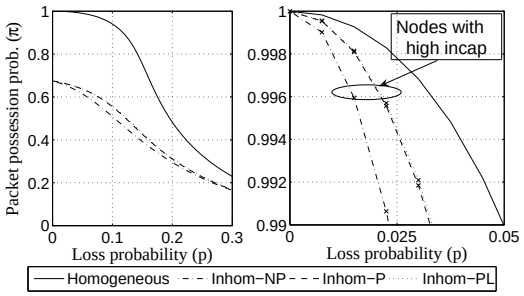

Fig. 16. π vs p for inhomogeneous input capacities. $\overline{N} = 10^4$,$m = 50$,$\tau = n = 4$,$k = 3$

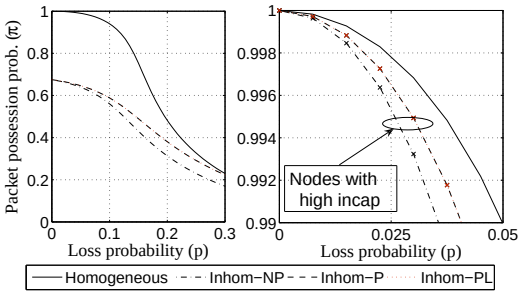

Fig. 17. π vs p for inhomogeneous input and output capacities. $\overline{N} = 10^4$,$m = 50$,$\tau = n = 4$,$k = 3$

7.7. *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 [11], as that is easier to estimate, but it would not help high contributor nodes joining the overlay for the first time.

Fig 15 considers an overlay with $\tau = 4$, and $\overline{N} = 10^4$, of which 65 percent are low contributors (LC) with $d^r = 2$ and 35 percent are high contributors (HC) with $d^r = 8$. This ratio of high and low contributors is similar to that considered in [11] 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 level spread with $\Delta_l = 2$. Prioritization does not make any difference for a stable overlay, as the number of levels 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 levels decreases due to prioritization 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 levels in the simulation and the one we calculated with.

Next, we consider inhomogeneous input capacities in Fig. 16 for $\tau = 4$ and $\overline{N} = 10^4$. 65 percent of the nodes have $|\mathcal{H}^r| = 2$ and the rest $|\mathcal{H}^r| = 4$. 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. 17: for the low contributors we use $|\mathcal{H}^r| = 2$, and for the high contributors we use $|\mathcal{H}^r| = 4$. The results show that the effects of prioritization are similar to those in Fig. 16, i.e., prioritization can give incentives to high contributors but does not improve the overall performance in the stable state. Limiting the level spread slightly improves the performance seen by high contributors, as expected.

8. Performance evaluation: Node churn

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 homogeneous input and output capacities and $E[\Xi_F] = E[\Xi_S]$, that is, the reconnection times are the same in the different trees. The simplicity of this scenario 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 $\mathbf{u}_\tau$ with graceful preemptions ($\alpha = 1$). We set $\overline{N} = 10^4$, $m = 50$. Figs. 18-20 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 $E[\Delta]$.

8.1. *Effects on the data distribution*

Next we apply the data distribution model to calculate π in the presence of node churn: for given κ we set $P(D_e^f = 0) = E[\Delta]$. The simulation results shown for $\mathbf{u}_0$ for the data distribution performance show a similarly good match in Fig. 21.

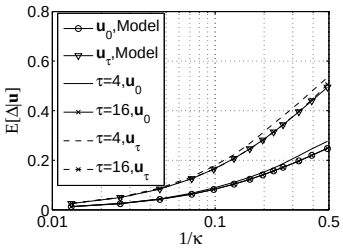

Fig. 18. $E[\Delta]$ vs $1/\kappa$ for log-normal lifetime and deterministic reconnection time distribution.

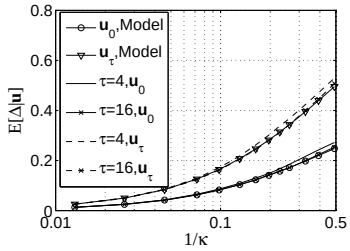

Fig. 19. $E[\Delta]$ vs $1/\kappa$ for Pareto lifetime and normal reconnection time distribution.

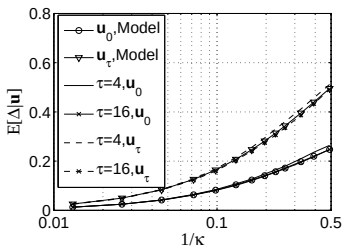

Fig. 20. $E[\Delta]$ vs $1/\kappa$ for Pareto lifetime and uniform reconnection time distribution.

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 as seen in Section 7. Fig. 22 shows that in the case of node departures this is not necessarily true. For $\tau = 4$, $n = 16$ the performance is equal to that of $\tau = 4$, $n = 8$, and in fact is equal to that of $\tau = n = 4$. Increased block length gives however increased performance if the number of trees and the number of trees in which a node forwards data increase as well, as shown in the figure for $s > 1$. The simulations were performed using the Pareto lifetime and normal reconnection time distributions and show that the approximation for $n > \tau$ is accurate.

8.2. *Why does preemption improve the performance?*

We showed in Section 7 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 [5,11] show that preemption does improve the overlay's stability. The two are not contradictory.

Fig. 23 shows π as a function of the ratio of the mean reconnection times of nodes in the trees in which they forward data ($E[\Xi_F]$) and in the ones in which they do not ($E[\Xi_S]$). 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 node that forwards data ($E[\Omega_S] = (t-1)/tE[M], \alpha = 0$). 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 [11] in Fig. 24. We consider an overlay with $\tau = n = 8$, $k = 6$, and $\overline{N} = 10^4$. 75% of the nodes are low contributors (LC) with maximum outdegree $d^r = 4$ and the rest are high contributors (HC) with maximum outdegree $d^r = 16$. The sum of all outdegrees is not enough for all nodes to connect to all trees. Hence, we consider four scenarios. In scenario *NP* 25% of the nodes connect to τ trees, 50% of them connect to $\tau - 1$ trees, and the rest to $\tau - 2$ trees independent of their contribution. In scenarios *P*, $Tax - P$ and $CA - P$ nodes are prioritized based on their maximum outdegrees. In scenario *P* the number of trees the nodes can join is random as in *NP*. In scenario $Tax - P$ every node connects to $\tau - 1$ trees (taxation). In scenario $CA - P$ HC nodes connect to τ trees, 67% of LC nodes connect to $\tau - 1$ trees, the remaining 33% connect to $\tau - 2$ 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 trees in which a node forwards data, but prioritizing HC nodes does not decrease their reconnection times. Based on Fig. 23 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

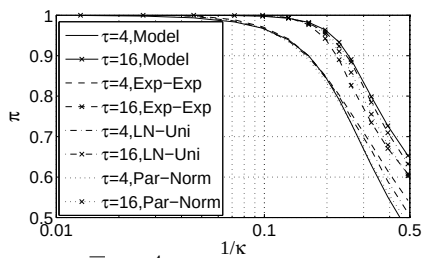

Fig. 21. π vs $1/\kappa$ for $\overline{N} = 10^4$, $n = \tau$, $k/n = 0.75$, $m = 50$, $\mathbf{u}_0$, the model and simulated lifetime distribution–reconnection time distribution pairs.

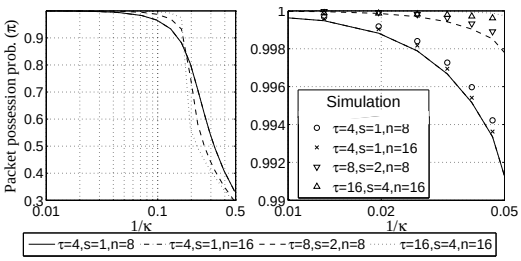

Fig. 22. π vs $1/\kappa$ for $s > 1$ and $n > \tau$. $m = 50$, $\overline{N} = 10^4$, $k/n = 0.75$.

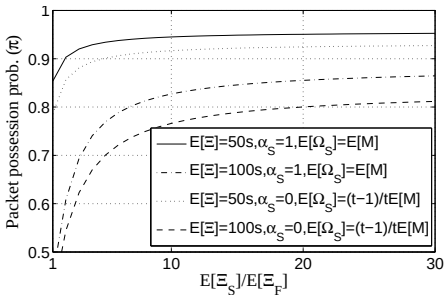

Fig. 23. π vs the ratio of reconnection times for the NP preemptive scheme. $m = 50$, $\overline{N} = 10^4$.

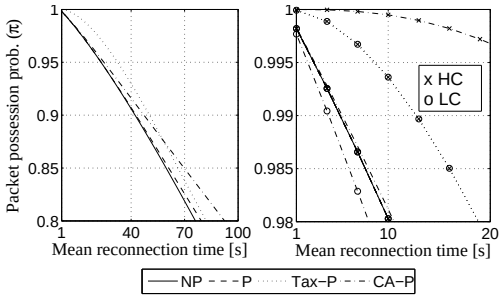

Fig. 24. π vs $E[\Xi]$ for $m = 50$, $\overline{N} = 10^4$, $\tau = n = 8$, $k = 6$. Taxation and contribution aware parent allocation.

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. 14).

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.

9. Conclusion and practical consequences

In this paper, we presented an analytical model of the data distribution performance of multiple-tree-based overlay multicast architectures. We developed lower and upper bounds for a class of overlays, and showed that the overlay's performance shows a phase transition depending on the packet loss probabilities and the size of the overlay. Our findings led us to the definition of an overlay architecture with limited level spread that shows improved stability and scalability properties. Using the model, we evaluated the effects of inhomogeneous and correlated losses, heterogeneous input and output capacities, and investigated how prioritization can improve the overlay's performance. We showed 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 performance in overlay multicast is determined by many system parameters, a number of which (e.g., loss probability, node churn and overlay size) change dynamically. Hence, the FEC block length and the ratio of redundancy have to be adjusted adaptively in order to maintain the system in a stable state. Albeit FEC can provide arbitrarily good performance, the necessary ratio of redundancy can be high if retransmissions are not used to decrease the packet loss rate between nodes.

Retransmissions and FEC are both needed to define an efficient and scalable overlay architecture. Nodes should maintain a list of backup parents in order to decrease the losses caused by node departures. 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. Otherwise, if 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.

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

Stability: If the overlay is stable, the number of levels does not influence the performance significantly. The number of levels influences however the region of stability, so that the number of levels 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 level spread: It is possible to increase the stability region of large overlays by limiting the spread between the levels where nodes receive data. Limiting the level 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 proposed model can easily be extended, and can be a useful tool for future system designers. 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.

References

[1] 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.

[2] 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.

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

[4] 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.

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

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

[7] 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.

[8] X. Hei, C. Liang, J. Liang, Y. Liu, and K.W. Ross, "A measurement study of a large-scale P2P IPTV system," *IEEE Trans. Multimedia*, vol. 9, no. 8, pp. 1672–1687, 2007.

[9] C. Wu and B. Li, "Characterizing peer-to-peer streaming flows," *IEEE J. Select. Areas Commun.*, vol. 25, no. 9, pp. 1612–1626, 2007.

[10] Y. Shan, I.V. Bajić, S. Kalyanaraman, and J.W. Woods, "Overlay multi-hop FEC scheme for video streaming," *Signal Processing: Image Communication*, vol. 20, no. 8, pp. 710–727, 2005.

[11] 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.

[12] G Wang, S. Futemma, and E. Itakura, "Multiple description coding for overlay network streaming," *IEEE Multimedia*, vol. 14, no. 1, pp. 74–82.

[13] T. Small, B. Liang, and B. Li, "Scaling laws and tradeoffs in peer-to-peer live multimedia streaming," in *ACM Multimedia*, October 2006.

[14] 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.

[15] Gy. Dán, V. Fodor, and I. Chatzidrossos, "Streaming performance in multiple-tree-based overlays," in *Proc. of IFIP Networking*, May 2007, pp. 617–627.

[16] 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.

[17] R. Kumar, Y. Liu, and K.W. Ross, "Stochastic fluid theory for P2P streaming systems," in *Proc. of IEEE INFOCOM*, May 2007.

[18] D. Leonard, Z. Yao, V. Rai, and D. Loguinov, "On lifetime-based node failure and stochastic resilience of decentralized peer-to-peer networks," *IEEE/ACM Trans. Networking*, vol. 15, no. 3, 2007.

[19] D. Carra, R. Lo Cigno, and E. Biersack, "Graph based modeling of P2P streaming systems," in *IFIP/TC6 Networking*, May 2007.

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

[21] Gy. Dán, V. Fodor, and G. Karlsson, "On the effects of the packet size distribution on FEC performance," *Computer Networks*, vol. 50, no. 8, pp. 1104–1129, 2006.

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

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

[24] 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.

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

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

[27] E. Veloso, V. Almeida, W. Meira, A. Bestavros, and S. Jin, "A hierarchical characterization of a live streaming media workload," *IEEE/ACM Trans. Networking*, vol. 14, no. 1, pp. 133–146, 2006.

[28] 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.

[29] 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.

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

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

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