Streaming Performance in Multiple-tree-based Overlays

György Dán, Viktória Fodor and Ilias Chatzidrossos

Laboratory for Communication Networks
School of Electrical Engineering
KTH, Royal Institute of Technology
Stockholm, Sweden
Email: {gyuri,vfodor,iliasc}@ee.kth.se

Abstract. In this paper we evaluate the data transmission performance of a generalized multiple-tree-based overlay architecture for peer-to-peer live streaming that employs multipath transmission and forward error correction. We give mathematical models to describe the error recovery in the presence of packet losses. We evaluate the data distribution performance of the overlay, its asymptotic behavior, the stability regions for the data transmission, and analyze the system behavior around the stability threshold. We argue that the composed measure of the mean and the variance of the packet possession probability can support adaptive forward error correction.

1 Introduction

The success of peer-to-peer overlays for live multicast streaming depends on their ability to maintain acceptable perceived quality at all the peers, that is, to provide data transmission with low delay and information loss. Live streaming solutions usually apply multiple distribution trees and some form of error control to deal with packet losses due to congestion and peer departures. In these systems peers have to relay data to their child nodes with low delay, which limits the possibilities of error recovery. Consequently, the main problem to be dealt with is the spatial propagation and thus the accumulation of losses, which results in low perceived quality for peers far from the source.

Several works deal with the management of the overlay, with giving incentives for collaboration, with peer selection and tree reconstruction considering peer heterogeneity and the underlying network topology ([1–6] and references therein).

In [7] the authors propose time shifting and video patching to deal with losses and discuss related channel allocation and group management issues. In [8] robustness is achieved by distributing packets to randomly chosen neighbors outside the distribution tree. In [9] retransmission of the lost data is proposed to limit temporal error propagation. CoopNet [10] and SplitStream [11] propose the use of multiple distribution trees and a form of multiple description coding (MDC) based on forward error correction

[1] This work was in part supported by the Swedish Foundation for Strategic Research through the projects Winternet and AWSI, and by Wireless@KTH

(FEC). In the case of packet losses peers can stop error propagation by reconstructing the video stream from the set of received substreams using error correcting codes.

There are several designs proposed and also implemented, the evaluation of these solutions is however mostly based on simulations and measurements. Our goal is to define abstract models of peer-to-peer streaming overlays that help us to understand some basic characteristics of streaming in multiple transmission trees and thus, can support future system design.

Previously, we proposed mathematical models to describe the behavior of CoopNet like architectures in [12–14]. Our results showed that the two architectures proposed in the literature, and used as a basis in recent works (e.g., [2, 15]) are straightforward but not optimal. Minimum depth trees minimize the number of affected peers at peer departure, minimize the effect of error propagation from peer to peer, and introduce low transmission delay. Nevertheless, the overlay is unstable and may become disconnected when one of the trees runs out of available capacity after consecutive peer departures. Minimum breadth trees are stable and easy to manage, but result in long transmission paths. Thus many nodes are affected if a peer leaves, there may be large delays, and the effect of error propagation may be significant.

In [16] we proposed a generalized multiple-tree-based architecture and showed that the stability of the overlay can be increased significantly by the flexible allocation of peer output bandwidth across several transmission trees. In this paper, we evaluate the performance of this generalized architecture. First we show how the allocation of peer output bandwidth affects the data distribution performance and discuss whether proper bandwidth allocation can lead to both increased overlay stability and good data distribution performance. We show that the packet possession probability at the peers decreases ungracefully if the redundancy level is not adequate, and evaluate how the variance of the packet possession probability can predict quality degradation.

The rest of the paper is organized as follows. Section 2 describes the considered overlay structure and error correction scheme. We evaluate the stability of the data distribution in Section 3. Section 4 discusses the performance of the overlay based on the mathematical models and simulations and we conclude our work in Section 5.

2 System description

The peer-to-peer streaming system discussed in this paper is based on two key ideas: the overlay consists of multiple transmission trees to provide multiple transmission paths from the sender to all the peers, and data transmission is protected by block based FEC.

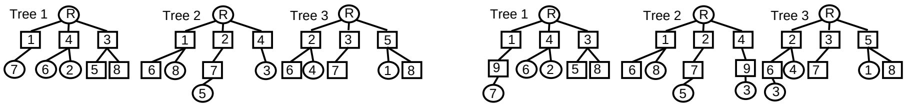
2.1 Overlay structure

The overlay consists of the streaming server and N peer nodes. The peer nodes are organized in t distribution trees with the streaming server as the root of the trees. The peers are members of all t trees, and in each tree they have a different parent node from which they receive data. We denote the maximum number of children of the root node in each tree by m, and we call it the multiplicity of the root node. We assume that nodes do not contribute more bandwidth towards their children as they use to download from their parents, which means, that each node can have up to t children to which it forwards data.

A node can have children in up to d of the t trees, called the fertile trees of the node. A node is sterile in all other trees, that is, it does not have any children. We discuss two different policies that can be followed to allocate output bandwidth in the fertile trees. With the *unconstrained* capacity allocation (UCA) policy a node can have up to t children in any of its fertile trees. With the *balanced* capacity allocation (BCA) policy a node can have up to $\lceil t/d \rceil$ children in any of its fertile trees.

By setting $d = t$ one gets the minimum breadth tree described in [10], and by setting $d = 1$ one gets the minimum depth tree evaluated in [2, 6, 10, 15]. For $1 < d < t$ the number of layers in the overlay is $O(logN)$ as for $d = 1$.

The construction and the maintenance of the trees can be done either by a distributed protocol (structured, like in [11] or unstructured, like in [9]) or by a central entity, like in [10]. The results presented in this paper are not dependent on the particular solution used. Nevertheless, we defined a centralized overlay construction algorithm in [16]. The objective of the algorithm is to minimize the depth of the trees and the probability that trees run out of free capacity after node departures. This is achieved by pushing sterile nodes to the lower layers of the trees and by assigning arriving nodes to be fertile in trees with the least available capacity. If we denote the number of layers in the trees by L, then in a well maintained tree each node is $1 \leq i \leq L$ hops away from the root node in its fertile trees, and $L-1 \leq i \leq L$ hops away in its sterile trees. As shown in [16], increasing the number of fertile trees increases the overlay stability, with significant gain already at $d = 2$, and the UCA policy giving the highest increase.



a)

b)

Fig. 1. a) Overlay with $N = 8, t = 3, m = 3$ and $d = 2$, b) the same overlay with $N = 9$. Identification numbers imply the order of arrival, squares indicate that the node is fertile.

Fig. 1 shows an overlay constructed with the proposed algorithm for $t = 3$, $m = 3$ and $d = 2$, also showing how the overlay changes when a new node joins.

2.2 Data distribution with forward error correction

The root uses block based FEC, e.g., Reed-Solomon codes [17], so that nodes can recover from packet losses due to network congestion and node departures. To every k packets of information it adds c packets of redundant information, which results in a block length of $n = k + c$. We denote this FEC scheme by FEC(n,k). If the root would like to increase the ratio of redundancy while maintaining its bitrate unchanged, then it has to decrease its source rate. Lost packets can be reconstructed as long as at most c packets are lost out of n packets. The root sends every t^{th} packet to its children in a given tree in a round-robin manner. If $n \leq t$ then at most one packet of a block is distributed over the same distribution tree. Peer nodes relay the packets upon reception to their respective child nodes. Once a node receives at least k packets of a block of n packets it recovers the remaining c packets and forwards the ones belonging to its fertile trees.

3 Data distribution model

We use two metrics to measure the performance of the data distribution in the overlay. The first metric is the probability π that an arbitrary node receives or can reconstruct (i.e., possesses) an arbitrary packet. If we denote by ρ_r the number of packets possessed by node r in an arbitrary block of packets, then π can be expressed as the average ratio of packets possessed in a block over all nodes, i.e., $\pi = E[\sum_r \rho_r/n/N]$. The second metric is σ, the average standard deviation of ρ_r/n over all nodes, i.e., $\sigma^2 = E[\sum_r (\rho_r/n)^2/N] - \pi^2$.

The mathematical model we present describes the behavior of the overlay in the presence of independent packet losses and without node dynamics. We denote the probability that a packet is lost between two adjacent nodes by p. We assume that the probability that a node is in possession of a packet is independent of that a node in the same layer is in possession of a packet. We also assume that nodes can wait for redundant copies to reconstruct a packet for an arbitrary amount of time. For the model we consider a tree with the maximum number of nodes in the last layer, hence nodes are fertile in layers $1..L-1$ and are sterile in layer L. For simplicity, we assume that $n = t$ and t/d is an integer. We will comment on the possible effects of our assumptions later. The model assumes the BCA policy. By modifying the weights in (1) the model can be used to consider other policies as well. For brevity we restrict the analytical evaluation to this case, and compare the results to simulations with other policies.

Hence, our goal is to calculate $\overline{\pi}^{(z)} = E[\sum_r (\rho_r/n)^{(z)}/N] = \sum_r E[(\rho_r/n)^{(z)}]/N$, the average of the z^{th} moment ($z \in \{1,2\}$) of the ratio of possessed packets. We introduce $\pi(i)^{(z)}$, the z^{th} moment of the ratio of possessed packets for a node that is in layer i in its fertile trees. For simplicity we assume that nodes are in the same layer in their fertile trees. We can express $\overline{\pi}^{(z)}$ by weighting the $\pi(i)^{(z)}$ with the portion of nodes that are in layer i of their fertile trees.

$$\overline{\pi}^{(z)} = \sum_{i=1}^{L-1} \frac{(t/d)^{i-1}}{((t/d)^{L-1}-1)/(t/d-1)} \pi(i)^{(z)}. \quad (1)$$

To calculate $\pi(i)^{(z)}$ we have to calculate the probabilities $\pi_f(i)$ that a node, which is in layer i in its fertile tree, receives or can reconstruct an *arbitrary packet in its fertile tree*. Since the root node possesses every packet, we have that $\pi_f(0) = 1$. The probability that a node in layer i receives a packet in a tree is $\pi_a(i) = (1-p)\pi_f(i-1)$. A node can possess a packet in its fertile tree either if it receives the packet or if it can reconstruct it using the packets received in the other trees. Reconstruction can take place if the number of received packets is at least k out of the remaining $n-1$, hence we can write for $1 \leq i \leq L-1$

$$\pi_f(i) = \pi_a(i) + \Bigg\{ (1-\pi_a(i)) \sum_{j=k}^{n-1} \sum_{u=max(0,j-n+d)}^{min(j,d-1)} \binom{d-1}{u} \pi_a(i)^u (1-\pi_a(i))^{d-1-u} \binom{n-d}{j-u} \pi_a(L)^{j-u} (1-\pi_a(L))^{n-d-j+u} \Bigg\}. \tag{2}$$

Based on the probabilities $\pi_f(i)$ we can express $\pi(i)^{(z)}$ $(1 \leq i \leq L-1)$. If a node receives at least k packets in a block of n packets then it can use FEC to reconstruct the lost packets, and hence possesses all n packets. Otherwise, FEC cannot be used to reconstruct the lost packets. Packets can be received in the d fertile trees and in the $t-d$ sterile trees. Hence for $\pi(i)^{(z)}$ we get the equation

$$\pi(i)^{(z)} = \frac{1}{n} \sum_{j=1}^{n-d} \sum_{u=1}^{d} \tau(j+u)^z \binom{d}{u} \pi_a(i)^u (1-\pi_a(i))^{d-u} \binom{n-d}{j} \pi_a(L)^j (1-\pi_a(L))^{n-d-j}, \tag{3}$$

where $\tau(j)$ indicates the number of packets possessed after FEC reconstruction if j packets have been received:

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

We use an iterative method to calculate the probabilities $\pi_f(i)$. First, we set $\pi_f(L-1)^{(0)} = (1-p)^{L-1}$ and calculate the probabilities $\pi_f(i)^{(1)}$, $1 \leq i < L$. Then, in iteration r, we calculate $\pi_f(i)^{(r)}$, $1 \leq i < L$ using $\pi_f(L-1)^{(r-1)}$. The iteration stops when $|\pi_f(L-1)^{(r-1)} - \pi_f(L-1)^{(r)}| < \varepsilon$, where $\varepsilon > 0$. The $\pi(i)^{(z)}$ can then be calculated using (3). The calculation of π and σ are straightforward, since $\pi = \overline{\pi}^{(1)}$, and $\sigma^2 = \overline{\pi}^{(2)} - \pi^2$

3.1 Asymptotic behavior for large *N*

In the following we give an asymptotic bound on π to better understand its evolution. It is clear that $\pi_f(i)$ is a non-increasing function of i and $\pi_f(i) \geq 0$. Hence, we can give an upper estimate of $\pi_f(i)$ by assuming that the nodes that forward data in layer i are sterile in the same layer. Then, instead of (2) we get the following nonlinear recurrence equation

$$\overline{\pi}_f(i+1) = \overline{\pi}_a(i+1) + (1-\overline{\pi}_a(i+1)) \sum_{j=n-c}^{n-1} \binom{n-1}{j} \overline{\pi}_a(i+1)^j (1-\overline{\pi}_a(i+1))^{n-1-j}. \tag{4}$$

This equation is the same as (2) in [13], and thus the analysis shown there can be applied to describe the evolution of $\overline{\pi}_f(i)$. For brevity, we only state the main results regarding

$\overline{\pi}_f(i)$, for a detailed explanation see [13]. For every (n,k) there is a loss probability p_{max} below which the packet possession probability $\overline{\pi}_f(\infty) > 0$ and above which $\overline{\pi}_f(\infty) = 0$. Furthermore, for any $0 < \delta < 1$ there is (n,k) such that $\overline{\pi}_f(\infty) \geq \delta$.

Consequently, in the considered overlay if $p > p_{max}$, then $\lim_{N\to\infty} \pi = 0$, because $\overline{\pi}_f(i+1) \geq \pi_f(i+1) \geq \pi(i+1)^{(1)}$, and $\lim_{N\to\infty} \overline{\pi}_f(L-1) = 0$. For $p < p_{max}$ stability depends on the number of layers in the overlay and the FEC block length due to the initial condition $\pi_f(L-1)^{(0)} = (1-p)^{L-1}$, but not directly on the number of nodes. This explains why placing nodes with large outgoing bandwidths close to the root improves the overlay's performance [2, 6]. In the case of stability $\pi_f(i) \geq \overline{\pi}_f(\infty) > 0$ and $\pi(i)^{(1)} \geq \overline{\pi}_f(\infty) > 0$.

3.2 Discussion of the assumptions

In the following we discuss the validity of certain assumptions made in the model. The model does not take into account the correlations between packet losses in the Internet. Nevertheless, it can be extended to heterogeneous and correlated losses by following the procedure presented in [13] for the minimum breadth trees. Losses occurring in bursts on the output links of the nodes influence the performance of the overlay if several packets of the same block are distributed over the same tree, that is, if $n > t$. Bursty losses in the backbone influence the performance if packets of different distribution trees traverse the same bottleneck. The analysis in [13] showed that loss correlations on the input links and heterogeneous losses slightly decrease the performance of the overlay.

The model can be extended to nodes with heterogeneous input and output bandwidths. The procedure is similar to that followed when modeling heterogeneous losses [13], but the effects of the heterogeneous bandwidths on the trees' structure have to be taken into account. We decided to show equations for the homogeneous case here to ease understanding.

In the analysis we assume that the number of nodes in the last layer of the tree is maximal. If the number of nodes in the last layer of the tree is not maximal then some nodes are in layer $L-1$ in their sterile trees, and the overlay's performance becomes slightly better. The results of the asymptotic analysis still hold however.

Our results for block based FEC apply to PET and the MDC scheme considered in [10], where different blocks (layers) of data are protected with different FEC codes. The packet possession probability for the different layers depends on the strength of the FEC codes protecting them, and can be calculated using the model.

We do not model the temporal evolution of the packet possession probability. We use simulations to show that the performance predicted by the model is achieved within a time suitable for streaming applications.

The model does not take into account node departures, an important source of disturbances for the considered overlay. Following the arguments presented in [13] node departures can be incorporated in the model as an increase of the loss probability by $p_\omega = N_d/N \times \theta$, where N_d is the mean number of departing nodes per time unit and θ is the time nodes need to recover (discovery and reconnection) from the departure of a parent node. The simulation results presented in [13] show that for low values of p_ω this approximation is accurate.

The results of the model apply for $n < t$ without modifications, and a similar model can be developed for $n > t$ by considering the distribution of the number of packets possessed by the nodes in their fertile trees. However, for $n > t$ node departures lead to correlated losses in the blocks of packets, which aggravates their effect on the performance.

4 Performance evaluation

In the following we analyze the behavior of the overlay using the analytical model presented in the previous section and via simulations. For the simulations we developed a packet level event-driven simulator.

We used the GT-ITM [18] topology generator to generate a transit-stub model with 10000 nodes and average node degree 6.2. We placed each node of the overlay at random at one of the 10000 nodes and used the one way delays given by the generator between the nodes. The mean delay between nodes of the topology is 44 ms. The delay between overlay nodes residing on the same node of the topology was set to 1 ms. Losses on the paths between the nodes of the overlay occur independent of each other according to a Bernoulli model with loss probability p. We consider the streaming of a 112.8 kbps data stream to nodes with link capacity 128 kbps. The packet size is 1410 bytes. Nodes have a playout buffer capable of holding 140 packets, which corresponds to 14 s of playout delay. Each node has an output buffer of 80 packets to absorb the bursts of packets in its fertile trees.

We assume that session holding times follow a log-normal distribution with mean $1/\mu = 306s$ and that nodes join the overlay according to a Poisson process with $\lambda = \overline{N}\mu$, supported by studies [19, 20]. 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 [21] and let nodes join and leave the overlay for 5000 s. The measurements are made after this warm-up period for a static tree over 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.

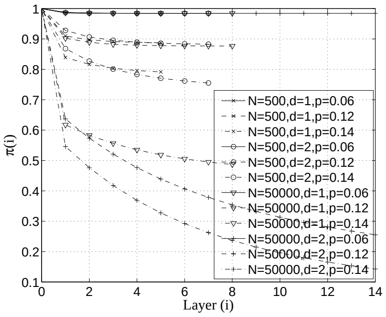


Fig. 2. $\pi(i)$ vs. i for $m = t = n = 4$, $c = 1$.

Fig. 2 shows $\pi(i)$ as a function of i for $t = n = 4$, $c = 1$, and different overlay sizes and values of d as obtained by the mathematical model (i.e., the BCA policy). The value of the threshold for the FEC(4,3) code is $p_{max} = 0.129$. The figure shows that when the overlay is stable (e.g., $\forall d, N$ at $p = 0.06$), neither its size nor d has a significant effect on $\pi(i)$. However, in the unstable state ($d = 2, N = 50000$ at $p = 0.12$; $\forall d, N$ at $p = 0.14$) both increasing N and increasing d decrease $\pi(i)$, as the number of layers in the overlay increases in both cases.

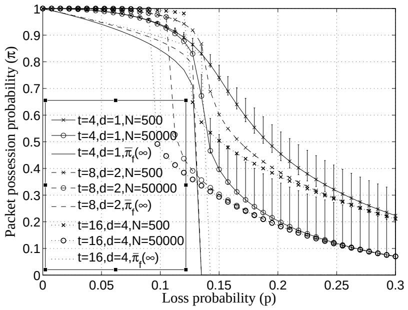


Fig. 3. π vs. p for $m = 4$, $n = t$, $k/n = 0.75$.

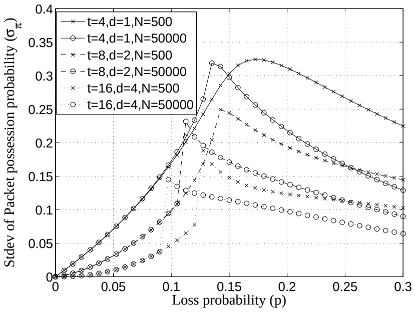
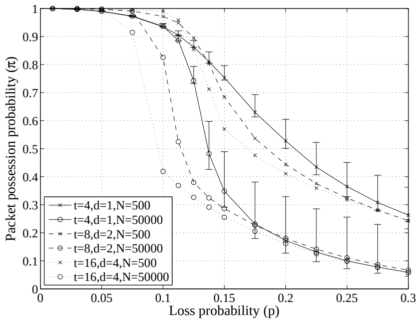


Fig. 4. σ_π vs. p for $m = 4$, $n = t$, $k/n = 0.75$.

Fig. 3 shows π as a function of p obtained with the mathematical model for $m = 4$ and $n = t$. The vertical bars show the values $\pi(1)$ at the upper end and $\pi(L-1)$ at the lower end. We included them for $d = 1$ only to ease readability, but they show the same properties for other values of d as well. The figure shows that π remains high and is unaffected by N and d as long as the overlay is stable. It drops however once the overlay becomes unstable, and the drop of the packet possession probability gets worse as the number of nodes and hence the number of layers in the overlay increases. At the same time the difference between $\pi(1)$ and $\pi(L-1)$ (the packet possession probability of nodes that are fertile in the first and the penultimate layers, respectively) increases. Furthermore, increasing t (and hence n) increases π in a stable system, but the stability region gets smaller and the drop of the packet possession probability gets faster in the unstable state due to the longer FEC codes. The curves corresponding to $\overline{\pi}_f(\infty)$ show the value of the asymptotic bound calculated using (4).

Due to the ungraceful degradation of π it is difficult to maintain the stability of the overlay in a dynamic environment by measuring the value of π. Hence, we look at the standard deviation of the packet possession probability. Fig. 4 shows the standard deviation σ_π as a function of p obtained with the mathematical model for $m = 4$ and $n = t$. The standard deviation increases rapidly even for low values of p and reaches its maximum close to p_{max}. Increasing the value of t decreases the standard deviation of π, i.e., the number of packets received in a block varies less among the nodes of the overlay. Its quick response to the increase of p makes σ_π more suitable for adaptive error control than π itself.

To validate our model we first present simulation results for the BCA policy. Figs. 5 and 6 show π and σ_π respectively as a function of p for the same scenarios as Figs. 3 and 4. Both figures show a good match with the analytical model and confirm that the increase of the standard deviation is a good indicator of the increasing packet loss probability. For long FEC codes the simulation results show slightly worse performance close to the stability threshold compared to the analytical results. The difference is due to late arriving packets, i.e., FEC reconstruction is not possible within the time limit set by the playout buffer's size.



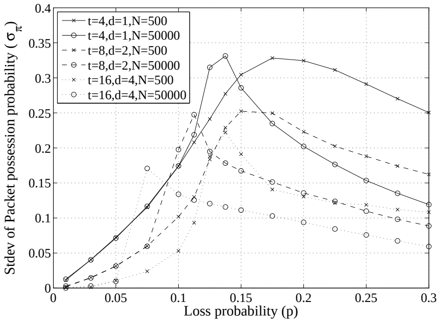
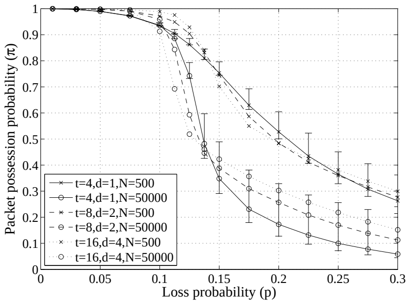


Fig. 5. π vs. p for $m = 4$, $n = t$, $k/n = 0.75$. BCA policy, simulation results.

Fig. 6. σ_π vs. p for $m = 4$, $n = t$, $k/n = 0.75$. BCA policy, simulation results.



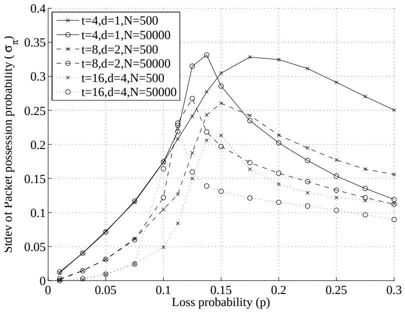


Fig. 7. π vs. p for $m = 4$, $n = t$, $k/n = 0.75$. UCA policy, simulation results.

Fig. 8. σ_π vs. p for $m = 4$, $n = t$, $k/n = 0.75$. UCA policy, simulation results.

To see the effects of the capacity allocation policy, we show π as a function of p in Fig. 7 for the same scenarios as in Fig. 5 but for the UCA policy. Comparing the figures we see that π is the same in the stable state, but is higher in the unstable state of the overlay. Furthermore, the region of stability is wider. Comparing the results for σ_π shown in Fig. 8 we can observe that for $\overline{N} = 50000$ and $d > 1$ the standard deviation is somewhat higher compared to the BCA policy and is similar in shape to the $d = 1$ case. This and the wider region of stability using the UCA policy is due to that the overlay has less layers than using the BCA policy as it is shown in Fig. 9. The figure shows the cumulative distribution function (CDF) of the layer where the nodes of the overlay are in their sterile trees.

There is practically no difference between the distributions for $d = 1$ and the BCA policy with $d > 1$ for the same t/d value. With the UCA policy nodes tend to have more children in the fertile tree where they are closest to the root due to the parent selection algorithm, so that the trees' structure is similar to the $d = 1$ case and the number of layers is lower than using BCA (compare the results for $t = 16$, $d = 1$ vs. $t = 16$, $d = 4$, UCA vs. $t = 16$, $d = 4$, BCA).

Hence, the data distribution performance of an overlay with t trees and $d = 1$ can be closely resembled with an overlay with $d > 1$ and td trees by employing a (centralized or distributed) mechanism that promotes parents close to the root, such as the UCA policy. Doing so allows the use of longer FEC codes, hence better performance in the stable region but a similar stability region due to the lower number of layers. At the same time one can decrease the probability that a node fails to reconnect to the overlay after the departure of a parent node [16]. Longer FEC codes decrease the variance of the packet possession probability and allow a smoother adaptation of the redundancy as a function of the measured network state.

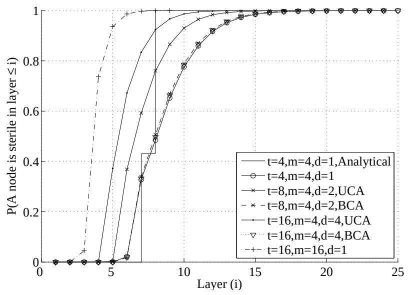


Fig. 9. CDF of the layer where nodes are sterile for $N = 50000$. Simulation results.

5 Conclusion

In this paper, we analyzed a peer-to-peer live streaming solution based on multiple transmission trees, FEC and free allocation of the output bandwidth of the peers across several trees. The aim of this design is to avoid tree disconnections after node departures, which can happen with high probability in resource scarce overlays if all the peers can forward data in one tree only.

We presented a mathematical model to express the packet possession probability in the overlay for the case of independent losses and the BCA policy. We determined the stability regions as a function of the loss probability between the peers, of the number of layers and of the FEC block length, and analyzed the asymptotic behavior of the overlay for a large number of nodes. We calculated the variance of the packet possession probability to study the overlay around the stability threshold. We concluded that the variance increases significantly with the packet loss probability between the peers and consequently it is a good performance measure for adaptive forward error correction. It will be subject of future work to design a robust stabilizing controller that can maintain a target packet possession probability in a dynamic environment.

We concluded that as long as the overlay is stable, the performance of the data transmission is not influenced by the number of fertile trees and the allocation policy, while longer FEC codes improve it. Increasing the number of fertile trees decreases however the packet possession probability in the overlay in the unstable region due to longer transmission paths. Nevertheless, with the UCA policy one can increase the number of trees, that of the fertile trees and the FEC block length, while the performance can be close to that of the minimum depth trees, because the UCA policy leads to shallow tree structures. These results show that adjusting the number of fertile trees can be a means to improve the overlays' stability without deteriorating the performance of the data distribution.

References

1. Liao, X., Jin, H., Liu, Y., Ni, L., Deng, D.: Anysee: Scalable live streaming service based on inter-overlay optimization. In: Proc. of IEEE INFOCOM. (April 2006)
2. Bishop, M., Rao, S., Sripanidkulchai, K.: Considering priority in overlay multicast protocols under heterogeneous environments. In: Proc. of IEEE INFOCOM. (April 2006)
3. Cui, Y., Nahrstedt, K.: Layered peer-to-peer streaming. In: Proc. of NOSSDAV. (2003) 162–171
4. Cui, Y., Nahrstedt, K.: High-bandwidth routing in dynamic peer-to-peer streaming. In: Proc. of ACM APPMS. (2005) 79–88
5. Tan, G., S.A., J.: A payment-based incentive and service differentiation mechanism for peer-to-peer streaming broadcast. In: Proc. of IEEE IWQoS. (2006) 41–50
6. Sung, Y., Bishop, M., Rao, S.: Enabling contribution awareness in an overlay broadcasting system. In: Proc. of ACM SIGCOMM. (2006) 411–422
7. Guo, M., Ammar, M.: Scalable live video streaming to cooperative clients using time shifting and video patching. In: Proc. of IEEE INFOCOM. (2004)
8. Banerjee, S., Lee, S., Braud, R., Bhattacharjee, B., Srinivasan, A.: Scalable resilient media streaming. In: Proc. of NOSSDAV. (2004)
9. Setton, E., Noh, J., Girod, B.: Rate-distortion optimized video peer-to-peer multicast streaming. In: Proc. of ACM APPMS. (2005) 39–48
10. Padmanabhan, V., Wang, H., Chou, P.: Resilient peer-to-peer streaming. In: Proc. of IEEE ICNP. (2003) 16–27
11. Castro, M., Druschel, P., Kermarrec, A., Nandi, A., Rowstron, A., Singh, A.: SplitStream: High-bandwidth multicast in a cooperative environment. In: Proc. of ACM SOSP. (2003)
12. Dán, G., Fodor, V., Karlsson, G.: On the asymptotic behavior of end-point-based multimedia streaming. In: Proc. of Internat. Zürich Seminar on Communication. (2006)
13. Dán, G., Fodor, V., Karlsson, G.: On the stability of end-point-based multimedia streaming. In: Proc. of IFIP Networking. (May 2006) 678–690
14. Dán, G., Chatzidrossos, I., Fodor, V., Karlsson, G.: On the performance of error-resilient end-point-based multicast streaming. In: Proc. of IWQoS. (June 2006) 160–168
15. Sripanidkulchai, K., Ganjam, A., Maggs, B., Zhang, H.: The feasibility of supporting large-scale live streaming applications with dynamic application end-points. In: Proc. of ACM SIGCOMM. (2004) 107–120
16. Dán, G., Chatzidrossos, I., Fodor, V.: On the performance of multiple-tree-based peer-to-peer live streaming. In: Proc. of IEEE INFOCOM. (May 2007)
17. Reed, I., Solomon, G.: Polynomial codes over certain finite fields. SIAM J. Appl. Math. **8**(2) (1960) 300–304
18. Zegura, E.W., Calvert, K., Bhattacharjee, S.: How to model an internetwork. In: Proc. of IEEE INFOCOM. (March 1996) 594–602
19. Veloso, E., Almeida, V., Meira, W., Bestavros, A., Jin, S.: A hierarchical characterization of a live streaming media workload. In: Proc. of ACM IMC. (2002) 117–130
20. Sripanidkulchai, K., Maggs, B., Zhang, H.: An analysis of live streaming workloads on the Internet. In: Proc. of ACM IMC. (2004) 41–54
21. Le Boudec, J.Y., Vojnovic, M.: Perfect simulation and stationarity of a class of mobility models. In: Proc. of IEEE INFOCOM. (March 2004)