An Analytical Study of Low Delay Multi-tree-based Overlay Multicast

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

ABSTRACT

In this paper we propose an analytical model that describes the temporal evolution of the end-to-end loss characteristics for live multicast streaming. We consider push-based architectures combined with retransmissions and forward error correction (FEC). We use the model to identify the primary sources of delay in overlay multicast, and to investigate the possible ways of decreasing the required playback delay. Based on the results we argue that in order to achieve good quality with low playback delays independent of the overlay's size, these systems have to adjust the FEC code rate dynamically. Our findings show that the available upload capacity is the key to efficient overlay multicast with low delay bounds.

Categories and Subject Descriptors

C.4 [**Performance of Systems**]: Modeling techniques

General Terms

Performance

Keywords

Modeling, Overlay multicast, Delay, Data distribution performance

1. INTRODUCTION

Overlay multicast is considered to be a promising means for distributing streaming content simultaneously to a large population of users. Its success will depend on its ability to provide data transmission with *low delay and information loss*. In such systems peers have to relay data with low delay, so that the possibilities of error recovery are limited. Consequently, the main problem to be dealt with is the propagation and thus the accumulation of losses, which results in low perceived quality for peers far from the source.

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

P2P-TV'07, August 31, 2007, Kyoto, Japan.

The architectures proposed for peer-to-peer streaming generally fall into one of two categories: push based or pull based [1]. Solutions in both categories utilize multi-path transmission. Multi-path transmission offers two advantages. First, disturbances on an overlay path lead to graceful quality degradation in the nodes. Second, the output bandwidth of the peers can be utilized more efficiently.

Several works deal with the management of such overlays ([2, 3] and references therein). There are also numerous proposals on how to improve the robustness of the overlays to errors using coding techniques such as forward error correction (FEC) and multiple description coding (MDC) [1]. The evaluation of the proposed solutions is however mostly based on simulations and small scale measurements; the analytical modeling of overlay multicast has not received much attention. We argue that if overlay multicast will ever become successful, population sizes will exceed those considered in the literature in simulation and experimental studies, and hence there is a need for an analytical understanding of the performance and the scalability of overlay multicast systems.

Models that describe the data distribution performance of multi-tree-based overlays were first proposed in [4, 5, 6] and showed that if forward error correction is the only means of resilience then these systems exhibit a phase-transition: the performance degrades ungracefully as the overlay's size or the loss probability reaches a threshold value. An approximate model was used in [7] to give insight into the temporal evolution of the data distribution performance. The effect of the forwarding capacity on multi-tree-based overlays was investigated in [8] using a queuing theoretic approach, and in [9] based on a fluid model. In [10] the authors derived a bound on the required playback delay for a pull based overlay assuming a complete graph and error free transmission. We are however not aware of any general model of the effects of the playback delay on the performance of multi-tree-based overlay multicast in the presence of losses, retransmissions and FEC.

This paper makes two important contributions. First, it presents an exact model of the temporal evolution of the data distribution in overlay multicast in the presence of losses, retransmissions and FEC. Second, it identifies the key factors that influence the required playback delay in overlay multicast, hence the minimum zapping delay, and discusses the possible ways of minimizing it.

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

2. SYSTEM DESCRIPTION

The overlay consists of a root node and N peer nodes. The peer nodes are organized in t distribution trees. Each peer node is mem-

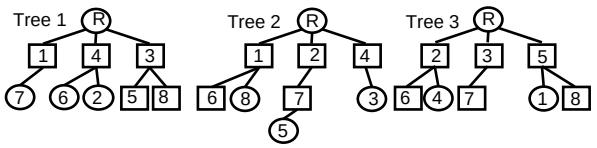

Figure 1: Overlay with $t = 3$, $m = 3$, $d = 2$, $N = 8$. Square indicates that the node is fertile.

ber of at least one tree, and in each tree it has a different parent node from which it receives data. We say that a node that is l hops away from the root node in tree e is in layer l of tree e. We denote the maximum number of children of the root node in each tree by m, and we call it the multiplicity of the root node.

Each node can have children in up to d of the t trees (d is a system parameter), called the fertile trees of the node. The node is called sterile in the other $t-d$ trees. If a node r has enough capacity and is willing to forward data to γ^r children then we say that the node has a total of γ^r cogs. For $d > 1$ the nodes balance their cogs between trees, i.e., a node can have up to $\lceil \gamma^r/d \rceil$ cogs in each of its fertile trees. We denote the number of children of a node by $\Gamma^r \leq \gamma^r$. We call an overlay well-maintained if the number of fertile nodes is maximal in every layer of its trees. Well-maintained overlays have the smallest depth for given N, t and d. For instance, in a well-maintained overlay with L layers, each node is $1 \leq l \leq L$ hops away from the root node in its fertile trees, and $L-1 \leq l \leq L$ hops away in its sterile trees.

One gets the minimum breadth trees described in [11] for $d = t$, and the minimum depth trees evaluated in [2, 11, 12] for $d = 1$. The case $1 < d < t$ was proposed in [8] to improve the overlay's stability under churn. Fig. 1 shows an overlay for $N = 8$, $t = 3$, $m = 3$ and $d = 2$.

Tree management: The purpose of the tree maintenance algorithm (centralized [11] or decentralized [12, 13]) is to find eligible parents for the nodes (arriving nodes, preempted nodes and nodes disconnected due to the departure of a parent) based on the parent selection criteria, such as closeness to the root and the priorities of the nodes. 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 its structure, rather than the efficiency of the tree maintenance algorithm. In Section 4 we briefly describe the tree maintenance algorithm used for the simulations.

Data transmission and error resilience: We denote the stream's bitrate by B, and the average packet size by a. The root splits the data stream into n stripes, with every n^{th} packet belonging to the same stripe, and it sends the packets at round-robin to its children in the different trees. Peer nodes relay the packets upon reception to their respective child nodes. We consider two means of error resilience: retransmissions and FEC.

Retransmissions are efficient if the loss of a packet can be detected quickly, and if the retransmission request is sent to a node that is present in the overlay and is in hold of the packet. The excess bandwidth used by retransmissions is proportional to the loss probability, and is difficult to predict.

Block based FEC, e.g., Reed-Solomon codes, is used by the root: it adds c redundant packets to every k packets, resulting in a block length of $n = k+c$. We denote this FEC scheme by FEC(n,k). Once a node receives at least k packets of a block of n packets, it may recover the remaining c packets. If a packet belonging to a fertile tree is recovered, then it is sent to the respective children. Duplicate packets are discarded by the nodes. If the root would like to increase the ratio of redundancy while maintaining its bitrate unchanged, then it has to decrease the source rate.

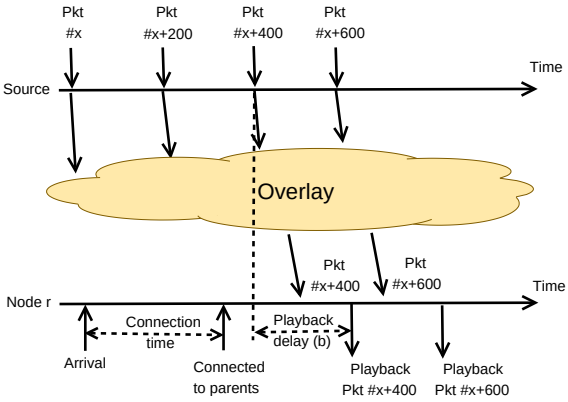

Figure 2: The playback delay and the connection time determine the minimum zapping delay in overlay multicast.

Playback delay We define the playback delay b as the lag between the time of the generation of a packet at the root node and the time of the playback at the peers, as shown in Fig. 2. This lag does not depend on the time needed for a node to connect to the overlay. It is however affected by node churn (e.g., the frequency of node departures and the time needed to reconnect to the overlay), by the node's distances from the root and by packet losses. It is the effect of these factors that we aim to capture in the model. The zapping delay does not have to be more than the sum of the playback delay and the time needed to connect to the overlay.

3. DATA DISTRIBUTION MODEL

We quantify the performance of the data distribution via the probability $\pi(b)$ that an arbitrary node receives or can reconstruct (i.e., possesses) an arbitrary packet in the overlay within the playback delay b. If we denote by $X_r(b)$ the number of packets possessed by node r in an arbitrary block of packets, then $\pi(b)$ can be expressed as the average ratio of packets possessed in a block over all nodes, i.e., $\pi(b) = E[\sum_r X_r(b)/n/N]$.

We model the behavior of the overlay in the presence of independent packet losses and retransmissions. We introduce the random variable D_d, the time it takes for a packet to travel between two nodes, given by its distribution function $F_d(h)$, and probability density function $f_d(h) = \frac{\partial}{\partial h} F_d(h)$. The model builds on the simplifying assumption that the probability that a node is in possession of a packet is independent of whether another node in the same layer is in possession of a packet. For brevity, we show equations for the case when n is a multiple of t, and $\gamma^r \geq t$ is equal for all nodes. Let us denote by L the number of layers in the overlay. We assume that nodes are in the same layer in their fertile trees, and in their sterile trees respectively, and we introduce L_l the layer where a node that is fertile in layer l is located in its sterile trees. Typically, $L-1 \leq L_l \leq L$. We will comment on the possible effects of our assumptions and on the possible extensions of the model in Section 3.3.

The key to the performance of the overlay is the probability $\rho_{j,l}(h)$ that a node in layer l receives an arbitrary packet of stripe j no later than h time *after the first packet of the block it belongs to is ready to be sent out from the root*. Let us introduce the binary random variable $R_{j,l}(h)$, such that $P(R_{j,l}(h) = 1) = \rho_{j,l}(h)$. Fig. 3 illustrates $\rho_{j,l}(h)$ and $R_{j,l}(h)$ in an overlay with $t = 4$, $n = 4$ and two layers.

In the following we present a system of algebraic and differen-

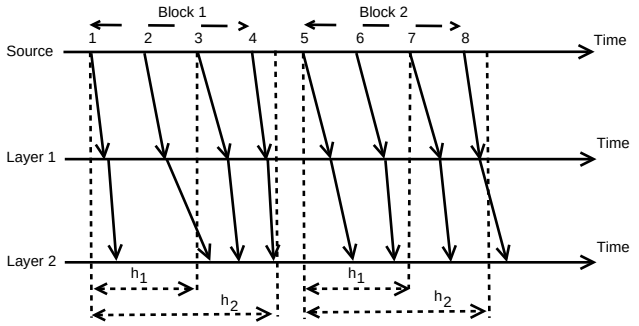

Figure 3: $\rho_{j,l}(h)$ **and** $R_{j,l}(h)$ **for** h_1 **and** h_2 **and two blocks of data.** $\rho_{2,2}(h_1) = 0.5$**,** $\rho_{2,2}(h_2) = 1$**,** $\rho_{4,2}(h_1) = 0$**,** $\rho_{4,2}(h_2) = 0.5$**.**

tial equations of convolution type that describes the evolution of this probability. The probability that nodes receive data from other nodes is determined by the probability that a node that forwards data in a tree can forward the data to its children. Hence, we introduce the probabilities $\pi^f_{j,l}(h)$ that a node that is in layer l in its fertile tree, possesses an *arbitrary packet* in stripe j no later than h. The evolution of the probability of packet reception in layer l ($1 \le l \le L$) and stripe j ($1 \le j \le n$) is described by

$$\frac{\partial \rho_{j,l}(h)}{\partial h} = \int_0^h \frac{\partial \pi^f_{j,l-1}(h-\upsilon)}{\partial h} f_d(\upsilon) d\upsilon. \quad (1)$$

The probability of packet possession at time h for stripe j depends on the packet reception probability and the possibility of reconstruction using FEC. A node in layer l possesses a packet of stripe j by time h either if it receives the packet by time h (i.e., $R_{j,l}(h) = 1$) or if it can reconstruct it using the packets received in the other stripes, i.e., it receives at least k out of the remaining $n-1$ packets,

$$\pi^f_{j,l}(h) = \rho_{j,l}(h) + (1-\rho_{j,l}(h)) P(\sum_{i \ne j} R_{i,l_i}(h) \ge k), \quad (2)$$

where $l_i = l$ for stripes in which the node is fertile and $l_i = L_l$ for stripes in which the node is sterile.

The initial condition of the problem is given by the time packets are ready to be sent out from the root node. If the packets of an FEC block are sent out smoothed over na/B time then

$$\pi^f_{j,0}(h) = H(h-(j-1)a/B), \quad (3)$$

where $H(\cdot)$ is the unit step function.

We solve the above system of differential-algebraic equations numerically in an iterative way. For playback delay b the value of $\rho_{j,l}(h)$ has to be evaluated for $h \le b + (n-1)a/B$.

Based on the probabilities $\pi^f_{j,l}(h)$ we can express $\pi_{j,l}(b)$ ($1 \le l \le L$), the probability that a node that is in layer l in the tree where stripe j is distributed possesses an arbitrary packet before its playout deadline given the playback delay b. The playout deadline for a packet in stripe j is $h_j = b + (j-1)a/B$, so that

$$\pi_{j,l}(b) \quad = \quad \rho_{j,l}(h_j) + (1-\rho_{j,l}(h_j)) P(\sum_{i \ne j} R_{i,l_i}(h_j) \ge k)$$

The probability that an arbitrary node possesses a packet is

$$\pi(b) = \frac{1}{n} \sum_{j=1}^{n} \frac{1}{N} \sum_{l=1}^{L} \pi_{j,l}(b) N_l, \quad (4)$$

where N_l is the number of nodes in layer l of the overlay.

The computational complexity of the calculation is $O(L|f_d|)$, where $|f_d|$ is the length of the vector used to approximate the p.d.f. of D_d. As L is $O(logN)$ in the considered overlays, the algorithm scales well with the number of nodes in the overlay.

For any $F_d(h)$ for which $lim_{h\to\infty} F_d(h) < 1$, the analysis of the asymptotic behavior presented in [6] with respect to N, p and the FEC code rate applies to $lim_{b\to\infty}\pi(b)$. If $lim_{h\to\infty} F_d(h) = 1$ then $lim_{b\to\infty}\pi(b) = 1$.

3.1 Approximating the overlay structure:

The number of fertile nodes in layer l of a well-maintained tree can be approximated by the recurrence $N_l = \sum_{r\in\mathcal{R}(l-1)} \Gamma^r/d$ with initial condition $N_1 = min(N/(t/d), m)$, where $\mathcal{R}(l-1)$ denotes the set of nodes fertile in layer $l-1$. The overlay's actual structure differs from this approximation due to node dynamics, but as our simulation results show, the difference does not have a significant effect on the accuracy of our model.

3.2 Path delay model

For at most r retransmission attempts and loss probability p we calculate the distribution of D_d as

$$F_d(h) = (1-p) \sum_{i=0}^{r} p^i P(D_{ph} + iD_{ret} < h), \quad (5)$$

where D_{ph} is the per-hop-delay in the forward direction and D_{ret} is the round trip time of a retransmission. We disregard the time needed to detect packet loss: not because it would be negligible but because it is implementation dependent. Consequently, our results represent the best case scenario for retransmissions.

The per-hop-delay consists of four components, the queuing delay on the output link of the source node (with capacity C_{out}), the propagation and queuing delays on the paths between the nodes, the queuing delay on the input link of the destination node (with capacity C_{in}) and the processing delay in the nodes

$$D_{ph} = D_{tr,o} + D_p + D_{tr,i} + D_{pr}. \quad (6)$$

For the considered block lengths and common streaming bitrates the processing delays D_{pr} (e.g., arithmetic operations for Reed-Solomon coding) are negligible compared to other sources of delay, and are not considered in this paper.

The queuing delay on the input link of a node with input bandwidth C_{in} can be expressed as

$$D_{tr,i} = W_{in} + a/C_{in} \ge a/C_{in}, \quad (7)$$

where W_{in} is the waiting time of a packet in the input link's buffer, and a/C_{in} is its transmission time. The input link's buffer can be modeled by a G/D/1 queue (assuming constant packet sizes), and the delay can be negligible if the nodes' input capacities are much higher than the stream's bandwidth.

We can model the output queue as a $GI^X/D/1$ queue with batch arrivals of constant size Γ^r/d [14]. *The queuing delay on the output link* of a node with upload bandwidth C_{out} is

$$D_{tr,o} = W_{out} + Ia/C_{out}, \quad (8)$$

where W_{out} is the waiting time of the first packets of the arriving batches of packets in the output buffer of the node, and I is a random variable with discrete uniform distribution on $[1, \Gamma^r/d]$. If we denote by u the link's utilization, i.e., $u = \Gamma^r/(tC_{out}/B)$, then we can rearrange (8) to

$$D_{tr,o} = W_{out} + uId/\Gamma^r at/B. \quad (9)$$

Bandwidth resources (sometimes measured with the resource index $\overline{C}_{out}/B$, where $\overline{C}_{out}$ is the average output capacity [15]) are usually

scarce in overlay multicast, hence u has to be high to maintain the overlay feasible, and $D_{tr,o}$ is potentially an important source of delay.

The round-trip-time of a retransmission is modeled as the sum of the propagation times and the queuing times as seen by a random arrival in the corresponding queues

$$D_{ret} = D^*_{tr,o} + D_p + D^*_{tr,i} + D^*_{tr,o} + D_p + D^*_{tr,i}, \qquad (10)$$

e.g., $D^*_{tr,o}$ is the sum of the remaining work as seen by a random arrival in a $GI^X/D/1$ queue and the transmission time of the packet, a/C_{out}. For reasonable loss probabilities and modest link utilization the influence of the retransmissions on the arrival processes at the input and the output links is negligible, hence we do not model it.

3.3 Discussion of the assumptions

In the following we discuss the validity of certain assumptions made in the model. The model can be extended to include heterogeneous losses by following the procedure presented in [4] for the minimum breadth trees. The effects of nodes with heterogeneous input and output bandwidths can be included in the model in a similar way. We decided to show equations for the homogeneous case here to ease understanding, though we show results for heterogeneous output bandwidths. It is not clear yet how the model can be extended to correlated losses without increasing its complexity. Nevertheless, the effect of correlations was evaluated in [7], so we dispense with its analysis in this paper.

Our results for block based FEC apply to PET and the MDC scheme considered in [11], 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.

Following the arguments presented in [16], the effects of node departures on an overlay that employs FEC can be incorporated in the model in the following way. Let us denote by κ the ratio of the average time before the departure of a parent node and the average time to find a new parent as seen by a node. Furthermore, we denote by α the ratio of the average time before the departure of a parent node and the average node lifetime. If nodes have i parents upon their arrival then the average ratio of their disconnected parents as seen by a random observer is

$$E[\Delta_i] = \frac{t + i\alpha}{t(\kappa + \alpha + 1)}. \qquad (11)$$

One can then use $p = E[\Delta_i]$ in the model to estimate the overlay's performance in the presence of node churn. Simulation results in [16] show the accuracy of this approach for FEC. For retransmissions, the distribution of the time to find a new parent influences the distribution of the retransmission time, unless a list of backup parents is maintained in every node. Consequently, our results represent an upper bound for the performance of retransmissions in the presence of node dynamics.

4. PERFORMANCE EVALUATION

In the following we first describe the simulation methodology then we present results obtained via the model and validate them via simulations.

4.1 Simulation methodology

We developed a packet-level event-driven simulator and used the GT-ITM topology generator [17] to generate a transit-stub network with 10^4 nodes. 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 (mean 67 ms, standard deviation 21 ms, maximum 180 ms). The delay between overlay nodes residing on the same node of the topology was set to 1 ms. The inter-arrival times of nodes are exponentially distributed, this assumption is supported by several measurement studies, e.g., [18]. The session holding times M follow the log-normal distribution, the mean holding time is $E[M] = 306$ s [18].

Tree maintenance: We assume that a distributed algorithm, such as gossip based algorithms, is used by the nodes to learn about other nodes, and that it provides random knowledge of the overlay such as in [15]. When a node wants to join the overlay, it contacts the root and obtains a random list of $g = 100$ members of every tree. The root tells the arriving node in which trees it should forward data: in the ones with the least amount of forwarding capacity. The arriving node then uses the following parent selection procedure to find a parent.

To select a parent in a tree, the node sorts the g members it is aware of into increasing order according to their distances from the root, and looks for the first node that has available capacity or has a child that can be preempted, i.e., which has lower priority. We consider two priority schemes: fertile nodes can preempt sterile nodes in the NP scheme [12]; nodes with more cogs can preempt others in the P scheme [19]. 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.

Unlike [15, 20], 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.

Data distribution: The stream consists of 1410 bytes long packets. The nodes have a playout buffer that can hold 150 packets. Every node has an input and an output buffer of 80 packets each to absorb the bursts of incoming and outgoing packets. We simulate independent packet losses on the input links of the nodes. To measure $\pi(b)$, for every node we record the portion of packets that it possesses b time after they were sent out from the root.

To obtain the results for a given overlay size $\overline{N}$, we start the simulation with $\overline{N}$ nodes in its steady state as done in [6, 8]. 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 tree structure. The measurements are made after the warm-up period during 1000 s and the presented results are the averages of 10 simulation runs. The results have less than 5 percent margin of error at a 95 percent level of confidence.

4.2 Numerical results

We consider the streaming of a $B = 112.8$ kbps data stream, and the capacity of the root node's output link is 10 Mbps, unless otherwise stated. We consider $m = 50$ throughout the paper for easy comparison, though the particular value of m does not affect the validity of our conclusions. We chose to use a high value in order to keep the effects of tree disconnections low in the simulations according to [8].

We consider three scenarios with different utilizations of the input and the output links. In the first scenario ("inf.cap.") the input and output link capacities are $C_{in} = C_{out} = 10$ Mbps (the number of cogs per node is still t), and consequently the per-hop-delay is determined by the propagation delays. In the second scenario ("inf.incap.") the input link capacities are $C_{in} = 10$ Mbps, the output link capacities are $C_{out} = 128$ kbps, i.e., close to the stream's

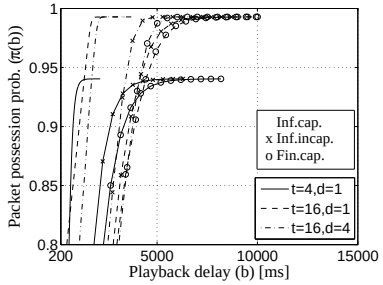

Figure 4: $\pi(b)$ **vs.** b **for** $\overline{N} = 10000$**,** $n = t$**,** $k/n = 0.75$**,** $p = 0.1$**. Deterministic arrivals.**

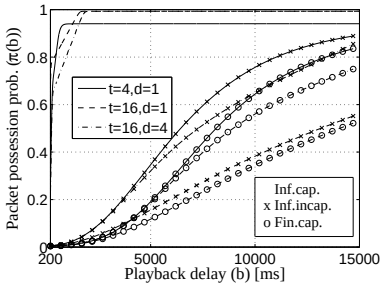

Figure 5: $\pi(b)$ **vs.** b **for** $\overline{N} = 10000$**,** $n = t$**,** $k/n = 0.75$**,** $p = 0.1$**. Poisson arrivals.**

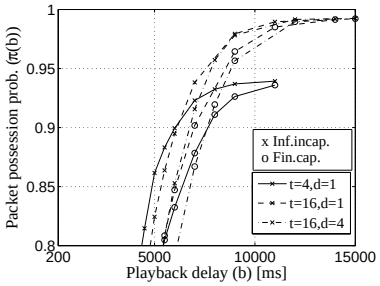

Figure 6: $\pi(b)$ **vs.** b **for** $\overline{N} = 10000$**,** $n = t$**,** $k/n = 0.75$**,** $p = 0.1$**. Simulation results.**

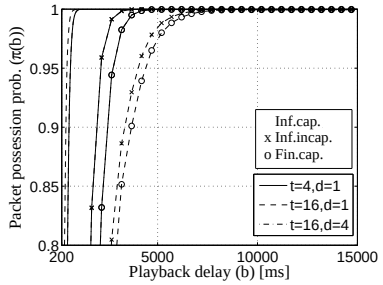

Figure 7: $\pi(b)$ **vs.** b **for** $\overline{N} = 10000$**,** $p = 0.1$**. Retransmissions and deterministic arrivals.**

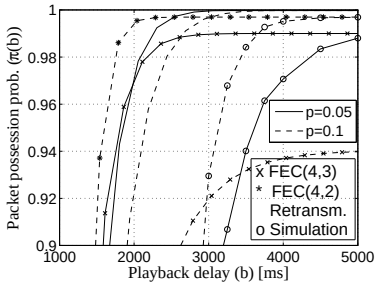

Figure 8: $\pi(b)$ **vs.** b **for** $\overline{N} = 10000$**,** $t = 4$**. Deterministic arrivals and simulations for FEC.**

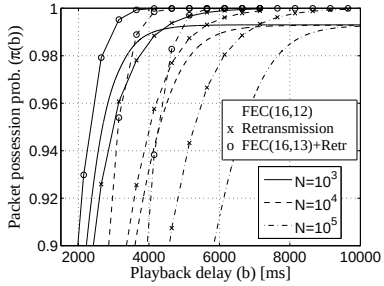

Figure 9: $\pi(b)$ **vs.** b **for** $t = 16$**,** $p = 0.1$ **and various overlay sizes. Deterministic arrivals.**

bitrate. In the third scenario ("fin.cap.") both the input and the output links' capacities are $C_{out} = C_{in} = 128$ kbps.

To see how the packet arrival process at the nodes affects the results, we consider two cases for the analytical model. First, the best case scenario, when the arrival processes are deterministic both on the input and on the output links of the peers. In this case $P(W_{in} = 0) = 1$ and $P(W_{out} = 0) = 1$ in (7) and (8) respectively. Furthermore, if Γ^r is proportional to the nodes' output bandwidths [2], i.e., u is equal for all nodes, the expected value of the second term on the right hand side of (9) is *independent of the distribution of the nodes' output bandwidths*. Figure 4 shows $\pi(b)$ as a function of b for different tree structures at $p = 0.1$. The effect of the propagation delay on the required playback delay is small compared to the effects of the output capacities, and decreases as t increases. It increases however as the output link capacities increase, hence in high-bandwidth overlays ($E[D_p] \approx at/\overline{C}_{out}$) proximity based neighbor selection can decrease the required playback delay.

Second, we consider Poisson arrivals. Figure 5 shows $\pi(b)$ as a function of b for the same scenarios as Fig. 4. We observe a significant deterioration of the streaming performance due to the waiting times.

Fig. 6 shows simulation results for the "fin.cap" and the "inf.incap" scenarios. The results closely match the analytical results for the deterministic arrival process. We conclude that the packet arrival process is regular even close to the stability threshold ($p = 0.129$ for FEC(4,3), see e.g. [6]) and for $u \approx 1$. In the following we show analytical results obtained for the deterministic arrival process, as it gives a reasonable match with the simulation results.

Fig. 7 shows analytical results for an overlay in which retransmissions are used for error control. The required playback delay using retransmission is not significantly lower than using FEC with $k/n = 0.75$, and shows similar behavior. Though this result suggests that retransmission is more efficient than FEC for a given overhead, we should remember that the time needed for detection can significantly worsen the performance of retransmissions, especially in the case of node churn.

Since the effects of C_{in} on the results are small compared to those of C_{out}, in the following we only show results for the "inf.incap" scenario. Fig. 8 shows that a lower playback delay is sufficient for lower loss probabilities. By increasing the FEC redundancy one can decrease the required playback delay for a given loss probability, but relying only on retransmissions one cannot influence it. The simulation results show similar behavior as the analytical results: the adequacy of the deterministic arrival process depends – apart from the resource index – on how far the system is from the stability threshold discussed in [6].

The ability to control the required playback delay is important as well when the overlay's size increases, as shown in Fig. 9. The required playback delay increases with the number of nodes, i.e., the number of layers both for FEC and retransmissions. We also observe that the playback delay can be decreased by combining FEC and retransmissions. Consequently, retransmissions are not sufficient in order to maintain a constant playback delay in a growing overlay: the ratio of FEC redundancy has to be adjusted dynamically as the overlay's size or the loss probability (due to network failures or node churn) increases.

Fig. 10 shows the overlay's performance for $\overline{N} = 10^4$, $t = 4$, $C_{root} = 100$ Mbps for various output capacity distributions and cog allocations. In the case of homogeneous capacities (CH) $C_{out} =$ 256 kbps for all nodes; in the case of inhomogeneous capacities (CI) $C_{out} = 128\,kbps$ for 65 percent of the nodes, and $C_{out} = 512\,kbps$ for the rest of them, similar to measured distributions shown in [2]. We call min-max fair (MM) allocation when $\Gamma^r = \gamma^r = \lfloor tC_{out}/\overline{C}_{out} \rfloor$ (so that $u \approx B/\overline{C}_{out}$ for all nodes), i.e., nodes upload proportional to their upload capacities. We call full utilization (FU) allocation,

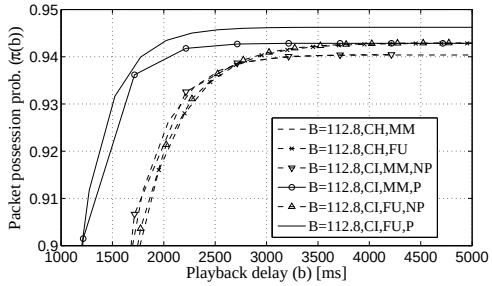

Figure 10: $\pi(b)$ vs. b for $\overline{N} = 10^4$, $p = 0.1$, FEC(4,3), and various allocations of the nodes' output capacity. Deterministic arrivals.

when $\Gamma^r \leq \gamma^r = \lfloor tC_{out}/B \rfloor$ (so that $0 \leq u \leq \gamma^r/(tC_{out}/B) \leq 1$), that is, some nodes contribute more upload capacity than their proportional share if there is abundant capacity in the overlay. P and NP stand for the prioritization schemes.

The results show that the required playback delay is stipulated by $\overline{C}_{out}$ of the *contributor* nodes. The FU allocation of the nodes' output capacities does not change the required playback delay significantly neither for CH nor for CI because it does not change the mean capacity of the contributor nodes; it only assigns more load to nodes close to the root. Nevertheless, prioritization decreases the required playback delay as it decreases the number of layers of the overlay for given $\overline{N}$ and output capacities. Less layers improve the overlay's stability when using FEC [6], but as shown, $\lim_{b\to\infty}\pi(b)$ is not increased much by prioritization. The FU allocation combined with prioritization performs best in the considered scenario: this combination can give considerable gains if a small subset of the nodes has high output capacity and is able to feed all nodes. For deterministic arrivals the source bitrate does not have a significant effect on the required playback delay. Nonetheless, close to $B = \overline{C}_{out}$ the arrival process is less regular, queues build up, hence the required playback delay increases. The simulations, not shown here for brevity, support these analytical results.

We conclude, that the ways to decrease the required playback delay are (i) decreasing the number of layers (by prioritization, FU allocation, and by increasing m as much as possible), (ii) using an adequate number of trees (though using a few trees only might imperil the stability of the overlay for given n,k,p [7]), (iii) dynamically adjusting the FEC redundancy, and (iv) using a bitrate not too close to $\overline{C}_{out}$.

5. CONCLUSION

In this paper, we presented a mathematical model to express the packet possession probability in multi-tree-based overlay multicast as a function of the playback delays of the peers. We identified the average available upload capacity at the nodes as the most important factor that influences the required playback delay in the overlay. The playback delay can be decreased by non-min-max fair allocation of the peers' forwarding capacities combined with prioritization, if bandwidth resources are abundant and the nodes' output capacities are inhomogeneous. Our evaluation shows that retransmissions and FEC have to be used together in order to achieve good quality overlay multicast with low playout delay: retransmissions decrease the FEC redundancy needed to maintain the stability and good performance of multi-tree-based overlay multicast. How to adjust the FEC parameters based on feedback from the peers, and how to extend the model to pull-based overlays will be subject of our future research.

6. REFERENCES

[1] V. Fodor and Gy. Dán, "Resilience in live peer-to-peer streaming," *IEEE Communications Magazine*, vol. 45, no. 6, June 2007.

[2] 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.

[3] X. Liao, H. Jin, Y. Liu, L.M. Ni, and D. Deng, "Anysee: Scalable live streaming service based on inter-overlay optimization," in *Proc. of IEEE INFOCOM*, April 2006.

[4] 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.

[5] Gy. Dán, I. Chatzidrossos, V. Fodor, and G. Karlsson, "On the performance of error-resilient end-point-based multicast streaming," in *Proc. of IWQoS*, June 2006, pp. 160–168.

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

[7] Gy. Dán and V. Fodor, "Modeling loss and delay in multi-tree-based overlay multicast," submitted to IEEE JSAC, Tech. rep. TRITA-EE 2007:016, March 2007.

[8] 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.

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

[10] L. Massoulie, A. Twigg, C. Gkantsidis, and P. Rodriguez, "Randomized decentralized broadcasting algorithms," in *Proc. of IEEE INFOCOM*, 2007.

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

[12] 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.

[13] 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.

[14] J. W. Cohen, *The Single Server Queue*, North-Holland Publishing, Amsterdam, 1969.

[15] 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.

[16] Gy. Dán and V. Fodor, "Understanding multiple-tree-based overlay multicast," School of Electrical Engineering, KTH, Tech. rep. TRITA-EE 2007:026, January 2007.

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

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

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

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