Analysing and Leveraging Client Heterogeneity in Swarming-based Live Streaming

Wasiur R. KhudaBukhsh*, Julius Rückert†, Julian Wulfheide†, David Hausheer†, and Heinz Koeppl*
* Bioinspired Communication Systems Lab (BCS), E-Mail: {wasiur.khudabukhsh | heinz.koeppl}@bcs.tu-darmstadt.de
† Peer-to-Peer Systems Engineering Lab (PS), E-Mail: {rueckert | julian.wulfheide | hausheer}@ps.tu-darmstadt.de,
Technische Universitaet Darmstadt, Germany

***Abstract*—Due to missing IP multicast support on an Internet scale, over-the-top media streams are delivered with the help of overlays as used by content delivery networks and their peer-to-peer (P2P) extensions. In this context, mesh/pull-based swarming plays an important role either as a pure streaming approach or in combination with tree/push mechanisms. The crucial impact of today's variety of client systems with their heterogeneous resources is not yet well understood. In this paper, we contribute to closing this gap by mathematically analysing the most basic scheduling mechanisms *latest deadline first* (LDF) and *earliest deadline first* (EDF) in a continuous time Markov chain framework and combining them into a simple, yet powerful, mixed strategy to leverage inherent differences in client resources. The contribution of this paper is, hence, twofold: (1) we develop a mathematical framework for swarming on random graphs with a focus on LDF and EDF strategies in heterogeneous scenarios; (2) we propose a mixed strategy, named SCHEDMIX, that leverages client heterogeneity. We show that SCHEDMIX outperforms LDF and EDF using different abstractions: a mean-field theoretic analysis of buffer probabilities, simulations of the stochastic model on random graphs, and a full-stack implementation of a P2P streaming system.**

I. INTRODUCTION

Media streaming dominates the traffic share on Internet. As new services are typically offered in an over-the-top (OTT) manner, they need to be efficient and scalable, without dependence on special network services. Because of its inherent limitations [3], IP multicast was not adopted in more than network islands and, in particular, is not usable for OTT content delivery. Instead, multicast functionality is realized at application layer in the form of content delivery networks (CDNs) and, to make delivery more profitable, peer-to-peer (P2P) mechanisms, or a combination of both [32]. In this work, we focus on live media streaming, an important application scenario with both high demand bandwidths and delays.

Over the years, different classes of P2P live streaming approaches were proposed [29], such as tree/push- and mesh/pull-based, as well as hybrid approaches. Due to their inherent robustness, mesh/swarming approaches continue to be of major importance, especially in hybrid settings where they often function as a substrate even when tree structures run on top of them [22], [24]. A key design issue in swarming is the data scheduling strategy used by individual peers to select chunks to be requested from their neighbours. Not only must it ensure continuous playback for an individual client, but also a healthy data replication to avoid content bottlenecks [20].

Several scheduling strategies of varying levels of complexity were proposed in the literature [29]. The impact of resource heterogeneity as observed in real client populations, however, is not yet fully understood. This leaves a big gap in the design space of practical P2P streaming approaches, where systematically leveraging resource imbalances could help simplifying complex scheduling strategies or designing new ones.

In this paper, we contribute to closing this gap by analysing the basic scheduling strategies *earliest deadline first* (EDF) and *latest deadline first* (LDF) based on a continuous time Markov chain framework. Our model can essentially be interpreted as a contact process [4], [15] on a random graph. An important facet of our framework is that it explicitly captures the degree-dependence of peers. Driven by the resulting analytic insights, we combine EDF and LDF into a simple, yet powerful, mixed strategy called SCHEDMIX to leverage differences in upload resources. Our theoretical efforts are complemented with a full-stack implementation of a P2P streaming system based on the SIMONSTRATOR [21] framework. The proposed strategy is shown to outperform the other two strategies using different abstractions: a mean-field theoretic analysis of buffer probabilities, simulation of the stochastic model, and discrete event-based simulation of the full-stack implementation. Thus we both theoretically and practically show the potential of using a combination of primitive scheduling mechanisms to improve overall performance in mesh-/pull-based media streaming. These results are encouraging to consider using primitive scheduling mechanism combinations in mesh-based as well as hybrid streaming and enable seamless switching (*transitions*) between them as proposed in [6].

The remainder of this paper is structured as follows: Section II presents the proposed mathematical framework, followed by our mean-field analysis in Section III. Subsequently, Section IV presents our findings from simulations of the stochastic process. Section V is devoted to the full-stack implementation. Finally, in Section VI we discuss related work and conclude our paper with a discussion in Section VII.

II. MODEL

A. The network

We describe the underlying network as a random graph. We assume the associated degree distribution has a finite mean. Let

ISBN 978-3-901882-83-8 © 2016 IFIP

$\mathcal{G}_M$ be the class of all simple and connected random graphs with M nodes. Let $\pi : \mathbb{N} \to [0,1]$ be the associated degree distribution. We also define the size-biased degree distribution, q as follows:

$$q(k) := \frac{k\pi(k)}{\sum_k k\pi(k)}, \tag{1}$$

for $k \in \mathbb{N}$. The quantity $q(k)$ is the probability that a given edge points to a vertex of degree k.

B. The peer-to-peer communication system

Suppose there are M peers and a single server. Let n denote the buffer length. The server uniformly selects a peer at random and uploads a chunk at buffer position 1. It continues to upload chunks to the chosen peer until there is a connection breakage/loss (an event that occurs with a small probability, say $\varepsilon \in (0,1]$) in which case the server chooses a peer again uniformly at random. The chunk at buffer position n, if available, is pushed for playback. After playback, the chunk is removed and all other chunks are shifted one index closer to playback. Each peer maintains a Poisson clock with rate proportional to its degree[1]. A peer, if not selected by the server, contacts one of its neighbours uniformly at random at each tick of its Poisson clock and seeks to download a missing chunk. The chunk it downloads from among all downloadable chunks is decided by its chunk selection strategy. For simplicity, we assume that the playback rate is one chunk per unit of time.

Let $\mathcal{G} := (\mathcal{V}, \mathcal{E}) \in \mathcal{G}_M$ be a given realisation of a random graph, where $\mathcal{V}$ and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ are the sets of vertices and edges, respectively. Each node is a peer. Let $\Omega := \{\omega \in \{0,1\}^{M\times n} \mid \sum_{i=1}^{M} \omega(i,1) = 1\}$ be the configuration space of all peers and buffers, and denote all subsets of Ω by $\mathcal{A}$. Define a continuous time Markov chain (CTMC) $\{X_t\}_{t\geq 0}$ on the measurable space $(\Omega, \mathcal{A})$ as $X_t(i,j) := 1$ if the j-th buffer location of the i-th peer is filled, and 0 otherwise. The rows of the matrix X_t, denoted as $X_t^1, X_t^2, \ldots, X_t^M$ represent buffer states of peers $1, 2, \ldots, M$, respectively.

Let $\mathbb{S} : \{0,1\}^{M\times n} \cup \{0,1\}^n \to \{0,1\}^{M\times n} \cup \{0,1\}^n$ denote the buffer shifting operator defined as $\mathbb{S}Y := (0, y_1, y_2, \ldots, y_{n-1})$ for $Y = (y_1, y_2, \ldots, y_n) \in \{0,1\}^{M\times n} \cup \{0,1\}^n$ where $y_1, y_2, \ldots, y_n$ denote the columns of Y.

Let us now define the transition rates of interaction for a node $v \in \mathcal{V}$ as follows

$$\mu^v(u, u+e_i) = \begin{cases} \displaystyle\sum_{l\in\mathcal{V}:(v,l)\in\mathcal{E}} \varsigma \mathbb{1}(X_t(l,i)=1)\alpha^v(i,u,X_t^l), & \\ & \text{if } i \neq 1, \\ \mathbb{1}(X_t(v,1)=1)(1-\varepsilon+\varepsilon/M) & \\ \quad +\mathbb{1}(X_t(v,1)=0)\varepsilon/M & \text{if } i = 1, \end{cases} \tag{2}$$

where $u = (u_1, u_2, \ldots, u_n) \in \mathcal{T} := \{0,1\}^n$, $i \in \mathcal{F} := \{1, 2, \ldots, n\}$, such that $u_i = 0$, $\varsigma > 0$ is a constant, $\mathbb{1}(.)$ is the indicator function, e_i is the i-th unit basis vector of the n-dimensional Euclidean space and $\alpha^v : \mathcal{F} \times \mathcal{T} \times \mathcal{T} \to [0,1]$ is the chunk selection function of the peer $v \in \mathcal{V}$. In words, $\alpha^v(i,u,X_t^l)\delta t$ is the probability of downloading chunk i when peer v is in buffer state u and contacts peer l in buffer state X_t^l. We defer an elaborate discussion of the chunk selection function to a later section. The system is described by the following *master equation*

$$\begin{aligned} \frac{d\mathrm{P}(X)}{dt} = &-\mathrm{P}(X) + \sum_{v'\in\mathcal{V}} \mathbb{1}(X(v',1)=1)\Big[\sum_{Y\in\Omega:\mathbb{S}Y=X-\Delta(v',1)} \\ &\mu^{v'}(Y^{v'}, Y^{v'}+e_1)\Big\{\mathrm{P}(Y) \\ &+ \sum_{i\in\mathcal{F}\setminus\{1\}} \sum_{v\in\mathcal{V}\setminus\{v'\}} \Big(\sum_{Z\in\Omega:Z=Y-\Delta(v,i)} \mu^v(Y^v - e_i, Y^v) \\ &\times \mathrm{P}(Z) - \mu^v(Y^v, Y^v+e_i)\mathrm{P}(Y)\Big)\Big\}\Big], \end{aligned} \tag{3}$$

for $X \in \Omega$, where $\Delta(v,i)$ is an $M \times n$ matrix of all zeroes except for a unity at position (v,i). We omit the time index whenever dependence is unambiguous.

The master equation (3) can not be solved analytically. We, therefore, carry out an aggregation of the chain into population counts. Define $\deg(v) := \sum_{l\in\mathcal{V}} \mathbb{1}((v,l) \in \mathcal{E})\ \forall v \in \mathcal{V}$ and $\mathcal{D} := \{d \mid \exists v \in \mathcal{V}, \deg(v) = d\}$. Consider a map T defined by $\mathrm{T}(X) := (z_x^k : x \in \mathcal{T}, k \in \mathcal{D})$ where $z_x^k := \sum_{v\in\mathcal{V}} \mathbb{1}(X^v = x)\mathbb{1}(\deg(v) = k)$, the number of degree-$k$ peers at buffer configuration x. Define an equivalence relation $\overset{\mathrm{T}}{\sim}$ on Ω as $X \overset{\mathrm{T}}{\sim} Y \iff \mathrm{T}(X) = \mathrm{T}(Y)$ and $\Omega_t := \{X \in \Omega : \mathrm{T}(X) = t\}$ for each t. Then, $\{\Omega_t\}$ is a partition of Ω and each Ω_t is an equivalence class. The induced probability is given by

$$\mathrm{P}(\mathrm{T}(X) = t) = \sum_{X\in\Omega:\mathrm{T}(X)=t} \mathrm{P}(X). \tag{4}$$

Such an aggregation is useful in reducing the state space if we now consider the lumped process T of population counts instead. In [10], we provide a necessary and sufficient condition for such an aggregation to engender state space reduction and also discuss worst case scenarios. We emphasize that we do lose information in the process of aggregation. Also, the lumped process is not necessarily Markovian [9].

III. Mean-field theoretic analysis

In this section, we approximate the lumped process T defined in Section II-B, when M is large. Mean-field theory is extensively used for this purpose [4], [13], [18]. As a first step in this direction, peers are assumed to be independently interacting with a mean environment. This allows us to treat each neighbour of a degree-k peer as an independent sample from a mean environment. We also impose that peers having the same degree play the same chunk selection strategy and thus, behave indistinguishably in a large random graph, suggesting that such a mean-field behaviour can very well be described by population counts. We, therefore, define a mean-field population model that lumps the original process according to the equivalence relation $\overset{\mathrm{T}}{\sim}$. We shall index all the relevant quantities by degree k in the following, instead of indexing by peers.

[1] That is, we place a Poisson clock on each edge of the graph.

A. Mean-field master equations

Consider the process $\{Z_t\}_{t\geq 0}$ defined as $Z_t := (z_x^k(t) : x \in \mathcal{T}, k \in \mathbb{N})$ where $z_x^k(t)$ is the number of degree-k peers at buffer configuration $x \in \mathcal{T}$ at time t. We get our mean-field transition rates for a degree-k peer as follows, for each $k \in \mathbb{N}, u \in \mathcal{T}$ and $i \in \mathcal{F} \setminus \{1\}$ such that $u_i = 0$,

$$\begin{aligned}\beta^k(u, u+e_i) &= \sum_{l=1}^{k} \varsigma \mathrm{E}[\mathbb{1}(Y_l(i) = 1)\alpha^k(i, u, Y_l)] \\ &= k\varsigma \mathrm{E}[\mathbb{1}(Y_1(i) = 1)\alpha^k(i, u, Y_1)],\end{aligned}$$

where $\{(Y_l, d_l) \mid Y_l = (Y_l(1), Y_l(2), \ldots, Y_l(n)) \in \mathcal{T}, d_l \in \mathbb{N}\}_{l=1}^k$ is a set of k independent and identically distributed (i.i.d.) samples from the mean environment of a degree-k peer. The first component of each neighbour is the buffer state and the second component, its degree. Note that d_l's are distributed according to q of eq. (1). Then,

$$\begin{aligned}&\mathrm{E}[\mathbb{1}(Y_1(i) = 1)\alpha^k(i, u, Y_1)] \\ &= \sum_{v\in\mathcal{T}:v_i=1} \sum_{m\in\mathbb{N}} \alpha^k(i, u, v)\mathrm{P}(Y_1 = v \mid d_1 = m)\mathrm{P}(d_1 = m) \\ &= \sum_{v\in\mathcal{T}:v_i=1} \sum_{m\in\mathbb{N}} q(m)\frac{\mathrm{E}[z_v^m]}{n_m}\alpha^k(i, u, v).\end{aligned}$$

where n_m is the number of peers of degree m. Thus, we get,

$$\beta^k(u, u+e_i) = k\varsigma \sum_{v\in\mathcal{T}:v_i=1} \sum_{m\in\mathbb{N}} q(m)\frac{\mathrm{E}[z_v^m]}{n_m}\alpha^k(i, u, v), \quad (5)$$

for each $k \in \mathbb{N}$, $u \in \mathcal{T}$ and $i \in \mathcal{F} \setminus \{1\}$ such that $u_i = 0$. For $i = 1$, we set β such that $\sum_{u\in\mathcal{T}:u_1=1} \frac{z_{u-e_1}^k}{n_k}\beta^k(u - e_1, u) = \frac{1}{M}$, the total input to the system by the server. Define the change vector $\varrho : \mathbb{N} \times \mathcal{T} \times \mathcal{F} \to \{-1, 0, 1\}^{|\mathcal{T}|\times\mathbb{N}}$ such that $Y = Z - \varrho(k, u, i) \implies y_u^k = z_u^k + 1, y_{u+e_i}^k = z_{u+e_i}^k - 1, y_x^l = z_x^l \forall l \in \mathbb{N} \setminus \{k\}, x \in \mathcal{T} \setminus \{u\}$. Broadening the scope of definition of β by setting it to 0 for all $u, u+e_i$ not covered in eq. (5), for large M, we have the following *mean-field master equation*

$$\begin{aligned}\frac{d\mathrm{P}(Z)}{dt} = &-\mathrm{P}(Z) + \sum_{\substack{Y:\sum_{\mathbb{S}v=u} y_v^l = z_u^l \\ \forall u,v\in\mathcal{T}, l\in\mathbb{N}}} \Bigg[\mathrm{P}(Y) \\ &+ \sum_{l\in\mathbb{N},u\in\mathcal{T},i\in\mathcal{F}} (y_u^l + 1)\beta^l(u, u+e_i) \\ &\qquad \times \mathrm{P}(Y - \varrho(l, u, i)) \\ &- \sum_{l\in\mathbb{N},u\in\mathcal{T},i\in\mathcal{F}} y_u^l \beta^l(u, u+e_i)\mathrm{P}(Y)\Bigg].\end{aligned} \quad (6)$$

In pursuance of the mean dynamics, we begin by first setting $\mathrm{P}(Y) = 0 \,\forall\, Y \notin \mathbb{N}_0^{|\mathcal{T}|\times\mathbb{N}}$ where $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$, and then by defining, for each $l \in \mathbb{N}, u \in \mathcal{T}, i \in \mathcal{F}$, the following quantity $\gamma_{l,u,i}(Z) := z_u^l \beta^l(u, u+e_i)$. Next, we note that, in mean field, we can write $\mathrm{E}[\gamma_{l,u,i}(Z)]$ as $\mathrm{E}[z_u^l]\beta^l(u, u+e_i)$. The following result encapsulates the mean dynamics of the system.

Result 1. *The process $\{Z_t\}_{t\geq 0}$ admitting master equation (6) satisfies*

$$\frac{d\mathrm{E}[Z]}{dt} = -\mathrm{E}[Z] + \mathrm{E}[Y] + \sum_{l\in\mathbb{N},u\in\mathcal{T},i\in\mathcal{F}} \varrho(l, u, i)\mathrm{E}[\gamma_{l,u,i}(Y)], \quad (7)$$

where $Y \in \mathbb{N}_0^{|\mathcal{T}|\times\mathbb{N}}$ is such that $y_u^l = \sum_{\mathbb{S}v=u} z_v^l \forall l \in \mathbb{N}, u \in \mathcal{T}$.

The proof is provided in [10]. Looking closely at eq. (7) and recalling the definition of $\varrho(l, u, i)$, we write down explicitly, for each $u \in \mathcal{T}, k \in \mathbb{N}$

$$\begin{aligned}\frac{d\mathrm{E}[z_u^k]}{dt} = &-\mathrm{E}[z_u^k] + \sum_{v\in\mathcal{T}:\mathbb{S}v=u} \Bigg[\mathrm{E}[z_v^k] \\ &+ \sum_{i\in\mathcal{F}} \mathrm{E}[z_{v-e_i}^k]\beta^l(v - e_i, v) - \sum_{i\in\mathcal{F}} \mathrm{E}[z_v^k]\beta^l(v, v + e_i)\Bigg],\end{aligned} \quad (8)$$

a self-consistent (autonomous) set of ordinary differential equations (ODEs) for the mean population counts.

It is convenient to work with proportions to study the mean dynamics. Therefore, define $W_t := (w_x^k(t) : x \in \mathcal{T}, k \in \mathbb{N})$ where $w_x^k(t) := z_x^k/n_k$. We argue that, when the number of peers is large, it suffices to study the mean dynamics of the proportions, for the fluctuation around mean is expected to be negligible for large systems [12]. Therefore, denoting $\mathrm{E}[w_x^k]$, with abuse of notation, by w_x^k itself, we write down the following *rate equations*,

$$\begin{aligned}\frac{dw_u^k}{dt} = &- w_u^k + \sum_{v\in\mathcal{T}:\mathbb{S}v=u} \Bigg[w_v^k \\ &+ \sum_{i\in\mathcal{F}} \Big(w_{v-e_i}^k \beta^k(v - e_i, v) - w_v^k \beta^k(v, v + e_i)\Big)\Bigg],\end{aligned} \quad (9)$$

for each $u \in \mathcal{T}, k \in \mathbb{N}$. We find steady-state proportions by setting $\frac{dw_u^{(k)}}{dt} = 0$, giving rise to following fixed point equations at steady state,

$$\begin{aligned}w_u^k = \sum_{v\in\mathcal{T}:\mathbb{S}v=u} \Bigg[w_v^k + \sum_{i\in\mathcal{F}} \Big(&w_{v-e_i}^k \beta^k(v - e_i, v) \\ &- w_v^k \beta^k(v, v + e_i)\Big)\Bigg].\end{aligned} \quad (10)$$

Observe that $\sum_{u\in\mathcal{T}} \frac{dw_u^k}{dt} = 0$ for all $k \in \mathbb{N}$. This is because of the fact that proportions sum up to 1, *i.e.*, $\sum_{u\in\mathcal{T}} w_u^k = 1 \forall k \in \mathbb{N}$. It does merit some attention that the population model presented here can be thought of as an infection model with 2^n distinct levels of a disease, each level being represented by a $u \in \mathcal{T}$ and (gradual) recovery being represented by the shifting of buffer state after playback. This amounts to saying, a peer with all buffer positions filled is infected to the highest extent of a disease and if it does not download any chunk, *i.e.*, if it does not get infected, within the next n-time units, it will gradually recover to a state of complete susceptibility (no chunk available).

One of the key metrics of performance in live streaming context is the buffer probability. The buffer probability of

index i of a degree-k peer is the probability that a degree-k peer has a chunk at buffer index i. In mean field, this becomes the proportion of degree-k peers that have chunks at buffer index i. Therefore, we define $p_k : \{1, 2, \ldots, n\} \to [0, 1]$, the buffer probability of a peer of degree $k \in \mathbb{N}$ as

$$p_k(i) = \sum_{u \in \mathcal{T} : u_i = 1} w_u^k. \tag{11}$$

The corresponding global performance of the network is linked to these degree-specific buffer probabilities through the associated degree distribution of $\mathcal{G}$ as follows

$$p(i) = \sum_{k \in \mathbb{N}} \pi(k) p_k(i). \tag{12}$$

Next, we try to derive a recurrence relation among p_k's by means of eq. (10) to understand their behaviour. We have the following result in that direction.

Result 2. *The process $\{W_t\}_{t \geq 0}$ of proportions obeying rate equation eq.* (9)*, admits the following recursion relation among the buffer probabilities at steady state*

$$\begin{aligned} p_k(i+1) &= p_k(i) + \sum_{u \in \mathcal{T} : u_i = 1} w_{u-e_i}^k \beta^k(u - e_i, u) \\ p(i+1) &= p(i) + \sum_{k \in \mathbb{N}} \pi(k) \sum_{u \in \mathcal{T} : u_i = 1} w_{u-e_i}^k \beta^k(u - e_i, u) \end{aligned}$$

for all $i, k \in \mathbb{N}$. Moreover, buffer probabilities are nondecreasing functions of their arguments, i.e., buffer indices.

The proof is omitted for want of space and is given in [10].

Interpretation of result 2: The left hand side of the recurrence relation gives the probability that the chunk required to fill the buffer location $i + 1$ is present. The right hand side tells us that there are two possible ways to have the chunk at buffer index $i + 1$ present. First, it could already be there at buffer index i, with probability of buffer index i, and was made available at index $i + 1$ due to shifting. Second, the chunk was not there, but the peer could download it in the mean time. Roughly speaking, this occurs with probability $\sum_{u \in \mathcal{T} : u_i = 1} w_{u-e_i}^k \beta^k(u - e_i, u)$ for a degree-k peer. This forms the basis of our further analysis of buffer probabilities.

Now we make use of a largely adopted assumption about the chunk selection function. We assume that the chunk selection function of a degree-k peer, $\alpha^k(i, u, v)$ does not depend on any particular value of u and v, but rather assigns probability to buffer indices according to their relative importance as pronounced by EDF and LDF. Call this simplified policy s_k, instead of α^k. This implies,

$$\begin{aligned} \beta^k(u, u + e_i) &= k\varsigma \sum_{v \in \mathcal{T} : v_i = 1} \sum_{l \in \mathbb{N}} q(l) w_v^l \alpha^k(i, u, v) \\ &= k\varsigma s_k(i) \sum_{l \in \mathbb{N}} q(l) p_l(i) = k\varsigma s_k(i) \theta_i, \end{aligned}$$

where $i \in \mathcal{F}$ and $\theta_i := \sum_{l \in \mathbb{N}} q(l) p_l(i)$ encapsulates the probability that an arbitrarily given edge points to a node where chunk i is available.

Let us now revisit the recurrence relation in result 2 and plug in the above simplified quantities. In order to do so, note that, for all $i \in \mathcal{F}$,

$$\begin{aligned} \sum_{u \in \mathcal{T} : u_i = 1} w_{u-e_i}^k \beta^k(u - e_i, u) &= \sum_{v \in \mathcal{T} : v_i = 0} w_v^k \beta^k(v, v + e_i) \\ &= k\varsigma \theta_i s_k(i) \sum_{v \in \mathcal{T} : v_i = 0} w_v^k \\ &= k\varsigma \theta_i (1 - p_k(i)) s_k(i). \end{aligned}$$

The recursion relation in result 2 then reads

$$p_k(i+1) = p_k(i) + k\varsigma \theta_i (1 - p_k(i)) s_k(i), \tag{13}$$

where $k \in \mathbb{N}$, $i = 1, 2, \ldots, n - 1$, and $\varphi := p_k(1) = \frac{1}{M}$. Such a recurrence relation in the special case of a homogeneous system has served as a starting point for the study of buffer probabilities in a number of articles in the literature, *e.g.*, [27], [33], [34]. In fact, by choosing $\pi(k) = \mathbb{1}(k = k^*), \varsigma = \frac{1}{k^*}$ for some $k^* \in \mathbb{N}$, we retrieve from eq. (13) the corresponding recurrence relation in the homogeneous setup, as found in [27], [33], [34]. Our endeavour was to provide a principled approach to derive such a recurrence relation in a more general heterogeneous setup exhibiting degree dependence of peers.

Remark. Equations (12) and (13) are two key instruments in our analysis of buffer probabilities. While eq. (13) describes the playback experience of a degree-k peer, a local aspect, eq. (12) allows us to combine these local information through degree distributions of arbitrary networks to give us a global view. This is notable because even this simple, approximate model allows us to capture the dependence of performance on network structure by plugging in its degree distribution.

We shall now focus on the two popular chunk selection strategies, namely, LDF and EDF. We follow the same interpretations of EDF and LDF as laid down in [34].

B. Chunk selection function

1) Latest deadline first (LDF) strategy: This strategy aims to download the rarest piece first. The priority is thus on the initial buffer indices. Therefore, $s_k(i)$ can be written as

$$s_k(i) = [1 - \varphi] \prod_{j=1}^{i-1} \left[p_k(j) + (1 - p_k(j))(1 - k\varsigma\theta_j) \right].$$

The explanation, omitted for want of space, is simple and is provided in [10]. This gives us the following result.

Result 3. 1) *The chunk selection function for the latest deadline first (LDF) strategy can be expressed as*

$$s_k(i) = 1 - p_k(i). \tag{14}$$

2) *The recursion relation for buffer probabilities for the latest deadline first (LDF) strategy has the following form, for $i = 1, 2, \ldots, n - 1$ and $k \in \mathbb{N}$*

$$p_k(i+1) = p_k(i) + k\varsigma\theta_i (1 - p_k(i))^2. \tag{15}$$

The proof is similar to [34], however, for the sake of completeness, it is provided in [10].

2) Greedy strategy: The greedy strategy or the earliest deadline first (EDF) strategy seeks to download pieces that are close to playback. The priority is thus on playback urgency and hence on the final buffer indices. Therefore, the chunk selection function can be expressed as

$$s_k(i) = [1-\varphi] \prod_{j=i+1}^{n-1} \left[p_k(j) + (1-p_k(j))(1-k\varsigma\theta_j)\right].$$

The explanation is similar to the case of the LDF strategy, with the notable exception that now we require to search buffer index n first, then $n-1$ and so on.

Result 4. 1) *The chunk selection function for the greedy strategy (EDF) can be expressed as*

$$s_k(i) = 1 - \varphi - p_k(n) + p_k(i+1). \tag{16}$$

2) *The recursion relation for buffer probabilities for the greedy strategy (EDF) has the following form, for $i = 1, 2, \ldots, n-1$ and $k \in \mathbb{N}$*

$$\begin{aligned} p_k(i+1) = p_k(i) + k\varsigma\theta_i(1-p_k(i)) \\ [1-\varphi-p_k(n)+p_k(i+1)]. \end{aligned} \tag{17}$$

The proof is provided in [10].

Remark. A typical EDF buffer probability curve exhibits a late, sharp increase, contrary to an LDF curve (see [33], [34]). However, when M is large, EDF hinders propagation of new chunks. While LDF is known to possess good scalability, EDF outperforms LDF when M is small. We wish to exploit this feature of EDF even when M is large. In order to do so, we must devise a way to arrest this content bottleneck. We conjecture that this can be done by employing a reasonably small percentage of strong peers (the ones with higher bandwidth, say, but not necessarily connected directly to the server) to play LDF so as to act as *pseudo-servers* in the system. We pursue this idea by studying different strategy profiles in a minimal setup with only two degrees, where we call the peers of higher degree strong peers and peers of smaller degree, weak peers.

Suppose there are only two degrees $k_1, k_2 \in \mathbb{N}$ in the system where $k_1 < k_2$. For typographical convenience, we shall subscript all the relevant variables with only $1, 2$ instead of k_1, k_2 respectively, whenever the degree of a vertex appears as a subscript or as an argument to a function, *e.g.*, π_1, π_2 in place of $\pi(k_1), \pi(k_2)$ respectively and $p_1(i), p_2(i)$ in place of $p_{k_1}(i), p_{k_2}(i)$ respectively.

C. Pure LDF strategy

As seen in Section III-B1, buffer probabilities for the two degrees k_1, k_2 when everybody plays LDF, are given by the following recursion relations

$$\begin{aligned} p_1(i+1) = \quad p_1(i) + k_1\varsigma\theta_i(1-p_1(i))^2, \\ p_2(i+1) = \quad p_2(i) + k_2\varsigma\theta_i(1-p_2(i))^2, \end{aligned} \tag{18}$$

for $i = 1, 2, \ldots, n-1$. We adopt a continuous approximation of the above two difference equations (as done in [27], [34], for instance). Treating the buffer index i as a continuous variable x and writing y_1, y_2, θ for $p_1(i), p_2(i)$ and θ_i respectively, we have the following differential equations

$$\begin{aligned} \frac{dy_1}{dx} &= k_1\varsigma\theta(1-y_1)^2, \\ \frac{dy_2}{dx} &= k_2\varsigma\theta(1-y_2)^2. \end{aligned} \tag{19}$$

The above luckily allows an exact solution which we present in the next result.

Result 5. *For the pure LDF strategy and large systems, i.e., when $M \to \infty$, the two buffer probabilities are related according to the following equation*

$$y_2 = \frac{y_1}{r + (1-r)y_1}, \tag{20}$$

where $r = \frac{k_1}{k_2}$ is the relative strength of the weak peers compared to the strong ones.

The proof is given in [10]. We immediately see that $y_2 > y_1$, *i.e.*, the stronger peers have better performance owing to their greater rate of interaction. However, this difference in performance for the weak peers due to degree disparity can be made arbitrarily small if a sufficiently large buffer is made available. Another interesting consequence is that the above can now be used to derive an expression for buffer-size requirements and facilitate sensitivity analysis therefrom. That is, given $\epsilon_1 = 1-p_1(n)$, the playback discontinuity of the weak peers, we can find the required buffer length of the weak peers $n_1 = f(\pi, r, \epsilon_1)$ that ensures performance at level ϵ_1 for some f[2]. Notice that the global performance is related to ϵ_1 by

$$1 - \epsilon = \pi_1(1-\epsilon_1) + \pi_2 \frac{1-\epsilon_1}{1-(1-r)\epsilon_1},$$

where $1 - \epsilon = p(n)$. This can be used when we intend to achieve a prespecified level of global performance.

D. Mixed strategy: SchedMix

Now we turn to the mixed strategy referred to as SchedMix. Suppose the weaker peers of degree k_1 adopt EDF and the stronger peers of degree k_2, LDF. Following Sections III-B1 and III-B2, we have the following recursion relations

$$\begin{aligned} p_1(i+1) = p_1(i) + k_1\varsigma\theta_i(1-p_1(i)) \\ [1-\varphi-p_1(n)+p_1(i+1)], \\ p_2(i+1) = p_2(i) + k_2\varsigma\theta_i(1-p_2(i))^2, \end{aligned} \tag{21}$$

for $i = 1, 2, \ldots, n-1$. As before, we shall use a continuous approximation to study their behaviour. Writing $\epsilon_1 = 1-p_1(n)$, we get the following differential equations:

$$\begin{aligned} \frac{dy_1}{dx} &= \frac{k_1\varsigma\theta(1-y_1)(y_1-\varphi+\epsilon_1)}{1-k_1\varsigma\theta(1-y_1)}, \\ \frac{dy_2}{dx} &= k_2\varsigma\theta(1-y_2)^2. \end{aligned} \tag{22}$$

[2]The exact expression is provided in [10].

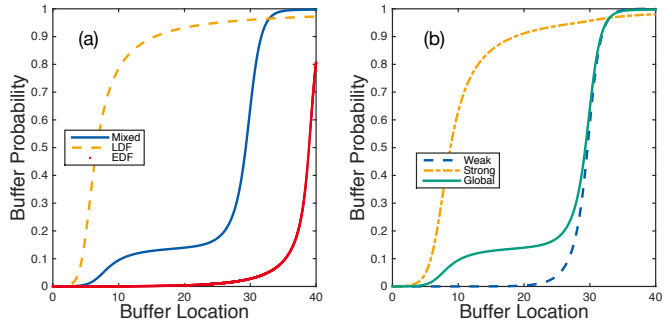


Fig. 1: Performance comparison based on mean-field analysis of buffer probabilities. (a) Global buffer probabilities for the three strategy profiles. SCHEDMIX gives higher playback continuity than both EDF and LDF for the given buffer length. (b) Comparison of weak versus strong under SCHEDMIX. Weak peers indeed eventually outperform the strong peers under SCHEDMIX. Parameter values: $M = 10000, k_1 = 5, k_2 = 15, \pi_1 = 0.85 = 1 - \pi_2, \varsigma = 0.20$.

The above equations, unfortunately, do not yield an analytic solution. Therefore, we resort to numerical solution to compare global performance of the system under different strategy profiles. It turns out that performance under SCHEDMIX is indeed better than that under the pure LDF strategy (see Fig. 1), substantiating our claim.

When we compared performance of weak peers versus strong ones, an interesting phenomenon was observed. The weak peers could eventually manage to outperform the strong ones, caused by a sharp increase in buffer probabilities that a typical "EDF curve" enjoys and what we call the boon of heterogeneity (see Fig. 1). This phenomenon is in agreement with our supposition and can be explained intuitively. Both strong and weak peers benefit from being exposed to a heterogeneous environment. In a homogeneous setup, one would expect somewhat similar availability of chunks among all its neighbours. On the contrary, a heterogeneous environment makes available a diverse collection of chunks. This prepones the steep rise that a typical "EDF curve" enjoys. Since an EDF curve has a greater growth-rate in the neighbourhood of 1 (see [33], [34]), weak peers can eventually outperform LDF-playing strong peers even for moderate buffer-lengths.

Remark. We do not consider the pure EDF strategy separately here as it can be studied in a similar fashion. In [10], we also provide a short stability analysis that gives an additional justification of why the weak peers outperform the strong ones.

IV. SIMULATION OF THE STOCHASTIC MODEL

In this section, we document our findings from the simulation of the stochastic model. This is carried out in two steps: first, generation of a random graph and second, simulation of the content delivery process in accordance with Section II.

We dispense with a description of how to simulate CTMCs due to insufficiency of space. Interested readers are referred to [10] where we also investigated the effect of assuming an exponential shifting time versus a deterministic one and confirmed that the behaviour of the strategies remained unaffected.

Startup latency: The second metric that we look at is the start-up latency. It is the time a peer should wait before starting playback. While there is no unanimity as to how one should define this quantity, it is reasonable to wait until a newly arrived peer's buffer attains a steady state. If it starts playback before that, it is likely to experience below steady state playback quality initially. On the other hand, waiting longer will not improve long-term playback experience. In a homogeneous set-up where everybody plays the same policy and has the same buffer probabilities, as argued in [33], this is well represented by $\sum_i p(i)$, the average number of available chunks at each peer. In our heterogeneous model, a higher degree peer interacts more often than a lower degree peer. Therefore, a newly arrived degree-k peer should have start-up latency of $k\varsigma \sum_i p(i)$ in the mean-field. The corresponding global metric follows as $\mathrm{E}[k]\varsigma \sum_i p(i)$. For aesthetic reasons, we normalise this quantity to $(0, 1)$.

Impact of network structure: In order to see the impact of network structure, we perform simulation of the model on Barabási-Albert (BA) preferential attachment [2] and Watts-Strogatz (WS) small world [25] networks. Simulation results on a BA network with 2000 peers (with 25% of them playing LDF) and that on a WS network with 5000 peers (with 20% of them playing LDF) are depicted in Fig. 2. In both cases, the mixed strategy SCHEDMIX gives a better performance, corroborating our claim. More importantly, it causes a significant reduction in start-up latency.

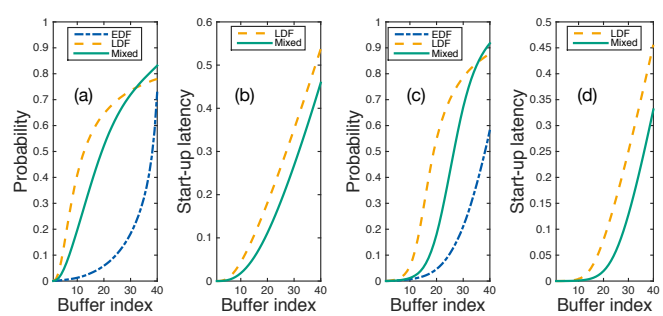


Fig. 2: Impact of network structure and performance evaluation in terms of buffer probabilities and the start-up latency on a Barabási-Albert (BA) and a Watts-Strogatz (WS) graph. Figures (a), (b) show performance on a BA graph with 2000 peers. Figures (c), (d) display performance on a WS graph with 5000 peers. In both cases, $n = 40, \varsigma = 0.25$. Please note that start-up latency is shown only for strategies ensuring playback continuity of at least 0.75 with buffer size $n = 40$.

Remark. Although Fig. 2 stands affirmatory to the fact that SCHEDMIX does outperform the pure LDF and the pure EDF strategies, the crux of employing SCHEDMIX remains in letting most peers play greedy. SCHEDMIX, thus, allow for

smaller start-up latency to ensure good playback performance for everyone (at least as good as pure LDF strategy). This is a significant benefit.

V. Full-stack simulation study

A. Practical system model

We also designed and implemented a practical P2P live video streaming system and considered communication network-related factors. The implementation is based on the Simonstrator API [21] and is evaluated using the network simulation framework PeerfactSim.KOM [23]. The full-stack implementation includes protocols for mesh establishment and maintenance, the scheduling mechanisms themselves, and buffer management. Due to space constraints, we omit some details that follow state-of-the-art P2P streaming systems and in particular [22], [26].

1) Mesh establishment: For the establishment of the mesh overlay structure, a join procedure is implemented that uses a BitTorrent-like tracker as central node registry. The tracker selects a maximum of 30 neighbours uniformly at random from the set of currently active peers and sends the list to the requesting peer. For both in- and outgoing directions, peers calculate a maximum number of connections by dividing 90% of the available bandwidth by the video bitrate and rounding the result to the next integer. Joining peers strive to fill their free incoming connections and thus contact multiple peers from the initial neighbour list in parallel and query the tracker for additional contacts if necessary. Receivers accept the requests depending on the availability of free connection slots. Limiting the number of connections is combined with a per-connection transfer queue to avoid too many parallel transfers that could stall each other, leading to a situation where video chunks would take an indefinite time to be delivered. In case a peer has no free connection slots, requests can still be accepted with a small probability to foster randomness in the mesh structure where early peers might otherwise be already blocked, and to allow peers with high bandwidths to eventually become well connected. The second aspect turned out to be important for applying the proposed mixed scheduling strategy, which relies on degree heterogeneity across peers.

2) Scheduling and data exchange: The actual scheduling of data transmissions is done by each peer individually based on its local clock with a rate proportional to its in-degree and buffer status. Chunks are selected by the scheduling strategy from a defined request window on the local buffer, which is used to limit the chunks requested. For the simulations, the window size is set to a default value of 20, starting at the beginning of the buffer for LDF and the end for EDF. The selected chunks are assigned uniformly at random to a peer's in-connections, are batched on a per-neighbour basis into chunk requests, and are sent out. The buffer length is 4 seconds, translating to 50 chunks at a rate of 8 chunks/s.

3) Playback policy: A simplified policy was realized for this initial simulation study. Joining peers learn about the current broadcasting position from the tracker and start their playback after 4 seconds (the buffer length). In the meantime, they establish connections and start requesting chunks. Once started, the playback proceeds based on the local clock and at the video bitrate. Chunks that miss the playback deadline are recorded in terms of *playback continuity*.

B. Full-stack simulation results

A simulative sensitivity analysis was conducted, covering key system and environment parameters. Due to space constraints, only results for the default configuration are presented here (see [10] for additional results). The defaults were obtained by conducting calibration runs for several parameter combinations. Due to the large configuration space, ensuring an overall optimal configuration was not possible and is hard to achieve in general. The presented results, thus, focus on studying the potential of the proposed scheduling mechanism, not on the absolute performance. All simulations were repeated 30 times with different random seeds. 95%-confidence intervals are reported for all mean values.

Simulation workload: Peers are divided into three resource classes based on bandwidth distributions reported in [17] (see Table I). We acknowledge that these bandwidths are rather high in comparison to configurations used in related works. Yet, we intend to reflect a setting in that the delivery is not primarily limited by peer bandwidths but rather focus on content bottlenecks resulting from the scheduling strategy itself [7]. This is important as peers can only use available bandwidth if scheduling ensures a timely replication of chunks. At the beginning of the simulation scenario, peers subsequently join the system in a random order and at a constant arrival rate. After the system stabilizes, performance and cost metrics are recorded on per-peer basis and aggregated for 60 seconds intervals and over the total simulation time of 90 minutes. Peers stay in the system until the end of the simulation.

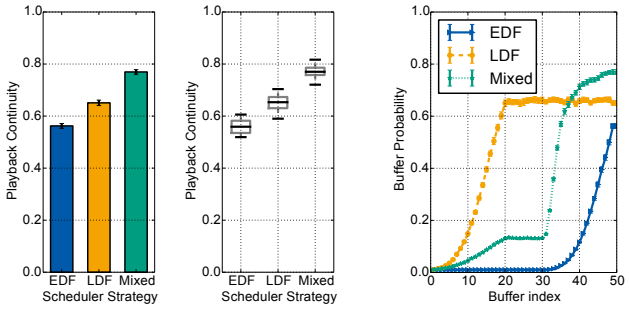
TABLE I: Used peer bandwidth distribution based on [17].

Class	Number	Share	UL BW (Mbps)	DL BW (Mbps)
Low	50	50%	5	26
Medium	30	30%	4.5	60
High	20	20%	56	134

As observed in [14], the source bandwidth plays an important role and, thus, its influence is studied in more detail in [10]. Per default, a single source is used with an up-/download bandwidth of 12.5Mbps. With a video bitrate of $1,500$ Kbps as commonly observed and recently reported [11], this translates to a maximum out-degree of 7.

Figure 3 shows the streaming performance for the default configuration. Here, it is to note that using a request window of size 20 is an extreme case as it artificially limits the request rate by localizing the chunks to be selected. This is done to highlight the key difference between the chunk selection strategies. Other configurations of this parameter are presented in [10]. Figure 3a shows that SchedMix achieves a significantly higher playback continuity compared to the pure strategies. The buffer probability (see Figure 3b) shows that

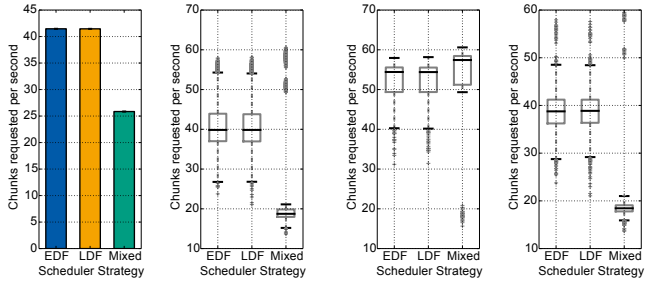
an early replication of new chunks greatly supports the greedy replication by EDF peers once entering their request window.



(a) Playback continuity. (b) Buffer probability.

Fig. 3: Streaming performance and buffer characteristics (default configuration), comparing EDF, LDF, and SCHEDMIX.

Figure 4 shows the resulting request rates for the individual chunk selection strategies. For the overall population, the request rate drastically drops using the mixed strategy, indicating a major reduction in overhead by roughly 50% for most peers, lending credence to the boon of heterogeneity. When separating strong peers (i.e. peers that play LDF in case of SCHEDMIX) from the rest of the population, it becomes apparent that this reduction is limited to the non-strong sub-population. The strong peers, however, are penalised as their average request rate is slightly increased for SCHEDMIX. At the same time (figures not shown here), the average playback continuity rate across the sub-populations does not show any difference. This supports the argument that there is a high incentive for strong peers to play LDF instead of EDF to improve the overall system performance (see [10] for a game theoretic perspective). These results strengthen our previous analytic arguments and establish that peer heterogeneity can be leveraged to form a powerful mixed scheduling strategy.



(a) All peers. (b) Strong (left) and rest (right).

Fig. 4: Number of requests (default configuration) for (4a) all peers as well separated into strong peers (LDF candidates in SCHEDMIX) and the remaining sub-population (4b).

VI. RELATED WORK

P2P live streaming has been studied extensively in the recent past, albeit in a homogeneous setup. Buffer probability received considerable attention for due reasons. Zhou *et al.* [33], [34] propose a simple model for its analysis based on mean field heuristics. Adamu *et al.* [1] also attempt to analyse it in the context of a discrete Markov chain. Zhao *et al.* [31] develop a population model and make interesting observations about optimal strategies. Outside live streaming, Hajek *et al.* [8], [35] highlight interesting aspects of stability of a P2P system and lay down insightful results on a CTMC formulation. However, there has been little investigation into heterogeneous strategies. The influence of degree, to the best of our knowledge, has also not been studied so far. Our endeavour in this article has been to carefully capture these two important aspects in a principled way. Infection models (see [4], [15], [18]) have proven useful in many computer science problems such as the study of security investments in networks [13], algorithms for distributed systems [5] and particularly many gossip algorithms that later found application in the peer-to-peer area as well [16], [30].

Zhang *et al.* [28] show that pull-based streaming can achieve high bandwidth utilisation and estimate a lower bound for the delivery ratio, based on simulations and a steady-state analysis of simple sender-requester topologies. Liang *et al.* [14] discuss scheduling as a key mechanism for P2P streaming and name source scheduling and bandwidth, the buffer sizes, and degrees as additional factors. Besides, they argue that scheduling plays a role only at a low resource index, whereas we observe clear differences due to content bottlenecks. In [7], a scheduling strategy is proposed, implicitly leveraging heterogeneity by maximising bandwidth utilisation of peers. The authors show a nearly optimal utilisation only for a fully connected mesh, greatly limiting the applicability to realistic setup. In contrast, we focus on pure pull strategies, do not assume a fully connected mesh, and do not focus on maximising bandwidth only. In [19], a mesh/push-based streaming system is proposed using LRU as scheduling strategy. The authors consider overlay rewiring and source scheduling to improve performance and provide supportive experimental results.

VII. DISCUSSION

In this paper, we contributed to building a sound mathematical framework for swarming on random graphs. The dependence of performance on degree was made explicit. The idea of a degree-based (strength-based) combination of primitive scheduling strategies led to two interesting revelations, namely, the boon of heterogeneity and the weak peers outperforming the strong ones. Inspired by these observations, we proposed our mixed strategy SCHEDMIX.

We showed that SCHEDMIX could guarantee good playback continuity at a smaller start-up latency and smaller unsuccessful download rate. The question, however, remains why the strong peers should opt to play LDF. We answer this question with a game theoretic argument in [10] where we established that SCHEDMIX is a Nash equilibrium.

The basic idea behind SCHEDMIX is rather simple: exploit the capabilities of the strong peers to help the weak ones. SCHEDMIX achieves this through degree-based assignment of strategies, but the notion goes beyond degrees. The virtues of

SCHEDMIX can also be achieved, perhaps more pronouncedly, by taking into account other important networking factors such as betweenness centrality, well-connectedness to the server. Our initial simulation results with betweenness centrality-based strategy assignment (not shown here) are affirmative.

We observed that it required only a small percentage of strong peers to uplift the weak peers and improve overall playback experience. However, the optimal percentage of strong peers required to do so is an open research problem.

Our mathematical framework can also serve as a foundation in problems other than the one in pursuit, *e.g.*, network security problems such as circulation of updates to anti-virus in the event of cyber attacks or the circulation of virus/malware itself, supply chain problems for products with limited validity, express consignment delivery problems. Its shifting feature makes it particularly interesting as it allows for multiple interpretations, *e.g.*, advertisement of promotional offers with deadlines, gradual recovery or mutation in the context of infection spread. Keeping analytic tractability aside, the prospect of incorporating more sophisticated mechanisms in practical implementation is broad. We expect to see application of SCHEDMIX in combination with more sophisticated mechanisms. One straightforward but important step is the application of SCHEDMIX in a state-of-the-art hybrid streaming system, where both mesh/pull and multi-tree/push-based mechanisms coexist. In this context it would also be interesting to understand the impact of other mechanisms, such as exchange of buffermaps or a streaming of layered media content. The results presented in this paper are encouraging in that SCHEDMIX could be used as an alternative to complex scheduling strategies in the growing number of scenarios where peer heterogeneity is inevitably given, *e.g.*, when bandwidth-constrained mobile users meet well-connected and high-capacity home users. Besides, the results could be used in the planning of *transitions* [6] between strategies when environmental conditions change.

ACKNOWLEDGEMENT

This work has been funded by the German Research Foundation (DFG) as part of project C03 within the Collaborative Research Center (CRC) 1053 – MAKI. The authors would like to thank Ralf Steinmetz for his valuable input and feedback.

REFERENCES

[1] A. Adamu, Y. Gaidamaka, and A. Samuylov, "Discrete Markov Chain Model for Analyzing Probability Measures of P2P Streaming Network," in *NEW2AN*, ser. LNCS. Springer, 2011, vol. 6869.

[2] A.-L. Barabási and R. Albert, "Emergence of Scaling in Random Networks," *Science*, vol. 286, no. 5439, 1999.

[3] C. Diot, B. Levine, B. Lyles, H. Kassem, and D. Balensiefen, "Deployment Issues for the IP Multicast Service and Architecture," *IEEE Network*, vol. 14, no. 1, 2000.

[4] R. Durrett, *Random Graph Dynamics*. Cambridge University Press.

[5] P. Eugster, R. Guerraoui, A.-M. Kermarrec, and L. Massoulié, "Epidemic Information Dissemination in Distributed Systems," *IEEE Computer*, vol. 37, no. 5, 2004.

[6] A. Frömmgen, B. Richerzhagen, J. Rückert, D. Hausheer, R. Steinmetz, and A. Buchmann, "Towards the Description and Execution of Transitions in Networked Systems," in *AIMS*, 2015.

[7] Y. Guo, C. Liang, and Y. Liu, "AQCS: Adaptive Queue-based Chunk Scheduling for P2P Live Streaming," in *IFIP NETWORKING*, 2008.

[8] B. Hajek and J. Zhu, "The Missing Piece Syndrome in Peer-to-Peer Communication," in *IEEE ISIT*, 2010.

[9] J. G. Kemeny and J. L. Snell, *Finite Markov Chains*. van Nostrand, Princeton, NJ, 1960.

[10] W. R. KhudaBukhsh, J. Rückert, J. Wulfheide, D. Hausheer, and H. Koeppl, "A Comprehensive Analysis of Swarming-based Live Streaming to Leverage Client Heterogeneity," Technische Universitaet Darmstadt, Germany, Tech. Rep., 12 2015. [Online]. Available: http://www.bcs.tu-darmstadt.de/media/bcs/Technical_Report_WKB_1.pdf

[11] D. Krishnappa, M. Zink, and R. Sitaraman, "Optimizing the Video Transcoding Workflow in CDNs," in *ACM MM*, 2015.

[12] T. G. Kurtz, *Approximation of Population Processes*. SIAM, 1981.

[13] M. Lelarge and J. Bolot, "A local mean field analysis of security investments in networks," in *Proceedings of the 3rd International Workshop on Economics of Networked Systems*. ACM, 2008.

[14] C. Liang, Y. Guo, and Y. Liu, "Is Random Scheduling Sufficient in P2P Video Streaming?" in *IEEE ICDCS*, 2008.

[15] T. M. Liggett, *Stochastic Interacting Systems: Contact, Voter and Exclusion Processes*. Springer, 1999.

[16] J. Liu, S. Rao, B. Li, and H. Zhang, "Opportunities and challenges of peer-to-peer internet video broadcast," *Proceedings of the IEEE*, vol. 96, no. 1, Jan 2008.

[17] Organisation for Economic Co-operation and Development, "OECD Broadband Report," Tech. Rep., 2014.

[18] R. Pastor-Satorras and A. Vespignani, "Epidemic Dynamics in Finite Size Scale-free Networks," *Physical Review E*, vol. 65, no. 3, 2002.

[19] F. Picconi and L. Massoulié, "Is There a Future for Mesh-based Live Video Streaming?" in *IEEE P2P*, 2008.

[20] R. Rejaie and N. Magharei, "On Performance Evaluation of Swarm-based Live Peer-to-Peer Streaming Applications," *Springer Multimedia Systems*, vol. 20, no. 4, 2014.

[21] B. Richerzhagen, D. Stingl, J. Rückert, and R. Steinmetz, "Simonstrator: Simulation and Prototyping Platform for Distributed Mobile Applications," in *ICST/ACM SIMUtools*, 2015.

[22] J. Rückert, B. Richerzhagen, E. Lidanski, R. Steinmetz, and D. Hausheer, "TopT: Supporting Flash Crowd Events in Hybrid Overlay-based Live Streaming," in *IFIP NETWORKING*, 2015.

[23] D. Stingl, C. Gross, J. Rückert, L. Nobach, A. Kovacevic, and R. Steinmetz, "PeerfactSim.KOM: A Simulation Framework for Peer-to-Peer Systems," in *IEEE HPCS*, 2011.

[24] F. Wang, Y. Xiong, and J. Liu, "mTreebone: A Collaborative Tree-Mesh Overlay Network for Multicast Video Streaming," *IEEE TPDS*, vol. 21, no. 3, 2010.

[25] D. J. Watts and S. H. Strogatz, "Collective Dynamics of 'Small-world' Networks," *Nature*, vol. 393, 1998.

[26] M. Wichtlhuber, B. Richerzhagen, J. Rückert, and D. Hausheer, "TRANSIT: Supporting Transitions in Peer-to-Peer Live Video Streaming," in *IFIP NETWORKING*, 2014.

[27] L. Ying, R. Srikant, and S. Shakkottai, "The Asymptotic Behavior of Minimum Buffer Size Requirements in Large P2P Streaming Networks," in *IEEE Information Theory and Applications Workshop (ITA)*, 2010.

[28] M. Zhang, Q. Zhang, L. Sun, and S. Yang, "Understanding the Power of Pull-based Streaming Protocol: Can We Do Better?" *IEEE JSAC*, vol. 25, no. 9, 2007.

[29] X. Zhang and H. Hassanein, "A Survey of Peer-to-Peer Live Video Streaming Schemes - An Algorithmic Perspective," *Computer Networks*, vol. 56, no. 15, 2012.

[30] X. Zhang, J. Liu, B. Li, and T. Yum, "Coolstreaming/donet: a data-driven overlay network for peer-to-peer live media streaming," in *IEEE INFOCOM 2005*, vol. 3, March 2005.

[31] B. Zhao, J. Lui, and D. Chiu, "Exploring the Optimal Chunk Selection Policy for Data-driven P2P Streaming Systems," in *IEEE P2P*, 2009.

[32] M. Zhao, P. Aditya, A. Chen, Y. Lin, A. Haeberlen, P. Druschel, B. Maggs, B. Wishon, and M. Ponec, "Peer-Assisted Content Distribution in Akamai NetSession," in *ACM IMC*, 2013.

[33] Y. Zhou, D. M. Chiu *et al.*, "A Simple Model for Analyzing P2P Streaming Protocols," in *IEEE ICNP*, 2007.

[34] Y. Zhou, D.-M. Chiu, and J. Lui, "A Simple Model for Chunk-scheduling Strategies in P2P Streaming," *IEEE/ACM TON*, vol. 19, no. 1, 2011.

[35] J. Zhu and B. Hajek, "Stability of a Peer-to-Peer Communication System," *IEEE Transactions on Information Theory*, vol. 58, no. 7, 2012.