Peer-to-Peer Gradient Topologies in Networks With Churn

Håkan Terelius and Karl Henrik Johansson, *Fellow, IEEE*

***Abstract*—We investigate the network topology convergence in a peer-to-peer (P2P) network system, where the goal of the system is to maximize live-streaming performance. The P2P system constructs a gradient overlay topology, characterized by a directed graph, where each node prefers neighbors containing higher utility values such that paths of increasing utilities emerge in the network topology. The gradient overlay network is built using gossiping and a preference function that samples nodes from a uniform random peer sampling service. Conditions for convergence to a gradient topology is derived, including the expected convergence time, and a threshold on the churn rate is provided for a gradient topology to emerge. Finally, a live-streaming video distribution experiment illustrates the benefits of constructing and utilizing the gradient topology for information dissemination in P2P systems.**

***Index Terms*—Network, overlay topology, peer to peer (P2P), video distribution.**

I. Introduction

THE Internet has penetrated our daily lives as the single most important information exchange system, and is used for sending messages, reading news, and watching television. The annual global IP traffic was expected to reach 1 ZB (10^{21} B) in 2016, and is expected to double until 2019. A majority of the Internet traffic consists of video delivery, constituting 64% of all consumer Internet traffic in 2014 and expecting to grow to 80% by 2019 [1]. Put into perspective, by 2019, a million minutes of video content will cross the Internet every second. This huge demand for network bandwidth is creating a lot of pressure toward efficient content distribution strategies.

Peer-to-peer (P2P) networking is a computer network architecture, where the nodes or peers both supply and consume resources. Thus, compared to a classical client–server architecture, in a peer-to-peer (P2P) network, peers are both clients and servers at the same time. Surveys of P2P networks have been carried out for content distribution technologies [2], search methods [3], resource discovery [4], and video-streaming systems [5], [6].

An important utility service while designing P2P architectures is the *peer sampling services*, which provides uniformly random samples of peers from the network. Gossip-based peer sampling systems have been developed in [7] and [8], extended to handle NAT traversal [9], and corrected for bias in networks with churn [10]. Randomized gossiping algorithms have also been used as tools for building distributed systems, in particular, in the areas of overlay networks, sensor networks, and cloud computing storage services [11], [12]. Convergence properties of gossip-based aggregation algorithms have been studied for fixed topologies [13] and accelerated methods for regular graphs, where each node has the same number of neighbors [14].

Research in gossiping has also focused on using the preferential connectivity model [15] to construct overlay network topologies, where nodes initially connected in a random graph use a preferential connection function to break the symmetry of the random graph, and build a topology that contains useful global information. Barabási [16] first described how a preferential attachment function in a growing network can build a scale-free network topology from a random graph. Barabási's preferential attachment functions are based on the global state, but in overlay networks, nodes only have a relatively small partial view of the system. Thus, the preference functions can only be based on the local state and the state of the node's neighbors. Examples of overlay networks that construct their topologies using gossiping and preference functions include Spotify, which preferentially connects nodes with similar music playlists [17], Sepidar, which preferentially connects P2P live-streaming nodes with similar upload bandwidth capacity [18], and T-Man, a framework that provides a generic preference function for building such overlays [19].

A fundamental property of P2P networks is user churn, i.e., that peers can join and leave the network at any time. Stutzbach and Rejaie [20] worked on characterizing churn models, whereas resilience against churn was considered in [21], [22] and [23], and Wang *et al.* [24] chose to identify stable peers in P2P services.

In this paper, we investigate a P2P network for efficient live-streaming television, inspired by gradienTv [25] and Sepidar [18]. The goal of this application is to distribute a data stream from a small set of seed nodes to every other node in the network, and the problem is to design distributed algorithms for creating an efficient overlay network topology. In particular, this paper

Manuscript received November 7, 2016; revised June 19, 2017 and December 16, 2017; accepted December 30, 2017. Date of publication January 23, 2018; date of current version December 14, 2018. This work was supported in part by the European Union Seventh Framework Programme under the Companion project, in part by the Swedish Research Council, and in part by the Knut and Alice Wallenberg Foundation. Recommended by Associate Editor Anna Scaglione. *(Corresponding author: Håkan Terelius.)*

The authors are with the ACCESS Linnaeus Centre, School of Electrical Engineering, KTH Royal Institute of Technology, Stockholm SE-100 44, Sweden (e-mail: hakante@kth.se; kallej@kth.se).

Digital Object Identifier 10.1109/TCNS.2018.2795704

provides a novel analytical characterization of the topology convergence problem, in which the network graph converges to a complete *gradient overlay network*, where most previous works have focused on experimental evaluations [7], [26], [27]. The gradient topologies are fundamental in self-organizing systems, and generalize the rooted trees topologies. The contribution of this paper is the convergence analysis of the given algorithm, including convergence rate estimates, and the derivation of a threshold on the churn rate for a gradient topology to emerge. This is an extension of our previous work [28] to include the global convergence rate and churn models.

The outline of this paper is as follows: In Section II, we introduce the network model and topology convergence problem, and in Section III, we give necessary and sufficient conditions for convergence. In Section IV, we study the convergence rate for the system, and in Section V, we study convergence properties when the network is subject to churn. In Section VI, we simulate the construction of a gradient topology using the model in Section II, and in Section VII, we evaluate the live-streaming performance in a real P2P application using the gradient overlay topology. Finally, Section VIII concludes this paper.

II. Gradient Topology Problem

An *overlay network* is a virtual network built on top of another network. Here, it denotes the P2P network topology built for television streaming over the Internet. The *gradient topology* belongs to the class of gossip-generated overlay networks that are built from a random overlay network through symmetry breaking using a preference function. Thus, we are given a node set $\mathcal{V} = \{1, \ldots, N\}$, and need to select directed edges $\mathcal{E}$ to construct our network $\mathcal{G}(\mathcal{V}, \mathcal{E})$.

In the live-streaming application, the idea is to utilize the nodes in the P2P network to aid in the content distribution, but since the peers are heterogeneous, not all peers will be equally useful. Thus, we classify each node $i \in \mathcal{V}$ with its *utility value* $u_i \in \mathbb{R}$, which captures, for example, the node's upload capacity, latency, and reliability for the P2P network. The initial sources for the live-streaming video feed would have the highest utility value in the network.

A gradient topology is an overlay network satisfying that, for any two nodes v_1 and v_2 with utility values u_{v_1} and u_{v_2}, if $u_{v_1} \geq u_{v_2}$, then $\text{dist}\,(v_1, v_\star) \leq \text{dist}\,(v_2, v_\star)$, where $v_\star$ is a node with highest utility in the system and $\text{dist}\,(\cdot, \cdot)$ is the length of the shortest path between the nodes in the network [29]. In other words, nodes with a higher utility value should be closer to the seed nodes compared to nodes with a lower utility value, so that gradient paths of increasing utilities emerge in the system (see Fig. 1).

In constructing the gradient overlay, the nodes $i \in \mathcal{V}$ build two sets of neighbors: a *similar view* $\mathcal{N}_i^s$ and a *random view* $\mathcal{N}_i^r$. For the similar view, nodes prefer neighbors with close but slightly higher utility values, whereas the random view is used to sample new nodes with uniform probability for possible inclusion in the similar view. Thus, node is neighbors are $\mathcal{N}_i = \mathcal{N}_i^s \cup \mathcal{N}_i^r$.

Each node i defines a *preference function* $>_i$ over its neighbors, where node i is said to prefer node a over node b (denoted by $a >_i b$) if

$$\begin{aligned} u_a \geq u_i > u_b \quad &\text{or if} \\ |u_a - u_i| < |u_b - u_i| \quad &\text{when } u_a, u_b > u_i \text{ or } u_a, u_b < u_i. \end{aligned}$$

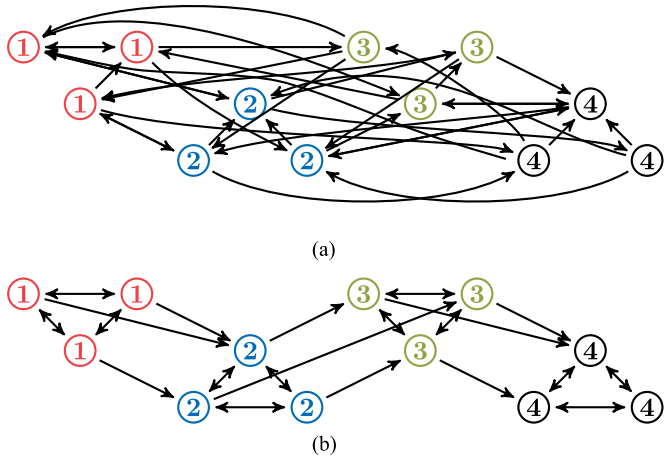

Fig. 1. Gradient network is described as a directed graph. The nodes are labeled with their respective utility value, and the edges from the similar neighbor sets are shown. In the gradient topology, paths of increasing utilities emerge. (a) Initial random overlay network. (b) Network after converging to a complete gradient topology

Algorithm 1: Topology Dynamics.

for each node $i \in \mathcal{V}$ **do**
  for every $t = 1, 2, 3, \ldots$ **do**
    Let $\mathcal{N}_i^r(t) = \{j\}$, where $j \in \mathcal{V}$ is a randomly selected node with uniform probability p_t, $0 < Np_t < 1$. Otherwise $\mathcal{N}_i^r(t) = \emptyset$.
    Choose $k \in \mathcal{N}_i^s(t-1)$ such that $\nexists v \in \mathcal{N}_i^s(t-1)$, $v \neq k$ and $k >_i v$.
    if $\mathcal{N}_i^r(t) \neq \emptyset$ and $j \notin \mathcal{N}_i^s(t-1)$ and $j >_i k$ **then**
      $\mathcal{N}_i^s(t) = \mathcal{N}_i^s(t-1) \cup \{j\} \setminus \{k\}$
    else
      $\mathcal{N}_i^s(t) = \mathcal{N}_i^s(t-1)$
    end if
  end for
end for

For any given initial overlay network, the topology is evolving through each node i at each time t updating its own neighbor set $\mathcal{N}_i(t)$ independently of the other nodes according to Algorithm 1. The algorithm can be summarized as repeatedly sampling random nodes from the network and evaluating their utility value. If the random node is preferred over the least preferred node in the similar set, then those two neighbors are exchanged.

It is assumed that the node out-degree $d_i(t) \doteq |\mathcal{N}_i^s(t)| = d_i$ stays constant throughout the algorithm. Note that the sampling probabilities p_t are time dependent and govern whether the random neighbor set $\mathcal{N}_i^r(t)$ is empty (with probability $1 - Np_t$). The reason for this is that a node can adapt its sampling

frequency to minimize the network overhead for building a gradient topology, and typically samples more frequently just after joining the network to improve its neighbor sets, and then lowering its sampling rate when the neighbors have stabilized. Notice also that Algorithm 1 can be run asynchronously on the nodes.

Remark: Note that no constraint is enforced on the in-degree of the nodes. However, in Section VII, the gradient topology is used for sampling nodes for a second auction algorithm, which limits the in-degree for the information dissemination network.

The preference function $>_i$ induced a partial order on the nodes $\mathcal{V}$. In order to study the network topology convergence to a gradient topology with the proposed algorithm, we let Λ_i to denote the set of optimal similar neighbor sets for node i, i.e., $\forall \widehat{N} \in \Lambda_i$, if there are no nodes $j \in \widehat{N}$ and $k \in \mathcal{V} \backslash \widehat{N}$ such that $k >_i j$. Notice that there could be multiple optimal neighbor sets.

For every node $i \in \mathcal{V}$, we define $X_i(t)$ as a counter for the number of nonoptimal neighbors in is similar neighbor set

$$X_i(t) \doteq d_i - \max_{\widehat{N} \in \Lambda_i} \left| \mathcal{N}_i^s(t) \cap \widehat{N} \right|.$$

Notice that $X_i(t)$ is monotonically decreasing under Algorithm 1 since an optimal neighbor will never be removed from the similar neighbor set $\mathcal{N}_i^s(t)$.

Let $\mathcal{G}(t)$ be the graphs generated by Algorithm 1. Then, we give the definition of gradient topology convergence as follows (see Fig. 1).

Definition 2.1: $\mathcal{G}(t)$ is said to converge to a complete gradient topology if

$$\lim_{t \to \infty} X_i(t) = 0$$

for all nodes $i \in \mathcal{V}$.

Remark: Connectivity of the final network is not always guaranteed. There exist some variations to the definition of the preference function and, for example, using a preference function, which prefers neighbors with strictly larger utility value would guarantee that for every finite network, the gradient topology will connect every node to the set of initial seeds.

Remark: In this paper, the utility value is assumed to be a global property. An interesting extension is to consider local utility values, which could, for example, capture differences in pairwise latency between the nodes.

III. Convergence Analysis

Since each node updates its neighbor set independently, the analysis can be carried out separately for each $X_i(t)$. We therefore simplify the notations in the following discussion, and let $X(t)$ represents $X_i(t)$ for an arbitrary node $i \in \mathcal{V}$.

Denote the maximum degree $D = \max_i\{d_i\}$, then $X(0) = D$ would be the worst possible initial condition. Furthermore, $X(t)$ decreases precisely when the randomly sampled node is a new optimal neighbor, and the probability of this event occurring is minimal when the optimal solution is unique, and then the probability is equal to

$$\mathbb{P}\left[X(t+1) = k-1 \mid X(t) = k\right] = kp_t, \quad k = 1, \ldots, D \tag{1}$$

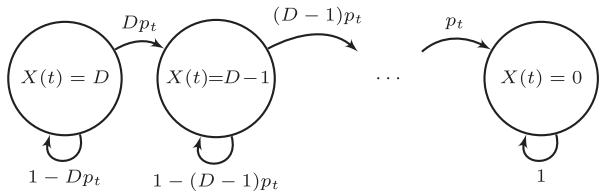

Fig. 2. Markov chain for the state evolution of a single node.

where k is the number of nonoptimal neighbors.

In the following theorem, we propose a necessary and sufficient condition for the probabilities p_t for almost sure convergence of Algorithm 1.

Theorem 3.1: The graph generated by Algorithm 1 converges to a gradient topology $(X(t) = 0)$ with probability 1 if and only if

$$\lim_{T \to \infty} \prod_{t=0}^{T} (1 - p_t) = 0.$$

Before proving Theorem 3.1, let us take a closer look at Algorithm 1, and notice especially that the stochastic process in (1) for $X(t)$ has the Markov property, hence we can describe it as a simple Markov chain (see Fig. 2).

Let $\boldsymbol{\pi}(t)$ denote the row vector of probabilities for the states $X(t)$, i.e.,

$$\pi_i(t) = \mathbb{P}\left[X(t) = D - i\right].$$

Remark: In this paper, we are using a zero-based indexing for $\boldsymbol{\pi}$, i.e., $\boldsymbol{\pi} = [\pi_0, \ldots, \pi_D]$ for notational simplicity.

The evolution of $\boldsymbol{\pi}(t)$ can be written in matrix form as

$$\boldsymbol{\pi}(t+1) = \boldsymbol{\pi}(t) P_t \tag{2}$$

where P_t is the transition matrix at time t

$$P_t = \begin{bmatrix} 1 - Dp_t & Dp_t & 0 & \cdots & 0 \\ 0 & 1 - (D-1)p_t & (D-1)p_t & \cdots & 0 \\ 0 & 0 & 1 - (D-2)p_t & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}.$$

Since P_t is a triangular matrix, the eigenvalues are given by the diagonal elements, i.e., the eigenvalues of P_t are $\lambda_i(t) = 1 - (D-i)p_t$, $i = 0, \ldots, D$. Notice that there is a single unit eigenvalue $\lambda_D(t) = 1$, and all other eigenvalues are strictly less than 1. Furthermore, all eigenvalues are distinct, hence the eigenvectors form a basis for $\mathbb{R}^{D+1}$. In the following lemma, we characterize the eigenvectors.

Lemma 3.2: The left-eigenvector $\xi^i(t)$ corresponding to eigenvalue $\lambda_i(t)$ is independent of $p_t \neq 0$, for $i = 0, \ldots, D$.

Proof: The left eigenvectors of P_t satisfy $\lambda_i(t)\xi^i(t) = \xi^i(t)P_t$. Let $\xi^i_j(t)$ denote the jth component of $\xi^i(t)$, then, by

inspection of the matrix P_t, we have the system of equations

$$\begin{aligned}(1-(D-i)p_t)\,\xi_0^i(t) &= (1-Dp_t)\,\xi_0^i(t)\\ (1-(D-i)p_t)\,\xi_j^i(t) &= (1-(D-j)p_t)\,\xi_j^i(t)\\ &\quad +(D-j+1)p_t\xi_{j-1}^i(t) \quad j=1,\dots,D\end{aligned}$$

which is equivalent to

$$\begin{aligned}i\xi_0^i(t) &= 0\\ (i-j)\xi_j^i(t) &= (D-j+1)\xi_{j-1}^i(t) \qquad j=1,\dots,D\end{aligned}$$

or further

$$\begin{aligned}\xi_j^i(t) &= 0 && \text{if } j<i\\ \xi_j^i(t)\frac{i-j}{D-j+1} &= \xi_{j-1}^i(t) && \text{if } j>i\end{aligned} \tag{3}$$

where $\xi_i^i(t)$ can be chosen as an arbitrary nonzero value, as a scaling factor for the eigenvector. From (3), it is evident that the eigenvectors are independent of p_t. ■

An important consequence of Lemma 3.2 is that all P_t, independent of t, have the same eigenvectors, and are thus simultaneously diagonalizable. Hence, we can simplify the notation by dropping the parameter t from the eigenvectors ξ^i.

Let us now return to the initial probability distribution $\boldsymbol{\pi}(0)$, and decompose it into the eigenvector basis as

$$\boldsymbol{\pi}(0) = \sum_{i=0}^{D} \alpha_i \xi^i \tag{4}$$

for some real numbers α_i.

Lemma 3.3: In the decomposition of the initial probability distribution $\boldsymbol{\pi}(0)$ into the eigenvector basis, we have $\alpha_D \xi^D = \mathbf{e}_D$, where $\mathbf{e}_i$ is the standard basis $\mathbf{e}_i = [0,\dots,0,\underbrace{1}_{i\text{th}},0,\dots,0]$.

Proof: Let us consider $\xi^i 11$ for $i=0,\dots,D-1$. By (3)

$$\xi^i 11 = \sum_{j=0}^{D}\xi_j^i = \sum_{j=i}^{D}\xi_j^i = \sum_{j=0}^{D-i}\xi_{i+j}^i.$$

We will show by induction that

$$\sum_{j=0}^{k}\xi_{i+j}^i = \frac{D-i-k}{D-i}\xi_{i+k}^i. \tag{5}$$

The case when $k=0$ is clearly true. Thus, assume that (5) holds for k and consider the case $k+1$

$$\begin{aligned}\sum_{j=0}^{k+1}\xi_{i+j}^i &= \sum_{j=0}^{k}\xi_{i+j}^i + \xi_{i+k+1}^i\\ &= \frac{D-i-k}{D-i}\xi_{i+k}^i + \xi_{i+k+1}^i\\ &= \frac{D-i-k}{D-i}\frac{-(k+1)}{D-i-k}\xi_{i+k+1}^i + \xi_{i+k+1}^i\\ &= \frac{D-i-(k+1)}{D-i}\xi_{i+k+1}^i.\end{aligned}$$

Using (5) implies that $\xi^i 11 = 0$, $i=0,\dots,D-1$, and thus, $\boldsymbol{\pi}(0)11 = \alpha_D\xi^D 11$. Now, since $\boldsymbol{\pi}(0)$ is a probability distribution, we know that $\boldsymbol{\pi}(0)11=1$, but (3) tells us that only the last component of ξ^D is nonzero, hence the lemma follows. ■

We are now ready to prove the main theorem.

Proof of Theorem 3.1: The almost sure convergence to a gradient topology, by Definition 2.1, can be expressed as

$$\lim_{T\to\infty}\mathbb{P}\left[X(T)=0\right]=1$$

or, equivalently for the probability vector

$$\lim_{T\to\infty}\boldsymbol{\pi}(T) = \mathbf{e}_D.$$

Equations (2) and (4) give us

$$\begin{aligned}\boldsymbol{\pi}(T) &= \boldsymbol{\pi}(0)\prod_{t=0}^{T-1}P_t\\ &= \sum_{i=0}^{D}\alpha_i\xi^i\prod_{t=0}^{T-1}P_t\\ &= \sum_{i=0}^{D}\alpha_i\xi^i\prod_{t=0}^{T-1}\lambda_i(t)\\ &= \sum_{i=0}^{D-1}\alpha_i\xi^i\prod_{t=0}^{T-1}\lambda_i(t) + \mathbf{e}_D.\end{aligned} \tag{6}$$

Consider the limit

$$\begin{aligned}\lim_{T\to\infty}|\boldsymbol{\pi}(T)-\mathbf{e}_D| &= \lim_{T\to\infty}\left|\sum_{i=0}^{D-1}\alpha_i\xi^i\prod_{t=0}^{T-1}\lambda_i(t)\right|\\ &\le \sum_{i=0}^{D-1}\left|\alpha_i\xi^i\right|\cdot\lim_{T\to\infty}\prod_{t=0}^{T-1}(1-p_t).\end{aligned}$$

Clearly, the right-hand side vanishes if $\lim_{T\to\infty}\prod_{t=0}^{T}(1-p_t)=0$. This proves the sufficiency part of the theorem.

Furthermore, for the necessity part, note that the set of initial probability distributions spawns $\mathbb{R}^{D+1}$. Thus, an initial probability distribution $\boldsymbol{\pi}(0)$ exists such that $\alpha_{D-1}\neq 0$. Assume that the limit $\lim_{T\to\infty}\prod_{t=0}^{T}(1-p_t)=c>0$ is strictly positive (the limit exists since it is a monotone bounded sequence), then

$$\begin{aligned}&\lim_{T\to\infty}|\boldsymbol{\pi}(T)-\mathbf{e}_D|\\ &\quad=\left|\sum_{i=0}^{D-2}\alpha_i\xi^i\left(\lim_{T\to\infty}\prod_{t=0}^{T-1}\lambda_i(t)\right)+c\alpha_{D-1}\xi^{D-1}\right|>0\end{aligned} \tag{7}$$

since the eigenvectors are linearly independent. Thus, we have proven the theorem. ■

Corollary 3.1: The graph generated by Algorithm 1 converges to a gradient topology with probability 1 if and only if

$$\lim_{T\to\infty}\sum_{t=0}^{T}p_t=\infty.$$

Proof: This follows directly from Theorem 3.1, and the relation

$$\lim_{T\to\infty}\prod_{t=0}^{T}(1-p_t)=0 \quad \Leftrightarrow \quad \lim_{T\to\infty}\sum_{t=0}^{T}p_t=\infty$$

for $0<p_t<1$ (from the Borel–Cantelli lemma [30]). ■

Remark: Corollary 3.1 can be interpreted such that the network converges to a gradient topology if and only if each node continues searching for its optimal neighbors for an expected infinite number of times.

IV. Convergence Rate Estimation

We will now investigate the convergence rate of $X(t)$, with a constant sampling probability $p_t=p$. Define the stochastic variable T_i as the time when $X(t)$ reaches 0, when starting at $X(0)=i$,

$$T_i=\inf_t[X(t)=0\mid X(0)=i].$$

Further, let $M_i=\mathbb{E}[T_i]$ denote the expected convergence time when starting at $X(0)=i$. Clearly, $M_0=0$, and for $i=1,\ldots,D$, we have the recursion

$$\begin{aligned}M_i&=1+\mathbb{P}\left[X(t+1)=i-1\mid X(t)=i\right]\cdot M_{i-1}\\&\quad+\mathbb{P}\left[X(t+1)=i\mid X(t)=i\right]\cdot M_i\\&=1+ipM_{i-1}+(1-ip)M_i\end{aligned}$$

which can be further simplified to

$$M_i=\frac{1+ipM_{i-1}}{ip}=\frac{1}{ip}+M_{i-1}.$$

By continuing with induction, we can sum up the expected convergence time as

$$M_i=\frac{1}{p}\sum_{d=1}^{i}\frac{1}{d}.$$

The worst initial case is when $X(0)=D$, where the expected convergence time is

$$M_D=\frac{1}{p}\sum_{d=1}^{D}\frac{1}{d}\le\frac{1+\ln(D)}{p}. \tag{8}$$

Remark: M_D is the expected time for an individual node's neighbor set to converge, not the expected time for *all* nodes to converge to a gradient topology. As such, it provides a lower bound on the convergence time. In Section IV-A, we will consider the global network convergence problem.

A. Global Convergence Rate

In this section, we will analyze the asymptotic convergence rate for the entire network to a gradient topology, in contrast to the analysis of a single node in Section III. We continue assuming a constant sampling probability ($p_t=p$, $P_t=P$), thus the probability distribution for a single node in (6) is simplified to

$$\boldsymbol{\pi}(t)=\boldsymbol{\pi}(0)P^t=\sum_{i=0}^{D-1}\alpha_i\xi^i\lambda_i^t+\mathbf{e}_D$$

where $\lambda_i=1-(D-i)p$, $i=0,\ldots,D$. The probability distribution for a single node approaches the gradient topology state $\mathbf{e}_D$ asymptotically as $1-\mathcal{O}\left(\lambda_{D-1}^t\right)$, where $\lambda_{D-1}=1-p$ is the second largest eigenvalue of P.

Here, we will study how the entire network convergence is affected by the network size N, and to this end, we consider a continuous-time approximation with a system for which the probability of being in the target state is $1-\lambda^t$ at time t (where $\lambda=1-p$).

Theorem 4.1: The expected global convergence time to a gradient topology for N nodes is

$$-\frac{1}{\log(\lambda)}\sum_{n=1}^{N}\frac{1}{n}$$

where the nodes are modeled by i.i.d. (independent identically distributed) processes, whose individual probability distribution is given by $1-\lambda^t$.

Proof: The probability for the entire system of N i.i.d. processes to be in the target state at time t is $\phi\doteq(1-\lambda^t)^N$. Notice that the probability for the system to reach the target state at time t is given by the derivative $\frac{d\phi}{dt}=-N(1-\lambda^t)^{N-1}\lambda^t\log(\lambda)$. The expected convergence time, i.e., the time to reach the gradient topology, is computed by

$$\int_0^\infty t\cdot\frac{d\phi}{dt}\,dt=-N\log(\lambda)\int_0^\infty t(1-\lambda^t)^{N-1}\lambda^t\,dt.$$

Using a variable substitution $x=\lambda^t$, this integral can be rewritten as

$$\int_0^\infty t(1-\lambda^t)^{N-1}\lambda^t\,dt=-\frac{1}{\log(\lambda)^2}\int_0^1\log(x)\,(1-x)^{N-1}dx.$$

Recall that this integral is equal to [31]

$$\int_0^1\log(x)\,(1-x)^{N-1}\,dx=-\frac{1}{N}\sum_{n=1}^{N}\frac{1}{n}.$$

Hence, the expected convergence time is

$$\int_0^\infty t\cdot\frac{d\phi}{dt}\,dt=-\frac{1}{\log(\lambda)}\sum_{n=1}^{N}\frac{1}{n}. \tag{9}$$

■

Remark: Notice that $-\frac{1}{\log(\lambda)}=-\frac{1}{\log(1-p)}\approx\frac{1}{p}$ for small p, thus this is in agreement with the upper bound in (8).

Remark: The convergence time scales asymptotically as $\mathcal{O}(\log(N))$ for large network sizes N since $\sum_{n=1}^{N}\frac{1}{n}<1+\log(N)$.

V. Convergence Rate With Network Churn

In this section, we consider the topology convergence to a gradient topology when the system is subject to churn, i.e., the nodes are changing over time. We model the churn as a probability $\epsilon>0$ that a node will be replaced with a new node that is starting from state $X=D$. The corresponding Markov chain for a single node is illustrated in Fig. 3.

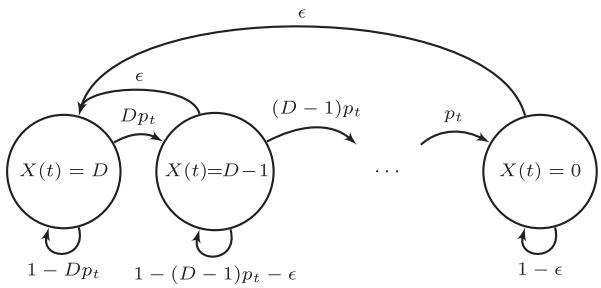

Fig. 3. Markov chain for the state evolution of a single node with churn.

The corresponding transition matrix P for the Markov chain $\boldsymbol{\pi}(t+1)=\boldsymbol{\pi}(t)P$ is

$$P=\begin{bmatrix} 1-Dp & Dp & 0 & \cdots & 0 \\ \epsilon & 1-(D-1)p-\epsilon & (D-1)p & \cdots & 0 \\ \epsilon & 0 & 1-(D-2)p-\epsilon & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \epsilon & 0 & 0 & \cdots & 1-\epsilon \end{bmatrix}.$$

Assuming that $0<\epsilon$, and also $0<Dp<1$ and $(D-1)p+\epsilon<1$, we have the following theorem characterizing the stationary distribution.

Theorem 5.1: The Markov chain in Fig. 3, describing the stochastic node process with churn, has a unique stationary distribution $\boldsymbol{\pi}$, which satisfies

$$\pi_0=\frac{\epsilon}{Dp+\epsilon}$$

and

$$\pi_i=\frac{(D-i+1)p}{(D-i)p+\epsilon}\pi_{i-1} \quad i=1,\ldots,D.$$

Proof: Notice that the Markov chain is finite ($D+1$ states), irreducible (every state can be reached from any other state), and aperiodic (because of the self-loops), thus it has a unique stationary distribution corresponding to the eigenvalue 1.

Consider now, the stationary distribution $\boldsymbol{\pi}$ satisfying $\boldsymbol{\pi}=\boldsymbol{\pi}P$, and especially for column $i=1,\ldots,D$ we have

$$\pi_i=(1-(D-i)p-\epsilon)\pi_i+(D-i+1)p\pi_{i-1}$$

that is

$$\pi_i=\frac{(D-i+1)p}{(D-i)p+\epsilon}\pi_{i-1}.$$

Next, let us show the following property for the partial sum $\pi_d+\cdots+\pi_D$ through induction:

$$\sum_{i=d}^{D}\pi_i=(D-d+1)\frac{p}{\epsilon}\pi_{d-1} \quad d=1,\ldots D.$$

The case $d=D$ follows directly from the previous recursion. Let us continue with the following induction step:

$$\sum_{i=d}^{D}\pi_i=\pi_d+\sum_{i=d+1}^{D}\pi_i=\pi_d+(D-d)\frac{p}{\epsilon}\pi_d$$
$$=\frac{(D-d)p+\epsilon}{\epsilon}\pi_d=\frac{(D-d)p+\epsilon}{\epsilon}\frac{(D-d+1)p}{(D-d)p+\epsilon}\pi_{d-1}$$
$$=(D-d+1)\frac{p}{\epsilon}\pi_{d-1}.$$

First, we use this to validate that the eigenvector satisfies $\boldsymbol{\pi}=\boldsymbol{\pi}P$ for the first column

$$\pi_0=(1-Dp)\pi_0+\epsilon\sum_{d=1}^{D}=(1-Dp)\pi_0+\epsilon D\frac{p}{\epsilon}\pi_0=\pi_0.$$

Second, the stationary probability distribution should be normalized such that

$$\sum_{i=0}^{D}\pi_i=1.$$

Thus

$$\sum_{i=0}^{D}\pi_i=\pi_0\sum_{i=1}^{D}\pi_i=\pi_0+D\frac{p}{\epsilon}\pi_0=\frac{Dp+\epsilon}{\epsilon}\pi_0$$

or

$$\pi_0=\frac{\epsilon}{Dp+\epsilon}$$

which proves the theorem. ■

Remark: Theorem 5.1 shows that if $p=\epsilon$, then the stationary distribution is uniform with $\pi_i=\frac{1}{D+1}$, $i=0,\ldots,D$; thus, a node is equally likely to be in any state. When $p>\epsilon$, the nodes are more likely to be in the later states, i.e., closer to a gradient topology, and when $p<\epsilon$, the nodes are more likely to be in the earlier states, i.e., having a random neighbor set. Thus, we conclude that for a gradient topology to appear, it is necessary for the sampling probability p to be greater than the churn rate ϵ.

Next, the convergence speed will be considered through analyzing the second largest eigenvalue of the transition matrix P.

Lemma 5.2: The asymptotic convergence time for the entire network with churn is

$$\log(N)\frac{1}{p+\epsilon}.$$

Proof: It is straightforward to verify that the remaining eigenvalues of P (less than 1) are

$$\lambda_i=1-(D-i)p-\epsilon$$

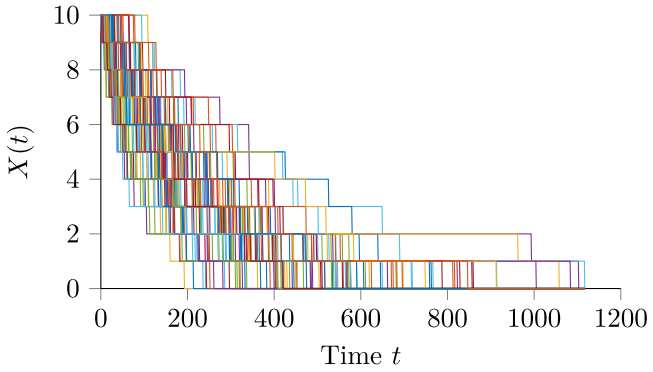

Fig. 4. State trajectories for a network with 100 nodes and degree $D = 10$. Each line represents a single node, and a constant sampling probability $Np_t = 1/2$ is used. The network converges to a gradient topology.

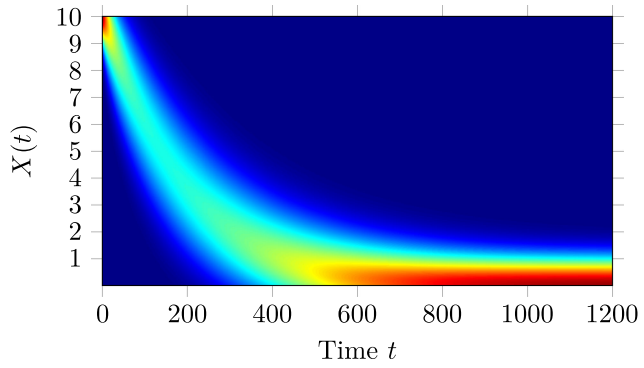

Fig. 5. State heat map for a network with 100 nodes, degree $D = 10$, and constant sampling probability $Np_t = 1/2$. Brighter colors indicate more likely states.

for $i = 0, \ldots, D-1$, with the corresponding eigenvectors

$$\xi_i = \Bigg[\underbrace{0, \ldots, 0}_{i}, (-1)^0 \binom{D-i}{0}, (-1)^1 \binom{D-i}{1}, \ldots, (-1)^{D-i} \binom{D-i}{D-i}\Bigg].$$

Hence, the second largest eigenvalue of P is $\lambda_{D-1} = 1 - p - \epsilon$.

Using Theorem 4.1, with $\lambda = 1 - p - \epsilon$, and the approximations $\sum_{n=1}^{N} \frac{1}{n} \approx \log(N)$ and $-\frac{1}{\log(\lambda)} = -\frac{1}{\log(1-p-\epsilon)} \approx \frac{1}{p+\epsilon}$ proves this lemma. ■

Remark: A larger ϵ will yield a faster convergence rate, but to a steady-state solution further from the complete gradient topology.

VI. Convergence Simulation

Here, we examine the convergence rate of Algorithm 1 using numerical simulations, and compare the outcome with our theoretical results. We start with a network consisting of $N = 100$ nodes, where the degree of each node is $D = d_i = 10$. The similar view $\mathcal{N}_i^s(0)$ is initialized with D nodes uniformly chosen among all nodes in the network, and the sampling probability p_t is held at a constant value of $\frac{1}{2N}$. Hence, for each node and at each iteration of the algorithm, the random view is empty with 50% probability. The state trajectories for all nodes are shown in Fig. 4, and the convergence times ranges from 193 to 1116 iterations, with an average convergence time of 554 iterations. The convergence time can be compared to the expected convergence time given by (8) for a single node, i.e., 585 iterations, and the global convergence rate given by (9), i.e., 1035 iterations. A corresponding heat map of the states are shown in Fig. 5. The system converges to a gradient topology, as guaranteed by Theorem 3.1.

In the second simulation, we change the sampling probability into a decaying probability $p_t = \frac{1}{N} \frac{1}{(1+t/100)^2}$. Notice that $\sum_{t=0}^{\infty} Np_t < 101$, hence, by Corollary 3.1, there is a positive probability that the algorithm does not converge to a gradient topology. This is also confirmed by the simulation trajectories in Fig. 6 and the corresponding state heat map in Fig. 7.

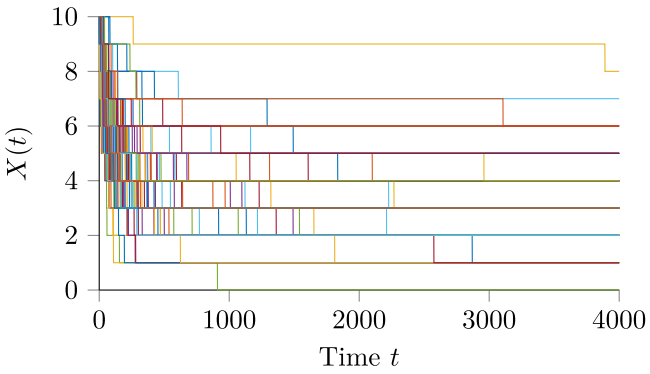

Fig. 6. State trajectories for a network with 100 nodes and degree $D = 10$. Each line represents a single node, and a decaying sampling probability $Np_t = \frac{1}{(1+t/100)^2}$ is used. The network does not converge to a gradient topology.

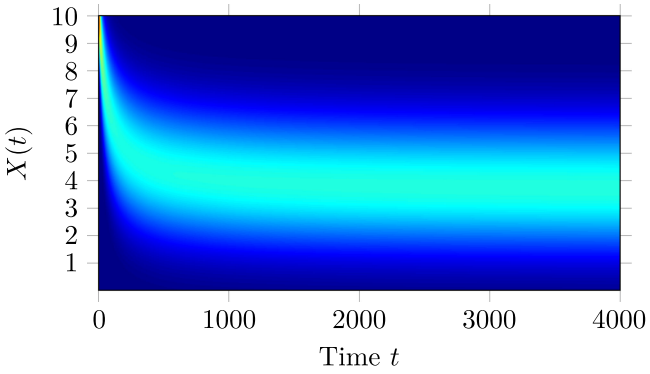

Fig. 7. State heat map for a network with 100 nodes, degree $D = 10$, and a decaying sampling probability $Np_t = \frac{1}{(1+t/100)^2}$. Brighter colors indicate more likely states. The network does not converge to a gradient topology.

In the third simulation, we return to the constant sampling probability $\frac{1}{2N}$, but consider a network with $N = 500$ nodes and a node degree of $D = 50$. The expected state heat map is shown in Fig. 8, and the convergence time can be compared to

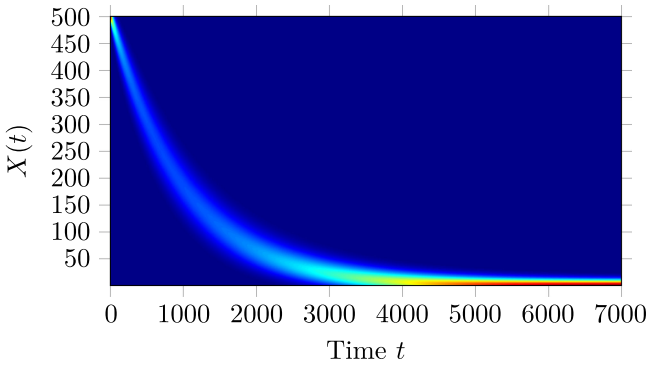

Fig. 8. State heat map for a network with 500 nodes, degree $D = 50$, and constant sampling probability $Np_t = 1/2$. Brighter colors indicate more likely states.

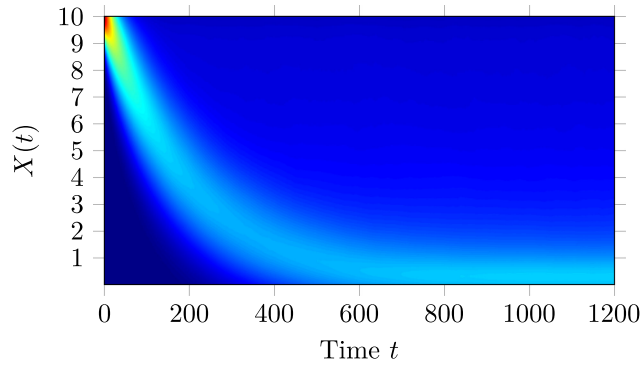

Fig. 9. State heat map for a network with churn, consisting of 100 nodes, degree $D = 10$, and a constant sampling probability $Np_t = 1/2$. Brighter colors indicate more likely states. The churn probability is $\epsilon = \frac{1}{2}p_t$, thus the network converges to a steady state close to a gradient topology.

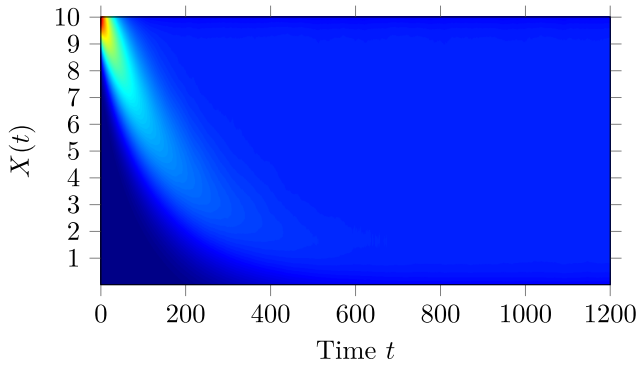

Fig. 10. State heat map for a network with churn, consisting of 100 nodes, degree $D = 10$, and a constant sampling probability $Np_t = 1/2$. Brighter colors indicate more likely states. The churn probability is $\epsilon = p_t$, thus the network converges to a steady state where every state is equally likely.

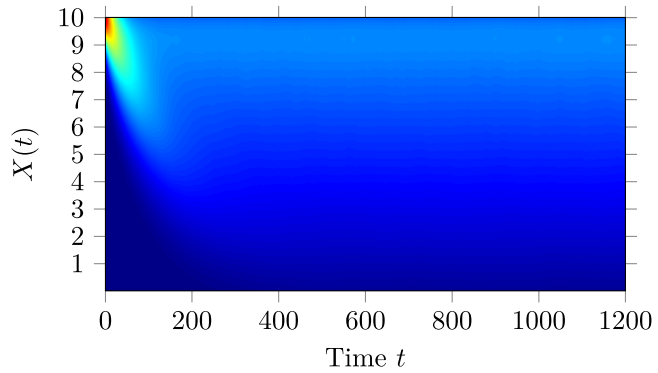

Fig. 11. State heat map for a network with churn, consisting of 100 nodes, degree $D = 10$, and a constant sampling probability $Np_t = 1/2$. Brighter colors indicate more likely states. The churn probability is $\epsilon = 2p_t$, thus the network converges to a steady state where the initial random neighborhood is more likely, and the gradient topology is missing.

the expected convergence time of 4499 iterations for a single node and 6789 iterations for the entire network.

Finally, we simulate the influence of churn on the network. Consider a network consisting of $N = 100$ nodes, with node degree $D = 10$ and a constant sampling probability $p_t = \frac{1}{2N}$. In Fig. 9, the churn rate is $\epsilon = \frac{1}{2}p_t$, and we see that nodes tend to favor states closer to a gradient topology, as predicted by Theorem 5.1. In fact, 27% of the nodes are in their optimal state $X(t) = 0$, with another 14% are in state $X(t) = 1$. In Fig. 10, the churn rate is increased to $\epsilon = p_t$, and all states are equally likely in the steady-state solution, whereas in Fig. 11, the churn rate is further increased to $\epsilon = 2p_t$ and nodes tend to have a more random neighborhood, with 17% of the nodes having a completely random neighborhood $X(t) = D$.

VII. Evaluating Live Streaming Using the Gradient Topology

Now, we turn to an evaluation of the effect of sampling nodes from the gradient overlay network compared to a random overlay network when building a P2P live-streaming application, called GLive. GLive is based on nodes cooperating to share a media stream supplied by a source node, and uses an approximate auction algorithm to match nodes that are willing and able to share the stream with one another. GLive extends the tree-based live streaming, gradienTv [25] and Sepidar [18], to mesh-based live streaming.

Nodes want to establish connections to other nodes that are as close as possible to the source. They bid for connections to the best neighbors using their own upload bandwidth, and nodes share their bounded number of connections with the nodes that bid the highest (contribute the most upload bandwidth). Auctions are continuous and restarted on failures or free riding. The desired effect of the auction algorithm is that the source will upload to the nodes that contribute the most upload bandwidth, which will, in turn, upload to the nodes that contribute the next highest amount of bandwidth, and so on until the topology is fully constructed.

One of the main problems with the lack of global information about nodes' upload bandwidths is that it affects the rate of convergence of the auction algorithm. Nodes would ideally like to bid for connections to other nodes that they can afford to connect to, rather than win a connection to a better node and later be removed because a better bid was received. The traditional way to discover nodes to bid on is using a uniform random

peer-sampling service [8]. Instead, we use the gradient overlay to sample nodes, where a node's utility value is the upload bandwidth it contributes to the system. As such, the gradient should provide other nodes with references to nodes that have well-matched upload bandwidths. Payberah *et al.* [18] showed that using the gradient overlay network reduced the rate of parent switching for tree-based live streaming by 20% compared to random peer sampling. Here, we show for GLive, the effect of sampling neighbors using random peer sampling (GLive/Random) versus sampling from the gradient overlay (GLive/Gradient).

GLive is implemented using Kompics' discrete-event simulator [27] that provides several bandwidth, latency, and churn models. In our experimental setup, we set the streaming rate to 512 kb/s, which is divided into blocks of 16 kB. Nodes start playing the media after buffering it for 5 s. The size of the similar view in GLive is 15 nodes, and in the auction algorithm, nodes have 8 download connections. To model the upload bandwidth, we assume that each upload connection has an available bandwidth of 64 kb/s and that the number of upload connections for the nodes is set to $2i$, where i is picked randomly from the range 1 to 10. This means that nodes have an upload bandwidth capacity between 128 kb/s and 1.25 Mb/s. As the average upload bandwidth of 704 kb/s is not much higher than the streaming rate of 512 kb/s, nodes need to find good parents to achieve the streaming performance. The media source is a single node with 40 upload connections, providing 5 times the upload bandwidth of the stream rate. We assume 11 utility levels, such that nodes contributing the same amount of upload bandwidth are located at the same utility level. Latencies between nodes are modeled using a latency map based on the King dataset [32]. We assume that the size of the sliding window for downloading is 32 blocks, such that the first 16 blocks are considered as the in-order set and the next 16 blocks are the blocks in the rare set. A block is chosen for download from the in-order set with 90% probability, and from the rare set with 10% probability. In the experiments, we measure the following metrics.

1) *Playback continuity*: The percentage of blocks that a node received before their playback time. We consider the case where nodes have a playback continuity greater than 99%.
2) *Playback latency*: the difference in seconds between the playback point of a node and the playback point at the media source.

We compare the playback continuity and playback latency of GLive/Gradient and GLive/Random in the following three scenarios.

1) *Churn*: In total, 500 nodes join the system following a Poisson distribution, with an average interarrival time of 100 ms. Then, until the end of the simulation, nodes join and fail continuously following the same distribution with an average interarrival time of 1000 ms.
2) *Flash crowd*: In total, 100 nodes join the system following a Poisson distribution with an average interarrival time of 100 ms. After 150 s, 1000 nodes join following the same distribution with a shortened average interarrival time of 10 ms.
3) *Catastrophic failure*: In total, 1000 nodes join the system following a Poisson distribution with an average interarrival time of 100 ms. After 150 s, 500 existing nodes fail following a Poisson distribution with an average inter-failure time of 10 ms.

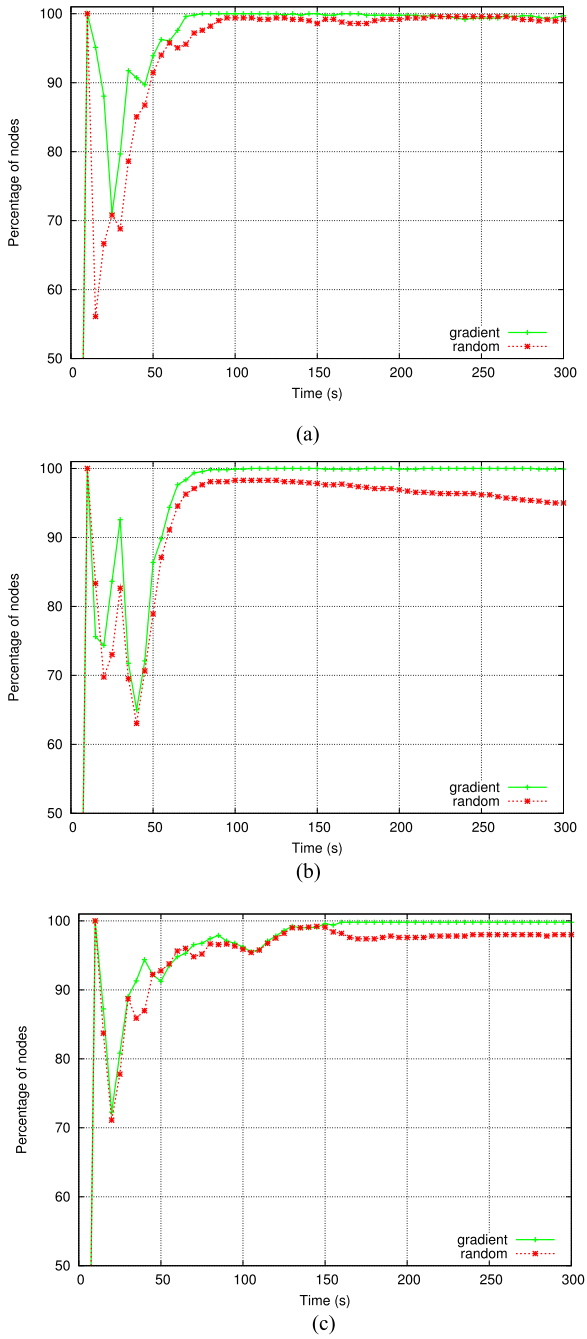

Fig. 12. Playback continuity of the GLive/Gradient and GLive/Random systems in the churn, flash crowd, and catastrophic failure scenarios. (a) Churn. (b) Flash crowd. (c) Catastrophic failure.

Fig. 12 shows the percentage of the nodes that have a playback continuity of at least 99%. We see that all the nodes in GLive/Gradient receive at least 99% of all the blocks very quickly in all scenarios, whereas it takes slightly more time for GLive/Random. This is because nodes in GLive/Gradient find a good set of matches faster than nodes in GLive/Random by running the auction algorithm against nodes with similar upload

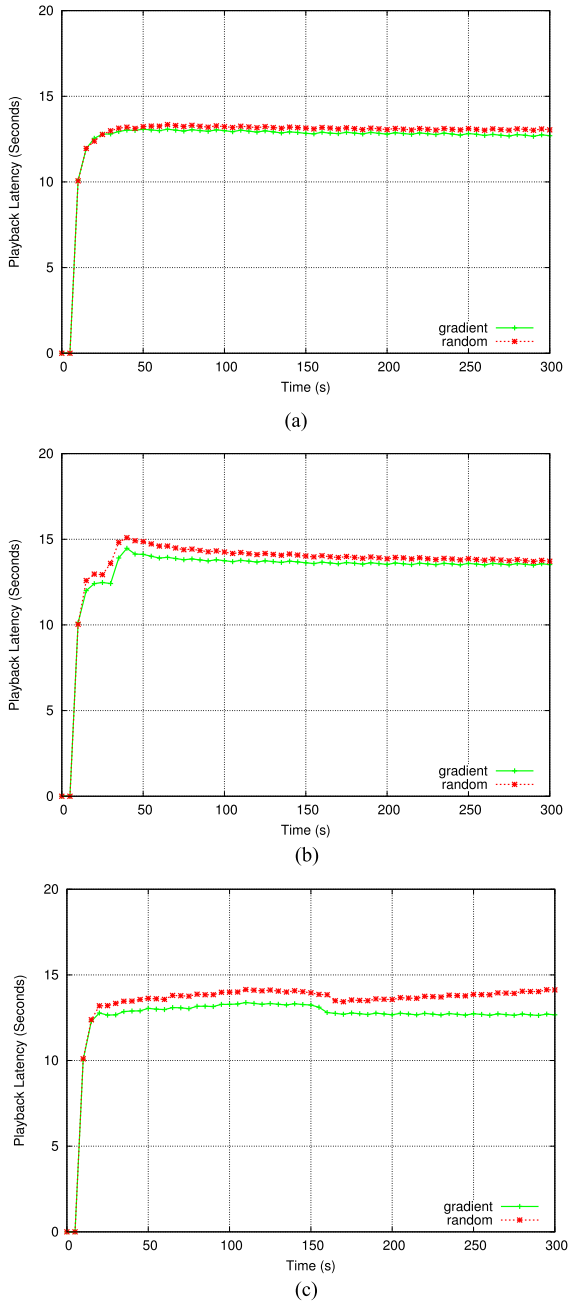

Fig. 13. Playback latency of the GLive/Gradient and GLive/Random systems in the churn, flash crowd, and catastrophic failure scenarios. (a) Churn. (b) Flash crowd. (c) Catastrophic failure.

bandwidth. One point to note is that the 5 s of buffering cause the spike in playback continuity at the start, which then drops off as nodes are joining the system. To summarize, using the gradient overlay instead of random sampling produces better performance when the system is undergoing large changes—such as a large numbers of nodes joining or failing over a short period of time.

Fig. 13 shows the playback latency of the systems in the different scenarios. As we can see, although there is only a small difference between the systems, GLive/Gradient consistently maintains relatively shorter playback latency than GLive/Random for all experiments. The playback latency includes both the 5 s buffering time and the time required to pull the blocks over the live-streaming overlay constructed using the auction algorithm.

VIII. Conclusion

In this paper, we studied the network topology convergence problem for the gossip-generated gradient overlay network. A necessary and sufficient condition for convergence to a complete gradient topology was shown in terms of the neighbor sampling probabilities. Further, the expected convergence time was characterized for a single node, and extended to an asymptotic convergence rate estimate for the entire network. Finally, networks with churn were considered, and a threshold on the churn rate was derived for a gradient topology to emerge.

Live-streaming experiments showed the potential advantages of network topologies built using preference functions. We showed how nodes can use implicit information, captured in the gradient topology, to efficiently find suitable neighbors compared to using random sampling. As such, this paper on proving convergence properties of the gradient topology could have significance for other future information-carrying topologies.

Acknowledgment

The authors would like to thank A. H. Payberah and J. Dowling from the Swedish Institute of Computer Science for their valuable feedback, and the development and evaluation of the GLive live-streaming application, and G. Shi and A. Gattami for valuable discussions.

References

[1] Cisco, "Cisco visual networking index: Forecast and methodology, 2014–2019," Cisco Syst., San Jose, CA, USA, Tech. Rep., FLGD 12352, May 2015.

[2] S. Androutsellis-Theotokis and D. Spinellis, "A survey of peer-to-peer content distribution technologies," *ACM Comput. Surv.*, vol. 36, no. 4, pp. 335–371, Dec. 2004.

[3] J. Risson and T. Moors, "Survey of research towards robust peer-to-peer networks: Search methods," *Comput. Netw.*, vol. 50, no. 17, pp. 3485–3521, Dec. 2006.

[4] E. Meshkova, J. Riihijärvi, M. Petrova, and P. Mähönen, "A survey on resource discovery mechanisms, peer-to-peer and service discovery frameworks," *Comput. Netw.*, vol. 52, no. 11, pp. 2097–2128, Aug. 2008.

[5] Y. Liu, Y. Guo, and C. Liang, "A survey on peer-to-peer video streaming systems," *Peer-to-Peer Netw. Appl.*, vol. 1, no. 1, pp. 18–28, Jan. 2008.

[6] S. M. Thampi, "A review on P2P video streaming," arXiv:1304.1235, Apr. 2013.

[7] M. Jelasity, R. Guerraoui, A.-M. Kermarrec, and M. van Steen, *The Peer Sampling Service: Experimental Evaluation of Unstructured Gossip-Based Implementations*. New York, NY, USA: Springer-Verlag, Oct. 2004.

[8] M. Jelasity, S. Voulgaris, R. Guerraoui, A.-M. Kermarrec, and M. van Steen, "Gossip-based peer sampling," *ACM Trans. Comput. Syst.*, vol. 25, no. 3, Aug. 2007, Art. no. 8.

[9] A. H. Payberah, J. Dowling, and S. Haridi, "Gozar: NAT-friendly peer sampling with one-hop distributed NAT traversal," in *6th International Federated Conferences on Distributed Computing Techniques*. Berlin, Germany: Springer, 2011, pp. 1–14.

[10] R. Baldoni, M. Platania, L. Querzoni, and S. Scipioni, "Practical uniform peer sampling under churn," in *Proc. IEEE 9th Int. Symp. Parallel Distrib. Comput.*, 2010, pp. 93–100.

[11] A.-M. Kermarrec and M. van Steen, "Gossiping in distributed systems," *ACM SIGOPS Oper. Syst. Rev.*, vol. 41, no. 5, pp. 2–7, Oct. 2007.

[12] S. P. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2508–2530, Jun. 2006.
[13] A. Olshevsky and J. N. Tsitsiklis, "Convergence rates in distributed consensus and averaging," in *Proc. 45th IEEE Conf. Decis. Control*, 2006, pp. 3387–3392.
[14] J. Liu, B. D. O. Anderson, M. Cao, and A. S. Morse, "Analysis of accelerated gossip algorithms," in *Proc. 48th IEEE Conf. Decis. Control/28th Chin. Control Conf.*, Shanghai, China, 2009, pp. 871–876.
[15] M. Mihail, C. Papadimitriou, and A. Saberi, "On certain connectivity properties of the internet topology," in *Proc. 44th Annu. IEEE Symp. Found. Comput. Sci.*, Washington, DC, USA, Oct. 2003, pp. 28–35.
[16] A.-L. Barabási, *Linked: The New Science of Networks*. New York City, New York, USA: Perseus, May 2002.
[17] G. Kreitz and F. Niemelä, "Spotify—Large scale, low latency, P2P music-on-demand streaming," in *Proc. 10th IEEE Int. Conf. Peer-to-Peer Comput.*, Delft, The Netherlands, 2010. [Online]. Available: http://ieeexplore.ieee.org/document/5569963/
[18] A. H. Payberah, F. Rahimian, S. Haridi, and J. Dowling, "Sepidar: Incentivized market-based P2P live-streaming on the gradient overlay network," in *Proc. 12th IEEE Int. Symp. Multimedia*, 2010, pp. 1–8.
[19] M. Jelasity, A. Montresor, and O. Babaoglu, "T-Man: Gossip-based fast overlay topology construction," *Comput. Netw.*, vol. 53, no. 13, pp. 2321–2339, 2009.
[20] D. Stutzbach and R. Rejaie, "Understanding churn in peer-to-peer networks," in *Proc. 6th ACM Conf. Internet Meas.*, New York, USA, 2006, pp. 189–202.
[21] P. Raftopoulou and E. G. M. Petrakis, "Peer rewiring in semantic overlay networks under churn," in *On the Move to Meaningful Internet Systems: OTM 2010*. Berlin, Germany: Springer, 2010, pp. 573–581.
[22] R. Baldoni, S. Bonomi, A. Rippa, L. Querzoni, S. T. Piergiovanni, and A. Virgillito, "Evaluation of unstructured overlay maintenance protocols under churn," in *Proc. 26th IEEE Int. Conf. Distrib. Comput. Syst. Workshops*, 2006, pp. 13–13.
[23] F. Kuhn, S. Schmid, and R. Wattenhofer, "A self-repairing peer-to-peer system resilient to dynamic adversarial churn," in *Proc. 4th Int. Workshop Peer-to-Peer Syst.* Berlin, Germany: Springer, 2005, pp. 13–23.
[24] F. Wang, J. Liu, and Y. Xiong, "Stable peers: Existence, importance, and application in peer-to-peer live video streaming," in *Proc. 27th IEEE Int. Conf. Comput. Commun.*, 2008, pp. 1364–1372.
[25] A. H. Payberah, J. Dowling, F. Rahimian, and S. Haridi, "gradienTv: Market-based P2P live media streaming on the gradient overlay," *Distrib. Appl. Interoperable Syst.*, vol. 6115, pp. 212–225, 2010.
[26] S. Voulgaris, D. Gavidia, and M. van Steen, "CYCLON: Inexpensive membership management for unstructured P2P overlays," *J. Netw. Syst. Manage.*, vol. 13, no. 2, pp. 197–217, 2005.
[27] C. Arad, J. Dowling, and S. Haridi, "Developing, simulating, and deploying peer-to-peer systems using the kompics component model," in *Proc. 4th ACM Int. Conf. Commun. Syst. Softw. Middleware*, Dublin, Ireland, 2009.
[28] H. Terelius, G. Shi, J. Dowling, A. H. Payberah, A. Gattami, and K. H. Johansson, "Converging an overlay network to a gradient topology," in *Proc. 50th IEEE Conf. Decis. Control Eur. Control Conf.*, Orlando, FL, USA, Dec. 2011, pp. 7230–7235.
[29] J. Sacha, "Exploiting heterogeneity in peer-to-peer systems using gradient topologies," Ph.D. dissertation, Trinity College, University of Dublin, Dublin, 2009.
[30] F. T. Bruss, "A counterpart of the Borel-Cantelli lemma," *J. Appl. Probab.*, vol. 17, no. 4, pp. 1094–1101, Dec. 1980.
[31] A. Devoto and D. W. Duke, "Table of integrals and formulae for Feynman diagram calculations," *La Rivista Del Nuovo Cimento*, vol. 7, no. 6, pp. 1–39, Jun. 1984.
[32] K. P. Gummadi, S. Saroiu, and S. D. Gribble, "King: Estimating latency between arbitrary internet end hosts," in *Proc. 2nd ACM SIGCOMM Workshop Internet Meas.*, 2002, pp. 5–18.

Håkan Terelius received the M.Sc. degree in engineering physics in 2010 and the Ph.D. degree in electrical engineering in 2016 from the Royal Institute of Technology, Stockholm, Sweden.

Karl Henrik Johansson (F'00) received the M.Sc. and Ph.D. degrees in electrical engineering from Lund University, Lund, Sweden, in 1992 and 1997, respectively.

He is the Director of the ACCESS Linnaeus Centre and a Professor with the School of Electrical Engineering, Royal Institute of Technology, Stockholm, Sweden. He is a Wallenberg Scholar and has held a Senior Researcher Position with the Swedish Research Council. He has held visiting positions with The University of California, Berkeley (1998–2000) and the California Institute of Technology (2006–2007). His research interests include networked control systems, hybrid and embedded control, and control applications in automotive, automation and communication systems.