Analysis of Realized Peer-to-Peer Streaming Topologies by Kronecker Graphs

Udo R. Krieger, Philipp Eittenberger
Faculty of Information Systems and Applied Computer Science
Otto-Friedrich-University
D-96047 Bamberg, Germany
Email: udo.krieger@uni-bamberg.de

Alex Borges Vieira
Computer Science Department
Universidade Federal de Juiz de Fora
Juiz de Fora, MG, 36036-900, Brazil
Email: alex.borges@ufjf.edu.br

***Abstract*—Modeling the overlay graph of peer-to-peer (P2P) data dissemination is inherently difficult due to the high dynamics of the peer behavior and the high degree of connectivity if we regard a mesh-pull architecture. We present a solution of the latter issue using the mathematical tools of Kronecker graphs. We are able to capture accurately the static structure of the overlay graph arising from a P2P streaming application. To validate our proposal, we use a large data set of a worldwide distributed measurement campaign arising from the live streaming system SopCast. First, we determine the basic parameters of our Kronecker graphs model by an EM-algorithm. Then we use it to generate a synthetic graph and compare the properties of both structures. The results of our analysis illustrate the coincidence of several graph-theoretical properties of the realized and synthetic overlay graph and underline the great potential of a modeling approach based on Kronecker graphs.**

***Index Terms* - P2P streaming, overlay topologies, Internet measurement, Kronecker graphs**

I. Introduction

A lot of scientific studies have investigated the structure and properties of peer-to-peer (P2P) overlays from a theoretical point of view. We may remember the related discussion on structured vs. unstructured overlay organization in the early days of P2P. However, still little knowledge has been generated about the graph-theoretical properties of real world P2P applications. There is no doubt that modeling complex graph structures produced by modern network applications is a difficult task (cf. [5], [11], [12]). Moreover, most of contemporary, popular P2P applications have proprietary protocols, which make it tremendously difficult to capture all properties of their related overlay topologies. In addition the number of clients involved in such modern applications increases every day, which makes it quite hard to gather efficiently a complete view of the underlying P2P system.

In this paper, we aim to understand the network structure of such a scenario and to get access to their realized overlay graphs. These items play a central role in computer and social network research [4]. As the performance of a P2P application may be affected by its overlay topology, it is crucial to understand it well enough in order to develop accurate models of the topology. Moreover, we study the generation of a synthetic topology derived from a monitored P2P streaming platform. More specifically, we propose the application of a theoretical overlay model based on Kronecker graphs to capture the inherent properties of P2P overlay structures.

Currently, most existing work on topology analysis and generation is not adequate and cannot be applied to environments that are in use now. Studies addressing the generation of a P2P topology from the early days research, for example, are not useful anymore. The reason simply is that the discussion about the construction of a structured vs. unstructured overlay and its maintenance is obsolete. Apart from BitTorrent, hardly any successful, worldwide deployed P2P application is using a DHT structure. Other reasons are that these studies may address applications which are no longer really popular, like Gnutella [5], or that the existing work does not properly address the overlay generation, but only simulates a P2P application protocol [12].

We highlight that the contribution of our study covers two important major aspects of P2P modeling. First, we introduce Kronecker graphs to describe the static topology of a P2P overlay network in time slices. Secondly, we evaluate whether the KronEM algorithm is an appropriate graph-theoretical tool to capture the overlay structure of a well-known P2P live streaming application.

II. Related Work

There exist a number of competing approaches regarding the construction of random graphs that can reproduce important properties of real networks. Unfortunately, these approaches do not capture many inherent aspects of real networks [3].

In this context Brite [11] is one of the most popular tools. It provides a wide variety of construction models focussing on the generation of Internet topology. Dimitropoulos et al. [3] also address the problem of generating synthetic graphs with realistic node relationships. The sole focus relies on modeling AS-relationships in the Internet. Jovanović et al. [5] report on the results of measurements w.r.t the large P2P application Gnutella. They show that Gnutella exhibits *small-world* properties and observe the properties of small diameter and clustering. Yang et al. [12] also address a P2P scenario. They present and validate a P2P simulator and develop an overlay model of BitTorrent. Finally, Gjoka et al. [4] propose a complete, practical methodology to generate graphs that resembles a real graph of interest. The generated topologies have the same properties as the target system, namely the joint degree distribution and the degree-dependent average clustering coefficient. Despite that these studies address the modeling and generation of overlay topologies, most of them are somewhat outdated or use no longer popular services like Gnutella and cannot be applied to current applications.

Here we propose the application of a generic new methodology to model the overlay graphs of P2P streaming sessions by Kronecker graphs (cf. [7]). In contrast to many other graph models, stochastic Kronecker graphs can offer the theoretical power to express large structures with heavy-tailed degree distributions that can very well approximate relevant properties of real networks. Moreover, they provide simple, efficient construction and analysis techniques derived from well-known primitives like Kronecker multiplication of adjacency matrices (cf. [7], [8]). This methodology is extensible, analytically and algorithmically tractable. It can be expanded to treat the generic case of complex graph structures evolving over time. Thus, it provides unique advantages compared to other topology models (cf. [11]).

III. Kronecker Graphs

Inspired by Leskovec et al. [7], we investigate the application of stochastic Kronecker graphs to model the overlay graph of P2P networks. Following closely the notation in [7], let us first briefly introduce the Kronecker graphs model.

Given two matrices $A \in \mathbb{R}^{m_1 \times n_1}$ and $B \in \mathbb{R}^{m_2 \times n_2}$, their Kronecker product $A \otimes B \in \mathbb{R}^{m_1 m_2 \times n_1 n_2}$ is defined by

$$A \otimes B = \begin{pmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{pmatrix}.$$

Leskovec et al. [7] define the *Kronecker product of two graphs* $G_A = (V_A, E_A), G_B = (V_B, E_B)$ with node sets V_A, V_B of sizes $|V_A| = n_A = m_1 = n_1, |V_B| = n_B = m_2 = n_2$, and edge sets E_A, E_B in terms of the Kronecker product of their underlying adjacency matrices $A = (A_{ij}) \in \mathbb{R}^{m_1 \times m_1}, B = (B_{ij}) \in \mathbb{R}^{m_2 \times m_2}$. The latter are defined by $x_{ij} = \begin{cases} 1 & (i,j) \in E_x \\ 0 & (i,j) \notin E_x \end{cases}$ for $i,j \in \{1,\ldots,n_x\}, x \in \{A,B\}$.

Then the model of the graph is determined by a small stochastic Kronecker parameter matrix $\Theta \in [0,1]^{\mathbb{N}_0 \times \mathbb{N}_0}$ representing an initial graph $G_{K(0)} = (V(0), E(0))$ of size $|V(0)| = N_0$, e.g. $N_0 = 2$. The entries of Θ can be regarded as probabilities Θ_{ij} of an edge $(i,j) \in E(0)$. By iteratively multiplying Θ with itself, one creates larger and larger stochastic adjacency matrices and related graphs $G_{K(k)}$. The latter are naturally self-similar [6]. The k-th power of Θ yields $\Theta^k \in [0,1]^{\mathbb{N}_0^k \times \mathbb{N}_0^k}$ and shall approximate a given monitored or theoretically constructed graph $G_M = (V_M, E_M)$ for $|V_M| = N, \lfloor \log_{N_0} N \rfloor = k \in \mathbb{N}$. Then each entry $(\Theta^k)_{ij}$ represents the probability that the corresponding edge $(i,j) \in E_{K(k)}$ exists in the constructed graph $K = (V_{K(k)}, E_{K(k)})$. Given a parameter matrix Θ, it is possible to construct in this manner Kronecker graphs $K \equiv K(k)$ by including edges $(i,j) \in E_K$ according to their computed probabilities $(\Theta^k)_{ij}$.

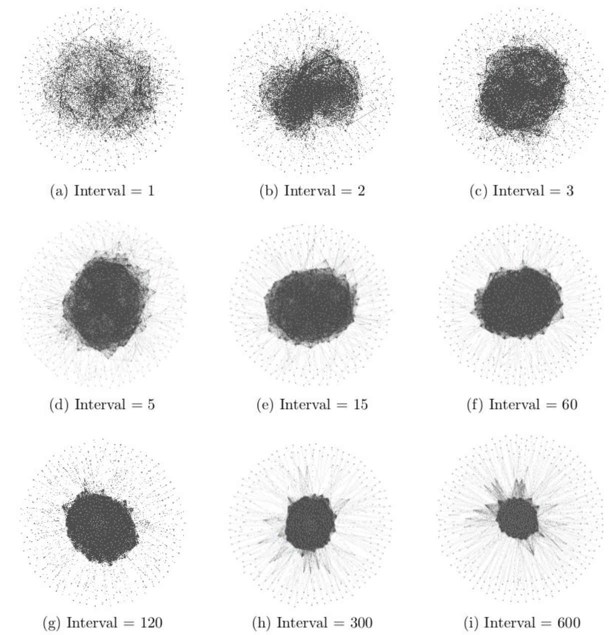

Figure 1. Evolution of the measured overlay graph that is visualized for different snapshot intervals (given in seconds). The same trace has been visualized by the Fruchtermann-Reingold graph drawing algorithm in all plots.

A major issue concerns the question how to fit the actual model parameters $\Theta = (\Theta_{ij})$ to some data set representing a monitored empirical graph $G_M = (V_M, E_M)$. Leskovec et al. [7] have proposed two methods to solve this task, called *KronFit* [8] and *KronEM* [6]. The latter *Kron*ecker *E*xpectation-*M*aximization algorithm was originally proposed to address the following network completion problem. Given an observed part $G = (V, E)$ of a graph G_M, the missing part $Z = (V_Z, E_Z)$ is linked through the Kronecker model parameters Θ in terms of a probability law P that determines the subgraph Z by $P(Z|G,\Theta)$. Then it is the objective of a network completion algorithm to infer the most likely outcome of the missing part Z. The missing edges E_Z and node embeddings $V \subset V_M$ are regarded as latent variables. Applying an MCMC-Expectation-Maximization algorithm *KronEM*, it is possible to iteratively fit Θ by maximizing the expected likelihoods of $P(G, Z|\Theta)$ (M-step) and updating Z by $P(Z|G,\Theta)$ (E-step). Regarding details of this KronEM to construct the initial set of Kronecker graphs, the readers are referred to [6].

IV. Analysis of Realized P2P Streaming Topologies by Kronecker Graphs

In this section we evaluate whether the KronEM algorithm is an appropriate technique to capture the structure of the overlay graph arising from a P2P live streaming application. Here we use the well studied P2P streaming system SopCast as representative example. For this purpose we analyze several extensive data sets captured in PlanetLab [2], i.e., the SopCast application has been running on more than 350 PlanetLab nodes while their exchanged traffic was recorded.

A. Generic Graph Model of an Overlay Topology Arising from P2P Live Streaming

SopCast is a representative of a mesh-pull P2P architecture. This means that it builds an overlay structure among peers interesting in the same multimedia object on top of the transport network. It is based on a mesh topology while the data dissemination is realized by a pull mechanism. To enable fast data distribution, SopCast uses the so called *swarming* technique. Here each disseminated video stream is divided into smaller parts, called *chunks*, and then transmitted to (respectively received from) a multitude of peers. To model such a mesh-pull architecture of a P2P network, we use a hierarchical multi-layered approach (cf. [10]). It means that we model the overlay structure by means of an undirected neighborhood graph $G_N = (V_N, E_N)$ among an object-specific, time-dependent peer population $P^{O_j}(t) = \{p_1(t), ..., p_{n(t)}(t)\} \subset \mathcal{U}$ within a finite universe $\mathcal{U}$ and a given object space $\mathcal{O} = \{O_1, ..., O_m\}$. In the case of a streaming network, the latter is given by all available video channels of the network. The engaged peers form an overlay structure called *neighborhood community* which evolves dynamically in time, since peers may join and leave the community related to an object O_j at any time. The video data propagation is described by a directed *dissemination flow graph* $G_V = (V_V, E_V, f, c)$, $V_V \subseteq V_N$, among pairs p_i, p_j of peers derived from the dissemination topology (V_V, E_V). $e = (p_i, p_j) \in E_V$ represents a flow $\phi(e)$ of requested data chunks transferred from peer p_i to p_j on request of p_j. In other terms, this means that a single packet sent from one peer to another one during a captured P2P session generates a directed edge in the *dissemination flow graph* G_V (later this is simply referred to as monitored *overlay graph* G_M). Additionally, G_V contains a capacity function $c: V_V \times V_V \to \mathbb{R}_0^+$ and a flow function $f: V_V \times V_V \times T \to \mathbb{R}_0^+$ that determines the intensity of each flow $\phi(e)$ as flow rate $f(e,t) \geq 0$ and the flow type as attribute $t \in T = \{t_1, ..., t_h\}$. Finally, the underlying transport network is taken into account. Usually, it is a TCP/UDP-IP network connecting the peers p_i, p_j by flow paths $p = w(p_i, p_j) = (e_1, ..., e_n)$ along capacity constrained links (e_k, c_k). f, c are not used in our derivation.

B. Description of the Analyzed Representative SopCast Traces

Nowadays, SopCast is one of the most popular P2P video streaming networks. Its protocol specification is proprietary and thus, the application itself is the target of many other scientific studies (see e.g., [1] or [9] among many others). The measurement traces exploited in our analysis have been collected using the testbed PlanetLab. In the related experiments we have set up several hundred PlanetLab machines to act as regular SopCast peers and vantage points that have collected packet traces. These machines, that are usually hosted on servers of scientific institutions, are spread worldwide. During our experiments we have used the largest number of available machines, which has normally varied between 350 and 450 stable PlanetLab peers. At the beginning of the measurement during the setup and configuration phase all machines have synchronized their clocks by NTP. Joining times are normally distributed over a given initial period. During the data capturing phase all machines have joined the same closed SopCast channel and captured all the traffic generated by SopCast. Thereby, we can assure that we control all the peers involved in the video data dissemination and, even more important, we can later access the complete set of pcap trace files. We have considered all the UDP/TCP packets exchanged through well-known ports of the P2P application SopCast. After the data capturing phase, we have downloaded and aggregated all pcap traces to construct the traffic matrices at a central server. Thus, we have been able to rebuild the exact overlayed dissemination topology of a SopCast session.

In summary, our modeling approach of the P2P overlay topology of a mesh-pull based architecture during some time slice yields a hierarchical aggregated flow model of superimposed flows in inbound and outbound direction to the peers. Using the captured data traces, we can describe the exchange of chunk sequences among the peers by a dynamically evolving traffic matrix according to the graph $G_M \equiv G_V$.

Then the next important issue is related to the time horizon of the analysis. As the traffic matrix evolves over time, it is necessary to determine the length of the period in which a measurement snapshot is selected for analysis. Figure 1 illustrates the outcome of different timing resolutions for the same measurement trace. For each graph a different timing resolution $t \in \{1, 2, 3, 5, 15, 60, 120, 300, 600\}$ seconds has been chosen to construct the resulting overlay graph. The graph layout has been determined by the Fruchtermann-Reingold algorithm which is a *force-directed* graph drawing algorithm. Using this layout, the main challenge becomes visible. As SopCast is a mesh-pull based P2P system using a Gossip-like communication protocol, the graph gets denser and denser if the length of the snapshot interval increases. To determine the appropriate interval length of a measurement snapshot, thus one has to find a compromise between computational complexity and a loss of the fine grained dynamics of the overlay graph. To elaborate on this item, Figures 2 and 3 illustrate the total number of edges of one trace over time in relation to different lengths of the snapshot interval. Then we can simply conclude, by visual inspection, that it may be adequate to use an interval size as large as 5 seconds. It yields an acceptable compromise between computational complexity and the dynamical details of the application at a minute scale.

Figure 4 shows the number of edges over time for 3 closed channel measurement campaigns and an interval length of 5 seconds. In the experiments 1 and 2 roughly 350 peers controlled by PlanetLab have participated in the video distribution by SopCast for 1 hour.[1]

C. Fitting the Model Parameters of Kronecker Graphs

To examine whether Kronecker graphs G_K constitute an appropriate model of a P2P overlay network G_M and its parameters Θ can be computed reliably by the KronEM algorithm,

[1] The experiment *Trace 3* obviously comprised less peers. However, the most important point is that the number of edges stays constant after the start phase and then oscillates around a given mean.

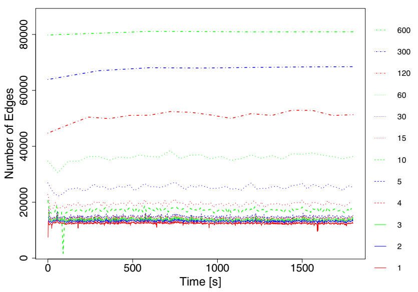

Figure 2. Number of edges over time for one trace with different lengths of the snapshot interval (the interval length is always given in seconds)

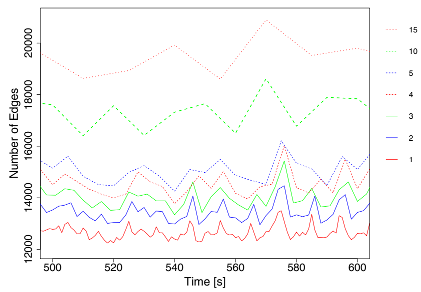

Figure 3. Number of edges over time for one trace with different lengths of the snapshot interval (same trace as shown in Fig. 2, but only for small interval lengths)

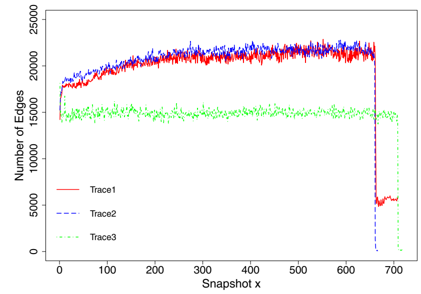

Figure 4. Number of edges over time for 3 closed channel measurement campaigns with approximately 300 peers controlled in Planet-Lab for all traces

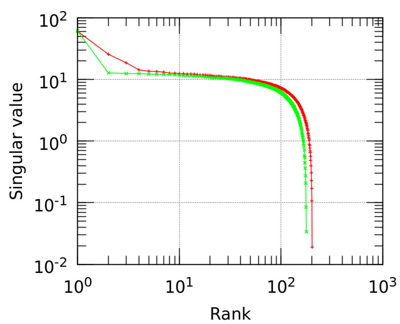

Figure 5. Scree plot: Real graph (red) vs. synthetic Kronecker graph (green)

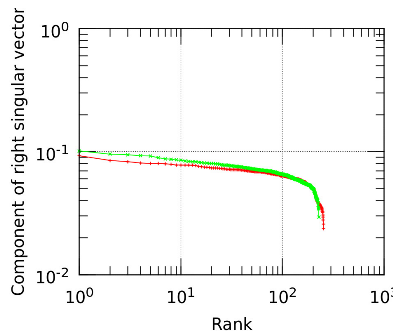

Figure 6. Singular vectors versus rank: Real graph (red) vs. synthetic Kronecker graph (green)

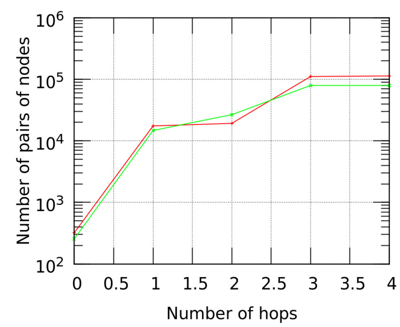

Figure 7. Hop-Plot: Real graph (red) vs. synthetic Kronecker graph (green)

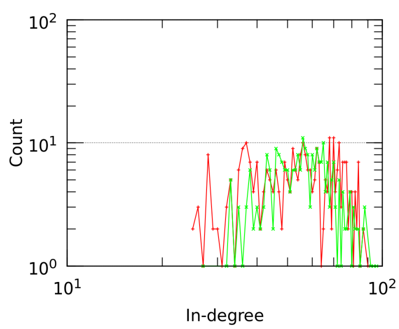

Figure 8. Distribution of in-degree: Real graph (red) vs. synthetic Kronecker graph (green)

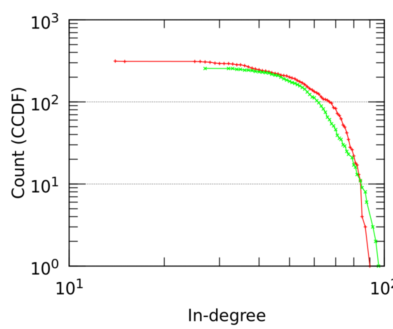

Figure 9. CDF of in-degree: Real graph (red) vs. synthetic Kronecker graph (green)

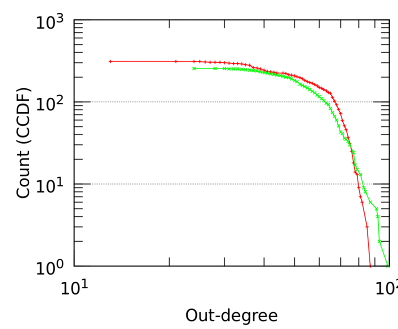

Figure 10. CDF of out-degree: Real graph (red) vs. synthetic Kronecker graph (green)

we present a case study conducted with a single, representative 5 seconds snapshot of a monitored closed channel SopCast session. The graph of this actual trace consists of 322 nodes. It means that 322 instances of SopCast have participated at dedicated PlanetLab nodes in the dissemination of the video content in a closed, i.e., private, channel. Therefore, we had full control of all peers constituting the overlay topology. Thus, we could reconstruct the dynamics of the full traffic matrix by means of the captured pcap traces. The graph consisted of 17509 edges in this selected 5 seconds snapshot. Here an edge $e = (i, j) \in E_M$ of the graph has been created by the transmission of at least one packet on e.

To fit the data to the Kronecker model and to determine the best fitting parameters of the initiator matrix Θ, we have investigated several choices regarding the best size of Θ. In agreement with [7] we found that even a 2×2 probability matrix Θ is appropriate to fit the data precisely. Regarding bigger matrices one may risk to overfit the model. In the considered case the estimate of the initiator matrix Θ is given by $\hat{\Theta} = \begin{pmatrix} 0.8646 & 0.9341 \\ 0.9306 & 0.5904 \end{pmatrix}$. We have already conducted extensive experiments with all the snapshots of the captured traces and found that the resulting initiator matrices do not diverge greatly for a specifically chosen interval length.

To construct a synthetic graph of the Kronecker model from the investigated empirical graph G_M, one simply has to take the 8th power of $\hat{\Theta}$. This yields an adjacency matrix consisting of $m_1 = 256$ nodes and 14749 edges. As the Kronecker model is stochastic, the outcome of the model fitting and the generation of new graphs is pseudo-random. However, the presented example is representative for the achievable results. Furthermore, one has to observe that due to this 2×2 probability matrix $\hat{\Theta}$, all sizes of the resulting adjacency matrices are a power of 2. Thus, it is not possible to capture exactly the number of nodes of the given real overlay graph without further adjustments. Further, as any other model, our Kronecker model intends to represent the overall properties of a given structure and not to mimic it in a special conduction.

D. Comparing the Stochastic Kronecker Model with the Overlay Graph of a SopCast Session

Here, we want to compare graph-theoretical metrics of a representative measurement snapshot with its fitted synthetic counterparts based on the stochastic Kronecker model. We start our comparison with an examination of the spectral structure of the resulting adjacency matrices. Figure 5 shows a so called scree plot. The latter plots the singular values (or eigenvalues) of the graph adjacency matrix versus their rank. The graph of the measurement snapshot is displayed in red and the synthetic graph based on Kronecker multiplication in green (analogously to all subsequent plots). It is obvious that the Kronecker graph captures the structure of the real graph very well. The minor inaccuracy can be explained by the already mentioned diverging sizes of the adjacency matrices.

Figure 6 illustrates the distribution of singular vector components. They are indicators of the so called *network value* versus the rank. The model again fits the actually measured data reasonably well. Both the scree plot and the plot of the singular vectors exhibit indications for a stretched exponential distribution. By visual inspection, one may simply conclude that the synthetic trace captures the nature of this distribution well enough. It is postponed to future studies to investigate and test the actual statistical fits of both distributions. The next metric related to the *network value* is provided by Figure 7. The hop-plot shows the number of reachable pairs within h hops or less as a function of the number of hops h. We observe that the synthetic Kronecker graph again matches the properties of the real graph very well. Figures 8, 9 and 10 show the count distribution of the in-degree, and the CDF of both the in-degree and out-degree. By visual inspection we realize again that the fits are accurate while small inaccuracies are due to the diverging sizes of the underlying adjacency matrices.

In conclusion, we see that the overlay modeling approach based on Kronecker graphs is able to capture very well the monitored graph and can adequately mimic its graph-theoretical metrics (at least, by a visual inspection; cf. [7]). This allows us to construct realistic synthetic models of overlay networks arising from real P2P streaming sessions.

V. Conclusion

We have proposed the application of Kronecker graphs as theoretical model to capture the overlay topology generated by a P2P live streaming session. The results derived from our analysis so far are promising. They illustrate that this approach is able to reflect the resulting topology adequately. Furthermore, we are able to create synthetic graphs resembling some important features of the original overlay structure.

We know, however, that there are a number of questions that are not answered yet. We found that the Kronecker model is able to capture the overlay topology statically along the time line of measurement snapshots. We still investigate more effective ways to describe the dynamics and evolution of the overlay topology without the need of the time consuming steps of model fitting. Furthermore, we have only evaluated a single P2P streaming application by our investigations. It will certainly be beneficial to validate the appropriateness of the model regarding P2P file sharing, e.g., by BitTorrent, or other live streaming systems. Further, new analytic test procedures to evaluate the quality of our fitted models based on local graph properties are required.

Our future work will tackle all these challenges. But we are convinced that the proposed modeling approach derived from Kronecker graphs has already revealed its great potential regarding the analysis of realistic P2P overlay networks.

Acknowledgments

A. Borges Vieira acknowledges the funding of his research by CNPq, CAPES, FAPEMIG, EU-IP mPlane (n-318627). U.R. Krieger thanks C. Fischer for her support regarding the postprocessing of the SopCast traces.

References

[1] I. Bermudez, M. Mellia, and M. Meo. Investigating overlay topologies and dynamics of P2P-TV systems: The case of SopCast. *IEEE Journal on Selected Areas in Communications*, 29(9):1863–1871, 2011.

[2] B. Chun, et al.. Planetlab: an overlay testbed for broad-coverage services. Computer Communication Review, 33(3):3–12, July 2003.

[3] C. X. A. Dimitropoulos, et al.. Towards a topology generator modeling AS relationships. In IEEE Int. Conf. Network Protocols, 2005.

[4] M. Gjoka, et al.. 2.5K-Graphs: from Sampling to Generation. In *Proc. IEEE INFOCOM '13*, Torino, Italy, April 2013.

[5] M. Jovanović, F. Annexstein, and K. Berman. Modeling peer-to-peer network topologies through small-world models and power laws. In *IX Telecommunications Forum, TELFOR*, pp. 1–4, 2001.

[6] M. Kim and J. Leskovec. The network completion problem: Inferring missing nodes and edges in networks. In *SIAM International Conference on Data Mining (SDM)*, pp. 47–58, 2011.

[7] J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, and Z. Ghahramani. Kronecker graphs: An approach to modeling networks. *Journal of Machine Learning Research*, 11:985–1042, March 2010.

[8] J. Leskovec and C. Faloutsos. Scalable modeling of real graphs using Kronecker multiplication. In *Proc. 24th International Conference on Machine Learning*, ICML '07, pp. 497–504, 2007.

[9] Y. Lu, B. Fallica, F. A. Kuipers, R. E. Kooij, and P. V. Mieghem. Assessing the quality of experience of SopCast. *International Journal of Internet Protocol Technology*, 4(1):11–23, March 2009.

[10] N.M. Markovich, A. Biernacki, P. M. Eittenberger, and U.R. Krieger. Integrated Measurement and Analysis of Peer-to-Peer Traffic. In *Proc. WWIC'10*, LNCS 6074, pp. 302-314, Springer, Heidelberg, 2010.

[11] A. Medina, et al.. Brite: An approach to universal topology generation. In *Proceedings Ninth International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems 2001,* pp. 346–353., IEEE, 2001.

[12] W. Yang and N. Abu-Ghazaleh. Gps: A general peer-to-peer simulator and its use for modeling BitTorrent. In *13th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems 2005,* , pp. 425–432. IEEE, 2005.