

A Population Dynamics Model for Data Streaming over P2P Networks

Jialing Xu, Guang-Hua Yang, Victor O.K. Li
 Department of Electrical and Electronic Engineering
 The University of Hong Kong
 Pokfulam Road, Hong Kong
 Email: {jlxu, ghyang, vli}@eee.hku.hk

Abstract—Data streaming (DS) over Peer-to-Peer (P2P) networks has been intensively studied in recent years and there have been various schemes proposed already. To evaluate these schemes, either measurement in experimental implementations, or simulation and theoretical analysis have been used. The former is inadequate as data are collected from different experiments, while the latter lacks a proper theoretical dynamics model. Our research aims at providing a general theoretical model to evaluate DS over P2P systems and analyze their dynamic behaviors. In this paper, with the analysis and abstraction of the characteristics of peers and their organization in DS over P2P, we propose a general population dynamics model for DS over P2P with fixed population. The model depicts the dynamic distribution of peers as a closed Markov queuing network. In particular, the model is scheme-independent and can be used with various schemes. Through theoretical analysis, we prove the model has equilibrium and only one closed-form solution. Besides, we verify the model through simulations, and show that it is a helpful analytical tool with a case study.

Keywords—peer-to-peer; data streaming; Markov queuing network; system dynamics;

I. INTRODUCTION

Due to its cost-effectiveness, the peer-to-peer (P2P) approach has been widely applied in data distribution in the Internet. Several traditional services, such as directory service, searching, and file sharing, have migrated onto this new platform. Among these migrants, data streaming (DS) has drawn special attention for its excellent performance in on-line multimedia delivery and for its special challenges in analysis and implementation.

Unlike ordinary data, streaming data has no clear start and stop points, and therefore is suitable for real-time applications, such as video and audio broadcasting, stock market information and weather information delivery. Traditionally, DS is based on the Client/Server (C/S) paradigm or built on multicast infrastructures. Such traditional DS approaches have been thoroughly studied and it is found that while reliability may be guaranteed, it is at the expense of low scalability and high initial deploying cost.

In contrast, a P2P network enjoys high scalability and no extra hardware cost is incurred for deployment. However, due to its changing and unstable environment, a DS system over P2P is also dynamic and unreliable. Providing efficient and reliable streaming services becomes a new challenge and excites the interests of researchers and engineers. In

the past few years, several DS schemes over P2P network have been proposed, including the early tree-based structured ones, such as SplitStream [1] and Chunkyspread [2], and the more recent swarming-based unstructured ones, like Coolstreaming [3] and RDS [4].

With the increasing number of proposed schemes, how to make effective and fair evaluation among them becomes a key issue. To achieve this objective, however, is not easy. At present, to evaluate such schemes, either measurement on experimental implementations, or simulation and theoretical analysis have been used. However, the data collected may not be useful to make fair comparisons among the various schemes as the data are collected from different experiments of different schemes, or from different implementations of the same scheme. Some efforts, such as [5], [6], [7], provide integral reviews of designs with regards to their design principles, strengths, weaknesses, scalabilities, etc. However, they are grounded on indirect comparison and intuitive analysis of the original data from the respective tests by their designers.

In addition, due to the lack of proper theoretical models, the theoretical analysis in previous studies is limited to simple static probabilistic characteristics, such as the average delay, average hops and average number of peers. These metrics, however, miss the dynamic characteristics of the P2P environment and may be misleading [8].

This work aims at providing a general theoretical model to evaluate DS over P2P systems and analyze their dynamic behavior. In this paper, we propose a general population dynamics model for DS over P2P with fixed population. The distribution of peers is modeled as a closed Markov queuing network. The model is scheme-independent and widely applicable to various schemes. We prove the existence of equilibrium in the model and that it has only one closed-form solution.

As far as we know, our work is the first theoretical study on the model of the population dynamics for DS over P2P networks. There are several existing studies on modeling data services over P2P. Our work is distinguished by its special consideration of peers' characteristics and organization in DS over P2P, and the proposed population dynamics model is a general model and applicable to various schemes. Meanwhile, the existing studies focus on the file-

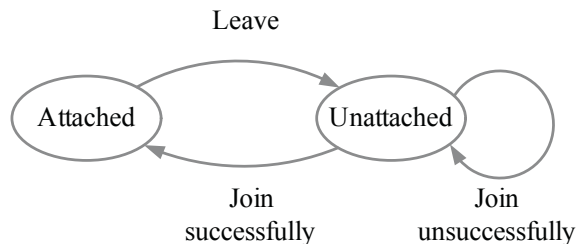


Figure 1. State transition diagram of a peer's behavior.

sharing application, and their proposed models are based on particular schemes, such as the BitTorrent [9] scheme or its variations [10], [8].

The rest of the paper is organized as follows. Sec. II is the analysis and abstraction of peers and their organization in the context of DS over P2P. Sec. III proposes the population dynamics model and the related theoretical analysis. Sec. IV verifies the model through simulation. Sec. V presents a case study based on the proposed model. Sec. VI concludes the paper.

II. ANALYSIS OF DATA STREAMING OVER P2P

In this section, we analyze the characteristics of peers in the context of DS over P2P and propose a series of abstractions of peers' roles, behaviors and their organization. The study provides necessary premise for the population dynamics model in Sec. III.

A. Roles of a peer

In DS over P2P, a peer's roles are jointly determined by the underlying P2P mechanism and the characteristics of the streaming application. As an elementary participant in the P2P environment, each peer takes two roles, "client" and "server", simultaneously. For data-centric applications over P2P networks, such as file sharing and DS, "caching" data from others and "forwarding" data to others are the tasks of "client" and "server" roles, respectively.

The characteristics of a peer are related to the properties of its roles. In general, the service consumption for peers as the "client" and the capability of service provision as the "server" are the two basic properties which interest us.

B. Behaviors of a peer

In general, DS schemes have their own specifications on peer behavior, usually including dozens of states and actions. To focus on the essentials, we propose a simple peer behavior model with only two possible actions, *Join* and *Leave*, and two possible states, *Attached* and *Unattached*.

Fig. 1 shows the possible transitions between the states. In particular, the only action a peer in the *Attached* state can take is *Leave*, entering the *Unattached* state. Similarly, the only action a peer in the *Unattached* state can take is *Join*. *Join* leads to two possible results. If the *Join* action succeeds,

then the peer enters the *Attached* state; otherwise, it stays in the *Unattached* state and retries after a scheme-determined delay.

To simplify the discussion, we assume the time spent on state transition is negligible. Once a peer joins a stream successfully, it gets served immediately. We also assume that an upstream peer's *Leave* does not cause its downstream peer's recursive *Leave*.

C. Delay of a peer

In all kinds of data service over P2P, data delivery is always accompanied by delay and delay jitter. In general, delay is defined as the interval between the time when the data chunk is generated at the source and the time when it is ready at the peer for playback and forwarding. Delay jitter is the variation of delay. In most schemes, delay jitter can be mitigated by increasing the local cache size. Take PPlive, a representative commercial P2P DS system for Internet-wide video service, as an example, the probability of smooth playback for peers with 150MB disk cache is about 90% [11]. In this paper, therefore, we focus on the delay of peers in streaming.

In general, DS applications are much more time sensitive than file sharing. Besides requiring identical rate in data playback and production, the streaming data has short effective time, after which the data becomes obsolete and useless. Live TV program broadcasting is an example of this type of applications. Given the properties of a peer's roles, the limited effective time puts a constraint on system capacity. The closer the population gets to the capacity, the higher probability the newly attached peer receives obsolete data, which leads to a *Leave* action. On the other hand, within the capacity, the delay experienced by newly attached peer increases with the increase of hop distance between the peer and the data source.

In general, the delay experienced by a peer is determined by three factors: delay of upstream peers, one-way transmission latency from the upstream peers to the peer, and the scheme-related delay. The scheme-related delay is determined by the size of the peer's local cache and caching mechanism and usually in the range from half a minute to several minutes. Also, most streaming applications expect continuous data playback at the same rate as the data is produced, so the scheme-related delay is relatively steady after a peer is attached to a stream. The transmission latency is quite small, varying from several milliseconds to less than 10 seconds. Compared with the scheme-related delay, the transmission latency is negligible. With this simplification and considering the data spreading process, delay is dominated by the scheme-related delay. Without loss of generality, we further assume discrete delay in our model.

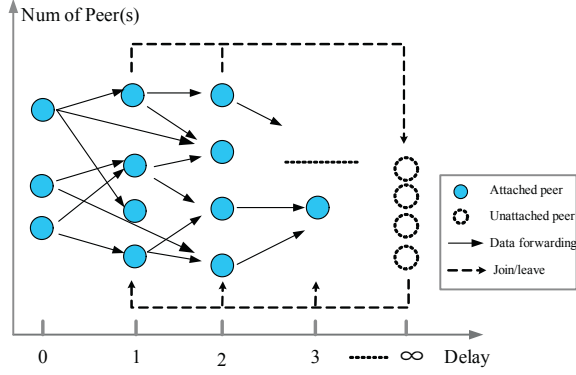


Figure 2. Dynamic organization of peers.

D. Definition of a peer

With the above analysis and discussion, a peer can be abstracted as a 4-tuple

$$[r_c, r_s, state, d],$$

where r_c and r_s denote the peer's service consumption and capability of service provision, $state$ denotes the peer's current state, and d denotes the delay experienced by the peer. To facilitate the comparison among different implementations, r_c and r_s are numbers without units. Obviously, d , r_c , and r_s are non-negative. We set $d = 0$ for the data source, and $d = \infty$ for the peers in the *Unattached* state.

E. Peer organization

In DS over P2P, the oriented diffusion model mentioned in [12] provides a vivid description of the data spreading process. It is similar to particle diffusion in physics: the newly generated streaming data is injected into the system at the data source, gradually diffuses among neighboring peers and finally saturates all peers attached to the stream.

Following this model and with the assumption of discrete delay, we propose to group peers in DS over P2P by the delay they experience. As illustrated in Fig. 2, all the peers are grouped into countable sets based on their delays. In this way, the population dynamics can be represented by the distribution of peers in these countable sets. Our population dynamics model in Section III provides a tool to track the evolution of such sets.

III. POPULATION DYNAMICS MODEL

In this section, we propose a general population dynamics model for DS over P2P with a fixed population. Through theoretical analysis, we prove that the model has equilibrium and one and only one closed-form solution.

A. Population dynamics model

We make the following assumptions:

- The peer population is fixed. No peer leaves the system and no peer joins the system from the outside.

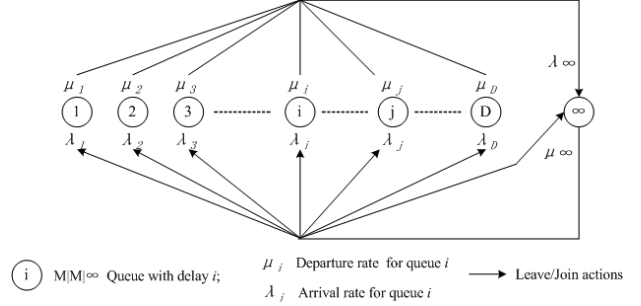


Figure 3. Markov queuing network model of closed population.

- The role and number of data sources are fixed.
- The time a peer stays in the *Attached* state or the *Unattached* state follows exponential distributions.
- All the peers are homogeneous. They have the same capability and the same behavior.

Inspired by the work of Kingman [13] and Chandy [14], our model maps the dynamic actions, *Join* and *Leave*, of peers to the customers' movements in a Markov queuing network with the structure illustrated in Fig. 3.

In Fig. 3, each vertex (a small circle labeled with an index) in the network represents a service queue. Peers are grouped into different queues according to their delay d . d is also used as the index of that queue and D is the maximum allowed delay. We use q_d , $d \in \{1, \dots, D\}$ to represent the queue with delay d ; and use x_d to represent the number of peers in q_d . Besides, q_0 denotes the queue of data sources, and q_∞ denotes the queue of unattached peers. There are $D+2$ queues in the closed system. Since the stream sources are unchanged, q_0 and x_0 are fixed, and thus they are omitted in the network. Due to the assumption on immediate service, all the queues in this model are $M|M|\infty$ queues.

In Fig. 3, an edge represents a feasible transition path among queues, through which a state transition of a peer can be made. According to the model, a path only exists between q_d , $d \in \{1, \dots, D\}$ and q_∞ .

Further, we define

$$s = [x_1, \dots, x_D, x_\infty]^T$$

as the state of the population in terms of x_d and represents the distribution of peers with different delays. In addition, we let S denote the space of all the possible states and hence S is irreducible. For convenience, in the rest of the paper, we use s_d to represent x_d of a given s .

With the homogeneity assumption, all peers have the same joining rate and leaving rate, represented by λ and μ , respectively. In addition, λ_d and μ_d are used to represent the arrival rate and service rate of q_d , $d \in \{1, \dots, D, \infty\}$ respectively. Obviously, μ_d satisfies

$$\mu_d = x_d \cdot \mu, \quad d \in \{1, \dots, D, \infty\}.$$

We find that λ_d is determined by μ_∞ and the transition probability of a peer moving from q_∞ to q_d , $d \in \{1, \dots, D\}$. This probability is determined by a specific scheme implementation and is also a function of the current state s .

To hide the difference among different schemes, we abstract their behavior into the transition probability, denoted by $P_{scheme}(s, d)$ in our model. The introduction of $P_{scheme}(s, d)$ makes the model scheme-independent. Then we have

$$\lambda_d = \begin{cases} P_{scheme}(s, d) \cdot x_\infty & , d \in \{1, \dots, D\} \\ \sum_{i=1}^D \mu_i = \mu \sum_{i=1}^D x_i & , d = \infty \end{cases}.$$

For simple schemes, such as the one used in Sec. IV, closed form expression of $P_{scheme}(s, d)$ is available; otherwise, $P_{scheme}(s, d)$ can be calculated through simulations or measurements in a real implementation.

In addition, we define V as the peer population including the source nodes, and V^* as the population excluding the source nodes. With the assumption of fixed population, V^* is fixed, thus the state space S satisfies

$$S = \{s \mid x_1 + x_2 + \dots + x_D + x_\infty = V^*\},$$

and obviously $\|S\|$ is finite.

B. Equilibrium and solution

Although the proposed Markov population dynamics model is similar to the general queuing network model proposed by Jackson [15], that model is based on the assumption of state-independent transition probability and hence can not be applied to our population dynamics analysis.

We define $p(s)$ as the stationary probability of s , ($s \in S$); $q(m, n)$, ($m, n \in S$) as the transition rate from state m to state n ; and $q(m)$ as the summation of the transition rates from state m to all the other states. $q(m)$ can be calculated as

$$q(m) = \sum_{m, n \in S} q(m, n), \quad (m \neq n). \quad (1)$$

For convenience, we let e_i denote a unit vector with the same form as s . Specially, its i th-coordinate equals 1 and all other coordinates equal zero. $q(m, n)$ can be calculated as

$$q(m, n) = \begin{cases} m_d \cdot \mu & (n=m-e_d+e_\infty) \cap (m_d > 0) \\ P_{scheme}(m, d) \cdot m_\infty & (n=m+e_d-e_\infty) \cap (m_\infty > 0), \\ 0 & otherwise \end{cases} \quad (2)$$

in which m_d is the number of peers in the d th queue of state m . As stated in Sec. II-C, delay d does not change while a peer is *Attached*. Changing only happens when a peer moves between q_d , $d \in \{1, \dots, D\}$ and q_∞ . In Eq. 2, the first case gives the state transition rate triggered by a peer moving

from q_d , $d \in \{1, \dots, D\}$ to q_∞ ; the second case shows the state transition rate triggered by a peer moving from q_∞ to q_d , $d \in \{1, \dots, D\}$; and the last case indicates that no other state transition is allowed.

Below, we prove that the proposed population dynamics model has equilibrium and closed-form solution.

Lemma 1: The proposed population dynamics model has equilibrium, and has one and only one solution.

Proof: Theorem 8 in [16] has proved that if a state space is irreducible, then it has equilibrium, and the stationary probability distribution of states has one and only one solution.

As stated above, $\|S\|$ is finite and S is irreducible. So the proposed population dynamics model has equilibrium and has one and only one solution. According to [16], the only solution to the stationary probability distribution of states is a set of positive numbers $p(n)$, ($n \in S$), which satisfies

$$\begin{aligned} \sum_{n \in S} p(n) &= 1, \\ q(n)p(n) &= \sum_{m \in S} q(m, n)p(m), \quad (n \in S). \end{aligned} \quad (3)$$

Unfortunately, Eq. 3 has no general closed-form solution. Considering the model as a special case of the above Markov queuing network, the numerical solution can be obtained by solving the equation set of

$$\begin{aligned} \left(\sum_{d=1}^D \lambda_d + \lambda_\infty + \sum_{d=1}^D \mu \cdot n_d + \mu \cdot n_\infty \right) \cdot p(n) = \\ \sum_{d=1}^D q(n + e_d - e_\infty, n) \cdot p(n + e_d - e_\infty) \\ + \sum_{d=1}^D q(n - e_d + e_\infty, n) \cdot p(n - e_d + e_\infty), \quad (n \in S). \end{aligned}$$

Lemma 2: The proposed population dynamics model has closed-form solution.

Proof: According to Kingman [13], reversibility leads to a closed-form solution. Below we prove that the proposed Markov queuing network is reversible.

Kolmogorov cycle condition [17] states that a system is reversible if and only if

$$q(n_1, n_2)q(n_2, n_3) \cdots q(n_r, n_1) = q(n_1, n_r)q(n_r, n_{r-1}) \cdots q(n_2, n_1)$$

is satisfied for any possible closed path of $\langle n_1, n_2, n_3, \dots, n_r, n_1 \rangle$, where $n_1, n_2, n_3, \dots, n_r$ are any distinct states in S . As a special case, for paths with length 2, the Kolmogorov cycle condition is always satisfied [17].

In our model, all the possible closed paths are composed of elementary paths between q_d , $d \in \{1, \dots, D\}$ and q_∞ . In other words, all the elementary closed paths have the form of $\langle m, n, m \rangle$, ($n, m \in S$) and with length 2. Therefore our model satisfies Kolmogorov cycle condition and is thus

reversible. According to [13], the closed-form solution to the stationary distribution of states is

$$p(n) = p(\infty) \prod_{d=1}^D H_d(n_d, n^d). \quad (4)$$

Due to space limitations, we only give the main formulas here. Detailed descriptions can be found in [13]. ■

IV. MODEL VERIFICATION

In this section, we verify the proposed population dynamics model through the comparison of its output to the simulation results in a case study of DS over P2P. Since the model is scheme-independent, we believe that its validity can be extended to other schemes.

A. Scheme specification

With the assumption on homogeneity, all peers are assumed to have the same r_c and r_s . We use R_c and R_s instead to denote the service consumption r_c and the capability of service provision r_s of all peers. In our verification, we let $R_c = 1$; thus the scheme has a tree-like structure.

In addition, we define the behavior of peers as follows: when an *Unattached* peer U tries to join a stream, it randomly selects a peer already *Attached* to the stream, say peer A , as its upstream provider. If A has enough available resource and $d_A < D$ is satisfied, then U joins the stream successfully and $d_U = d_A + 1$. Otherwise, U stays in the *Unattached* state and waits for a random period before another trial. When in *Attached* state, a peer leaves the stream with rate μ . When in the *Unattached* state, a peer attempts to join the stream with rate λ . $\mu = 4$ and $\lambda = 1$ are used in the verification.

B. Analysis

With the above specifications on peers' behavior, the corresponding $P_{scheme}(s, d)$ satisfies

$$P_{scheme}(s, d) = \begin{cases} \frac{s_{d-1}}{V-s_\infty} & s_{d-1} > 0 \cap s_d < r_s \\ & \cap s_\infty > 0 \text{ for } d \in [1, D] \\ \frac{s_{d-1}}{V-s_\infty} \left(1 - \frac{C_{R_s \cdot s_{d-1}}^{R_s}}{C_{R_s \cdot s_{d-1}}^{R_s}} \right) & s_{d-1} > 0 \cap R_s \leq s_d \leq R_s \cdot s_{d-1} \\ & \cap s_\infty > 0 \text{ for } d \in [1, D] \\ 1 - \sum_{j=1}^D P_{scheme}(s, j) & d = \infty \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

In the above equation, the first two cases represent the probabilities that a peer changes from the *Unattached* state to the *Attached* state successfully, i.e., moving from q_∞ to q_d , $d \in [1, \dots, D]$; the third case represents the probability that a *Join* attempt fails, and the peer stays in the *Unattached* state; and the last case represents that no other state transition is allowed. With a closer view, in the

Table I
MAJOR STATISTICS OF THE SIMULATION

Statistics	Value
All possible states $C_{V^*+D}^D$	$C_{10+4}^{10-1} = 1001$
Number of different states observed	420
Number of state transitions	39092830
Simulation time	2479299.48 sec
Average occurrences per state	91125.48
Max occurrences of observed state	6557038
Average duration per observed state	5779.25
Max duration of observed state	504054
Min duration of observed state	1.65×10^{-6}

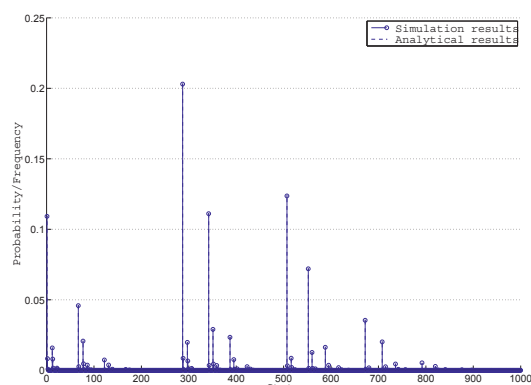


Figure 4. Comparison of simulation results and analytical results.

first two cases, $P_{scheme}(s, d)$ is the joint probability of two independent steps. First, pick a peer as the possible data provider with delay $i - 1$ at random. Clearly, the probability for this step is $\frac{s_{d-1}}{V}$. Second, check if the selected candidate has available upload resource. If $s_d < r_s$ and $s_{d-1} > 0$, the check always leads to successful attachment and the joint probability is $\frac{s_{d-1}}{V-s_\infty}$, given by the first case of Eq. 5. If currently there is insufficient resource left, the probability of successful attachment should be $\left(1 - \frac{C_{R_s \cdot s_{d-1}}^{R_s}}{C_{R_s \cdot s_{d-1}}^{R_s}} \right)^1$, and the joint probability is $\frac{s_{d-1}}{V-s_\infty} \left(1 - \frac{C_{R_s \cdot s_{d-1}}^{R_s}}{C_{R_s \cdot s_{d-1}}^{R_s}} \right)$, given by the second case of Eq. 5.

C. Simulation results

We have implemented and simulated the scheme on a P2P network based on our RDS protocol [4]. In the simulation, we choose $x_0 = 2$, $V^* = 10$, $D = 4$ and $R_s = 3$, and record all the state transitions. In order to get accurate statistical results, the simulation was run for long enough to get a

¹This is the same as the probability of failing to randomly pick up R_s black balls from $R_s \cdot s_{d-1}$ balls in which s_d are black and the rest are white.

Table II
CONFIGURATIONS OF λ AND μ .

Conf.	Arrival rate λ	Departure rate μ	Ratio $\frac{\lambda}{\mu}$
1	5	4	1.25
2	8	4	2
3	9	3	3
4	8	2	4

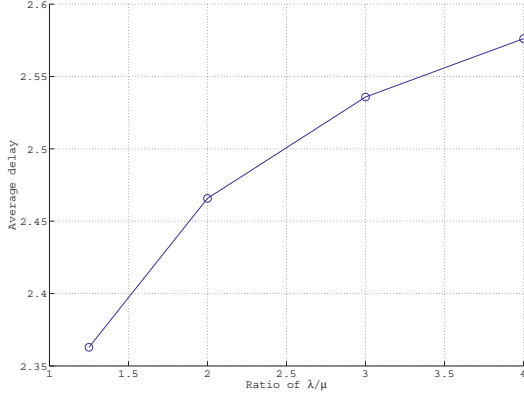


Figure 5. Average delay vs. $\frac{\lambda}{\mu}$.

large number of state transitions. The major statistics are listed in Table I.

Fig. 4 shows the frequencies of states observed in the simulation and the analytical results based on the proposed population dynamics model. From the figure, we can find that the proposed model matches the simulation results very well. In fact, the results overlap each other as shown in the figure. The mean square of differences between the frequencies, $P_{sim}(s)$, and their corresponding probabilities, $P_{num}(s)$, is

$$\frac{\sum_{s \in S} (P_{sim}(s) - P_{num}(s))^2}{||S||} = 3.2646 \times 10^{-10}.$$

Since the proposed model is scheme-independent, we are convinced that the validity of our model can be extended to other possible schemes.

V. APPLICATIONS OF THE MODEL

In this section, we apply the model in a case study of the population dynamics. As implied in [13], the ratio $\frac{\lambda}{\mu}$ is a factor of the stationary probability distribution of population states. The case study investigates the influence of the ratio with the scheme defined in Sec. IV. In practice, we evaluate the influence with four different configurations of the ratios listed in Table II. In addition, we let $V^* = 15$, $D = 4$ and $R_s = 3$ in all the configurations.

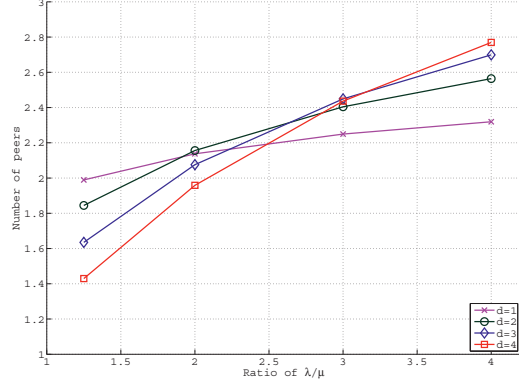


Figure 6. Average peer number vs. $\frac{\lambda}{\mu}$.

The proposed model provides the stationary distribution of the probabilities of all the states. Based on this, we calculate the average number of peers with different delays and the average delay for all the attached peers. Table III lists all the calculated values for the four different $\frac{\lambda}{\mu}$ ratios. The average delay for all the attached peers are shown in Fig. 5. The results indicate that the average delay increases non-linearly with $\frac{\lambda}{\mu}$.

In addition, Fig. 6 shows the change of calculated average number of peers as a function of different $\frac{\lambda}{\mu}$ ratios for various delays. The results give detailed insights on the changing of peer numbers over delays with increasing $\frac{\lambda}{\mu}$. The average peer numbers for all delays increase with $\frac{\lambda}{\mu}$. The phenomena suggest that the ratio $\frac{\lambda}{\mu}$ reflects peers' intention to join and stay attached in a stream.

Second, with the increasing of $\frac{\lambda}{\mu}$, the average peer numbers with large delays increase faster than that of small delays. Specifically, when $\frac{\lambda}{\mu} \leq 2$, the peers with $d = 1$ has the largest population, and the peer population decreases with the delay increasing. With the increasing of $\frac{\lambda}{\mu}$, the differences among them for different delays diminish. When $\frac{\lambda}{\mu} \geq 3$, the populations of peers with large delays become more than that of small delays. In addition, these observations suggest the possible existence of some $\frac{\lambda}{\mu}$ values which leads to even population distributions.

The case study demonstrates that the population dynamics model is a helpful analytical tool in the study of the dynamics in DS over P2P.

VI. CONCLUSION

In this paper, we analyze the characteristics of DS over P2P and propose a general population dynamics model for DS over P2P with fixed population. The model depicts the dynamic distribution of peers over their experienced delays as a closed Markov queuing network. Different streaming schemes are treated as different probability distribution

Table III
AVERAGE PEER NUMBER AND AVERAGE DELAY VS. $\frac{\lambda}{\mu}$.

$\frac{\lambda}{\mu}$	Average number of peers with delay d					Average delay
	$d = 1$	$d = 2$	$d = 3$	$d = 4$	$d = \infty$	
1.25	1.9893345	1.844334	1.6349594	1.4294358	8.1019363	2.3628477
2	2.1375977	2.1562003	2.0756941	1.9585349	6.6719729	2.4657954
3	2.2493547	2.404296	2.4491919	2.4364818	5.4606756	2.5358189
4	2.3194356	2.5642009	2.6990861	2.7695584	4.647719	2.5761424

functions (PDF) of peer's movements in the network. This makes the model scheme-independent and can be applied to various schemes for analysis. Through theoretical analysis, we prove that the model guarantees the existence of equilibrium and one and only one closed-form solution. With our simulation, the validity of the model is verified. The case study demonstrates that the model can be a helpful analytical tool in the study on the dynamics of DS over P2P.

REFERENCES

- [1] M. Castro, P. Druschel, A.-M. Kermarrec, A. Nandi, A. Rowstron, and A. Singh, "SplitStream: high-bandwidth multicast in cooperative environments," *SIGOPS Oper. Syst. Rev.*, vol. 37, no. 5, pp. 298–313, 2003.
- [2] V. Venkataraman, K. Yoshida, and P. Francis, "Chunkspread: Heterogeneous unstructured tree-based peer-to-peer multicast," in *ICNP '06. Proceedings of the 2006 14th IEEE International Conference on Network Protocols, 2006.*, Nov. 2006, pp. 2–11.
- [3] X. Zhang, J. Liu, B. Li, and Y.-S. Yum, "Coolstreaming/DONet: a data-driven overlay network for peer-to-peer live media streaming," in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, vol. 3, March 2005, pp. 2102–2111.
- [4] J. Xu and V. O. Li, "Request-driven swarming scheme for p2p data streaming," *Computer Communications*, vol. 32, no. 12, pp. 1410 – 1417, 2009.
- [5] B. Li and H. Yin, "Peer-to-peer live video streaming on the internet: issues, existing approaches, and challenges [peer-to-peer multimedia streaming]," *Communications Magazine, IEEE*, vol. 45, no. 6, pp. 94–99, June 2007.
- [6] S. Venot and L. Yan, "Peer-to-peer media streaming application survey," pp. 139–148, Nov. 2007.
- [7] Y. Liu, Y. Guo, and C. Liang, "A survey on peer-to-peer video streaming systems," *Peer-to-Peer Networking and Applications*, vol. 1, no. 1, pp. 18–28, Mar. 2008. [Online]. Available: <http://dx.doi.org/10.1007/s12083-007-0006-y>
- [8] J. V. Riikka Susitaival, Samuli Aalto, "Analyzing the dynamics and resource usage of p2p file sharing by a spatio-temporal model," in *Computational Science - ICCS 2006*, vol. 3994/2006. Springer Berlin / Heidelberg, 2006, pp. 420–427.
- [9] B. Cohen, "Incentives build robustness in BitTorrent," 2003. [Online]. Available: <http://www.bittorrent.org/bittorrentecon.pdf>
- [10] D. Qiu and R. Srikant, "Modeling and performance analysis of bittorrent-like peer-to-peer networks," in *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*. New York, NY, USA: ACM, 2004, pp. 367–378.
- [11] Y. Chen, C. Chen, and C. Li, "Measure and model p2p streaming system by buffer bitmap," in *High Performance Computing and Communications, 2008. HPCC '08. 10th IEEE International Conference on*, Sept. 2008, pp. 242–249.
- [12] M. T. Akura and K. Altinkemer, "Diffusion models for b2b, b2c, and p2p exchanges and e-speak," *Journal of Organizational Computing and Electronic Commerce*, vol. 12, pp. 243 – 261, 2002.
- [13] J. F. C. Kingman, "Markov population processes," *Journal of Applied Probability*, vol. 6, no. 1, pp. 1–18, 1969.
- [14] K. M. Chandy, "Analysis and solutions for general queueing networks," in *Sixth Annual Princeton Conference on Information*, 1972.
- [15] J. R. Jackson, "Networks of waiting lines," *Operations Research*, vol. 5, no. 4, pp. 518–521, 1957.
- [16] D. G. Kendall and G. E. H. Reuter, "The calculation of the ergodic projection for markov chains and processes with a countable infinity of states," *Acta Mathematica*, vol. 97, no. 1-4, pp. 103–144, Mar. 1957.
- [17] D. G. Kendall, "Unitary dilations of one-parameter semi-groups of markov transition operators, and the corresponding integral representations for markov processes with a countable infinity of states," in *Proceedings London Mathematical Society*, vol. s3-9, no. 3, 1959, pp. 417–431.