Providing Statistically Guaranteed Streaming Quality for Peer-to-Peer Live Streaming

Miao Wang, Lisong Xu, and Byrav Ramamurthy
Department of Computer Science and Engineering
University of Nebraska-Lincoln
Lincoln, NE 68588-0115
mwang@cse.unl.edu, xu@cse.unl.edu, byrav@cse.unl.edu

ABSTRACT

Most of the literature on peer-to-peer (P2P) live streaming focuses on how to provide best-effort streaming quality by efficiently using the system bandwidth; however, there is no guarantee about the provided streaming quality. This paper considers how to provide statistically guaranteed streaming quality to a P2P live streaming system. We study a class of admission control algorithms which statistically guarantee that a P2P live streaming system has sufficient overall bandwidth. Our results show that there is a tradeoff between the user blocking rate and user-behavior insensitivity (i.e., whether the system performance is insensitive to the fine statistics of user behaviors). We also find that the system performance is more sensitive to the distribution change of user inter-arrival times than to that of user lifetimes.

Categories and Subject Descriptors

C.2 [**Computer-Communication Networks**]: Network Protocols

General Terms

Algorithms, Performance, Design, Theory

Keywords

Peer-to-Peer Networks, Admission Control, Video Streaming

1. INTRODUCTION

Even though peer-to-peer (P2P) live streaming has been extensively studied [14], most of the literature focuses on how to provide *best-effort streaming quality* by efficiently using system bandwidth. That is, a P2P streaming system makes its best effort to provide good streaming quality by constructing an efficient P2P overlay architecture and running an efficient block scheduling algorithm; however, there is no guarantee about the provided streaming quality.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
NOSSDAV'09, June 3–5, 2009, Williamsburg, Virginia, USA.
Copyright 2009 ACM 978-1-60558-433-1/09/06 ...$5.00.

This paper considers how to provide *guaranteed streaming quality* to a P2P live streaming system. Due to the dynamic nature of a P2P system, it is impossible to provide an absolute guarantee. Instead, we consider a statistical guarantee, which ensures that the streaming quality provided by a P2P system is statistically guaranteed. Statistically guaranteed streaming quality can greatly improve the satisfaction of streaming users, compared to the best-effort streaming quality provided by the current P2P live streaming systems. In some cases, statistically guaranteed streaming quality is highly desired. For example, for a P2P live streaming system with some free channels and some paid channels, it is highly desirable that a paid channel provides a high streaming quality guarantee probability, whereas a free channel provides only best-effort streaming quality or a low streaming quality guarantee probability.

There are two different ways to provide statistically guaranteed streaming quality. First, at the individual peer level where the specific streaming quality provided to each peer is statistically guaranteed. Second, at the overall channel level where the average streaming quality provided to the whole channel is statistically guaranteed. The peer-level quality guarantee can provide a more accurate guarantee for each peer; however, it heavily depends not only on the overall system bandwidth but also on the underlying overlay construction method and block scheduling algorithm. On the other hand, the channel-level quality guarantee mainly depends on the overall system bandwidth. In addition, even though the channel-level quality guarantee cannot ensure the accurate streaming quality provided to each individual peer, it ensures the average streaming quality provided to all peers of a channel. In this paper, we consider only the channel-level quality guarantee.

A fundamental problem in providing statistical channel-level quality guarantee is the statistical bandwidth guarantee problem, which is how to statistically guarantee that a channel has sufficient overall bandwidth for its streaming. We assume that the upload capacity of users is the only bottleneck for a P2P live streaming system. That is, the download capacity of a user is higher than the streaming rate, and bandwidth bottlenecks are located at the edges instead of the core of the Internet, which are reasonable assumptions [19] for the current Internet. With this assumption, *the statistical bandwidth guarantee problem becomes how to guarantee that the probability for a channel to have sufficient overall upload bandwidth is higher than a threshold.*

In order to achieve statistical bandwidth guarantee, we study a class of admission control algorithms, which admits or rejects a user based on the user information and the channel state. Another way to achieve statistical bandwidth guarantee is to drop users when a P2P system has insufficient overall bandwidth. However, dropping a user is usually considered more annoying to the user than rejecting a user, thus in this paper, we consider only admission control algorithms. We are particularly interested in the user-behavior insensitivity of an admission control algorithm, which is whether the algorithm performance is insensitive to the fine statistics of user behaviors including both the distribution of user inter-arrival times and the distribution of user lifetimes. This is because we believe that user-behavior insensitivity is the key to designing an admission control algorithm that is robust and has predictable bandwidth guarantee in a dynamic and heterogeneous P2P system. We have the following observations from our results.

- *There is a tradeoff between the user blocking rate and user-behavior insensitivity when maintaining the same bandwidth guarantee.* Intuitively, this is because in order to reduce the user blocking rate, an algorithm uses more channel state information, which however makes the algorithm more sensitive to the statistics of user behaviors.

- *The statistical bandwidth guarantee achieved by an algorithm is more sensitive to the distribution change of user inter-arrival times than to that of user lifetimes.* Our simulation results show that the bandwidth guarantee probability obtained with a Poisson user arrival process can be used as the upper bound of the probability for general user arrival processes. The bandwidth guarantee probability obtained with an exponential user lifetime distribution can be used as a good estimate of the probability for general user lifetime distributions.

The rest of the paper is organized as follow. Section 2 reviews the related work. Section 3 formulates the problem of statistical bandwidth guarantee. Section 4 proposes a class of admission control algorithms. Section 5 evaluates the performance of these admission control algorithms by simulation. Finally, Section 6 concludes the paper.

2. RELATED WORK

The proposed statistical streaming quality guarantee for a P2P live streaming system (referred to as P2P Quality of Service, or P2P QoS) is different from the traditional QoS for IP networking (referred to as IP QoS) [7, 3, 8]. IP QoS focuses on the bandwidth allocation in the Internet backbone, and it mainly considers the bandwidth mismatch between the edges and backbone of the Internet [8]. On the other hand, P2P QoS focuses on the upload-bandwidth allocation at the Internet edges, and it mainly considers the bandwidth asymmetry between the download and upload at the Internet edges. The current Internet has both asymmetric users (e.g. cable and Asymmetric Digital Subscriber Line (ADSL) users) and symmetric users (e.g Symmetric Digital Subscriber Line (SDSL) users), and we believe that the future Internet will remain as it is due to the heterogeneous nature of the Internet.

Admission control has been extensively studied in IP [10], ATM [15], wireless [1], and satellite [17] networks to provide bandwidth guarantee. However, existing admission control algorithms cannot be directly applied to P2P systems, since they are fundamentally different from previous types of networks, in that the total upload capacity of a P2P system is dynamic and dependent on the total number of users, which however is not the case for other types of networks.

Notation	Description
r	streaming rate
c^S	the upload capacity of the streaming server
λ^H	average arrival rate of super users
λ^L	average arrival rate of ordinary users
$1/\mu^H$	average life time of super users
$1/\mu^L$	average life time of ordinary users
c^H	upload bandwidth of a super user
c^L	upload bandwidth of an ordinary user
N^H	number of super users
N^L	number of ordinary users
C	total upload bandwidth of a system
R	total required bandwidth of a system
δ	required bandwidth guarantee probability

Table 1: Notation

Due to the dynamic nature of a P2P system, it is very challenging to provide statistical service guarantee for a P2P system. There is very little related work on this topic. Bindal *et al.* [2] examine the factors that determine the statistical service guarantee in P2P file sharing applications, such as BitTorrent. They conclude that "*self-organizing P2P file distributions indeed need external help in order to provide QoS guarantees, but such guarantees are achievable with proper enhancements to the P2P network.*" Raghuveer *et al.* [16] consider how to ensure that each peer gets sufficient bandwidth with a high probability if the system has sufficient overall bandwidth. Kung *et al.* [12] and Xu *et al.* [20] use admission control to determine whether a peer should accept the request of another user to be a neighbor. Different from previous works, our work considers how to use admission control to ensure that the system has sufficient overall upload capacity with a high probability.

3. PROBLEM FORMULATION

This section proposes a queueing model used to study the statistical bandwidth guarantee for a channel of a P2P live streaming system. The notation is summarized in Table 1.

Inspired by the stochastic fluid model by Kumar *et al.* [11], we model a channel of a P2P system by the queueing model shown in Figure 1, which captures two fundamental properties of P2P streaming, i.e., heterogeneous upload capabilities and peer churn. For example, CoolStreaming [13] conducted an experiment in 2006 with a streaming rate (denoted by r) of 768Kbps. Their results show that about 7% users can upload the stream at a rate higher than r, about 10% users can upload at a rate between r and $r/2$, about 20% users can upload at a rate between $r/2$ and $r/6$, and the remaining 63% users can upload at a rate even lower than $r/6$. This experiment clearly shows the different upload capabilities of different users.

Our model considers two classes of users (and can be extended to more classes). Class 1 contains a group of super users each capable of uploading at a high rate of c^H, and class 2 contains a group of ordinary users each capable of

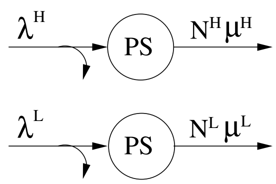

Figure 1: A state-dependent processor-sharing (PS) queueing model for a channel of a P2P live streaming system with two types of users: super users and ordinary users.

uploading at a low rate of c^L. We have $c^H > r > c^L$. A new user arrives at the system randomly with an average rate of λ^H and λ^L for a super user and an ordinary user, respectively. A new user may be admitted or rejected (also called blocked) by an admission control algorithm based on the upload bandwidth of the user and the current state of the system. If a user is admitted, it stays in the system for a random lifetime with average $1/\mu^H$ and $1/\mu^L$ for a super user and an ordinary user, respectively. Each class is modeled as a state-dependent processor-sharing (PS) queueing node [6] shown in Figure 1. The service rate of a queueing node depends on the current node state. For example, the service rate of the super user node (i.e., the top node in the figure) is $N^H\mu^H$, where N^H is the current number of super users.

We say that a system has sufficient upload bandwidth if $C \geq R$, where C and R denote the total upload bandwidth and the total required bandwidth, respectively. The total upload bandwidth C of the system is a function $f_C(\cdot)$ of the current system state as defined below

$$C = f_C\left(N^H, N^L\right) = N^H \times c^H + N^L \times c^L + c^S \qquad (1)$$

where c^S is the upload capacity of the streaming server. The total required bandwidth R of the system is a function $f_R(\cdot)$ of the current system state as defined below

$$R = f_R\left(N^H, N^L\right) = \left(N^H + N^L\right) \times r \times \varepsilon \qquad (2)$$

where $\varepsilon \geq 1.0$ indicates the control overhead and bandwidth inefficiency of the system, and depends on the underlying overlay architecture and block scheduling algorithm of the system. For example, packet-level simulation results [21] show that ε is about 1.15 for an overlay with a mesh-based overlay architecture and a random block scheduling algorithm.

Finally, *the statistical bandwidth guarantee problem is to determine whether a new user is admitted or rejected in order to guarantee $\mathbb{P}[C \geq R] \geq \delta$, where δ is the required bandwidth guarantee probability*.

We are interested in the following performance metrics when evaluating an admission control algorithm for achieving statistical bandwidth guarantee.

- *Implementation Difficulty*: How difficult is it to implement the algorithm?
- *Blocking Rate*: What is the average blocking rate for the algorithm to achieve a certain bandwidth guarantee probability δ?
- *Retry Robustness:* How robust is the algorithm in case that a rejected user repeatedly retries its admission request?
- *User-Behavior Insensitivity*: How insensitive is the algorithm to the fine statistics of user behaviors (i.e., user arrival process and user lifetime distribution)?

4. ADMISSION CONTROL ALGORITHMS

In this section, we propose three admission control algorithms for achieving statistical bandwidth guarantee.

Inspired by insensitive load balancing work by Bonald *et al.* [4], we study the following three admission control algorithms for a channel.

1. *Static User Admission Control (SUAC)* admits all super users into the channel, and randomly admits an ordinary user with probability β_{SUAC}, where $\beta_{SUAC} \in [0, 1]$.

2. *Semi-Static User Admission Control (SSUAC)* admits all super users, and admits an ordinary user if the following condition is true, where $\beta_{SSUAC} \in [0, 1]$.

$$\frac{f_R\left(\mathbb{E}[N^H], N^L + 1\right)}{f_C\left(\mathbb{E}[N^H], N^L + 1\right)} \leq \beta_{SSUAC} \qquad (3)$$

3. *Dynamic User Admission Control (DUAC)* admits all super users, and admits an ordinary user if the following condition is true, where $\beta_{DUAC} \in [0, 1]$.

$$\frac{f_R\left(N^H, N^L + 1\right)}{f_C\left(N^H, N^L + 1\right)} \leq \beta_{DUAC} \qquad (4)$$

Since the upload bandwidth c^H of a super user is greater than the streaming rate r, all three algorithms always admit a super user. But they make different admission decisions for an ordinary user. SUAC is "static" in the sense that its admission decision for an ordinary user does not depend on the current system state (i.e. N^H and N^L), whereas DUAC is "dynamic" in that its admission decision depends on the current system state. SSUAC is "semi-static" since its admission decision depends only on the current state of ordinary users (i.e. N^L) but not on the current state of super users (i.e. N^H).

Below, we compare these three admission control algorithms according to the performance metrics described in Section 3.

- *Implementation Difficulty*: SUAC is the easiest to implement, since it does not need to measure anything. DUAC is the hardest to implement, since it is not trivial to accurately and quickly measure the current N^H and N^L. SSUAC is in the middle, since the average value of N^H can be obtained by using the history information and then it only needs to accurately and quickly measure the current N^L.
- *Blocking Rate*: Intuitively, since DUAC makes a dynamic decision based on the current channel state, it should achieve the lowest blocking rate for a given δ, whereas SUAC makes a static decision, it should achieve the highest blocking rate. The performance of SSUAC should fall somewhere in between. This is verified in the next section by numerical results.

- *Retry Robustness:* Both SSUAC and DUAC are robust in case of user retries, since for a given system state, no matter how many times a rejected ordinary user retries its admission request, it will always be rejected by both SSUAC and DUAC. However, with SUAC, a rejected ordinary user can keep retrying its admission request until it is finally admitted. One possible solution for SUAC is to keep track of all recently rejected users (e.g. their IP addresses).
- *User-Behavior Insensitivity*: Intuitively, since DUAC is more dynamic than SUAC (i.e., more dependent on the channel state), DUAC is more sensitive to the fine statistics of user behaviors than SUAC. Specifically, we have the following insensitivity theorem.

We say that an admission control algorithm is *insensitive to the user lifetime distribution*, if the steady state distribution of a P2P live streaming system using this algorithm depends only on the average lifetime (i.e. $1/\mu^H$) of super users and that (i.e., $1/\mu^L$) of ordinary users, but does not depend on the lifetime distribution of super users and that of ordinary users. We have the following theorem.

THEOREM 1. *Under the assumption that a super user arrives as a Poisson process and an ordinary user arrives as a Poisson process, the sufficient and necessary condition for an admission control algorithm to be insensitive to the lifetime distribution is that its admission decisions do not depend on the current number of super users (i.e., N^H).*

Proof: Let $a^H(N^H, N^L)$ and $a^L(N^H, N^L)$ denote the arrival rate of admitted super users and that of admitted ordinary users, respectively, when there are N^H supers users and N^L ordinary users.

According to the insensitivity theory of processor-sharing queueing networks developed by Bonald and Proutere [5, 4], the queueing model shown in Figure 1 is insensitive to the user lifetime distribution if and only if

$$a^H(N^H, N^L)a^L(N^H+1, N^L) = a^H(N^H, N^L+1)a^L(N^H, N^L)$$

Since every super user is admitted, we have $a^H(N^H, N^L) = \lambda^H$ for any N^H and N^L, and thus the sufficient and necessary condition for insensitivity becomes

$$a^L(N^H + 1, N^L) = a^L(N^H, N^L)$$

That is, the admission decision is independent of N^H, but can be dependent on N^L. □

It is then easy to see that both SUAC and SSUAC are insensitive to the user lifetime distribution under the Poisson user arrival assumption, but DUAC is sensitive to the user lifetime distribution. This implies that the bandwidth guarantee probability achieved by SUAC and SSUAC depends on the user lifetime distribution through the mean only.

We say that an admission control algorithm is *insensitive to the user arrival process*, if the steady state distribution of a P2P live streaming system using this algorithm depends only on the average arrival rate (i.e. λ^H) of super users and that (i.e., λ^L) of ordinary users, but does not depend on the inter-arrival time distribution of super users and that of ordinary users.

However, we do not have a theorem for the insensitivity to the user arrival process, and as shown in the next section, all three algorithms are sensitive to the user arrival process.

5. NUMERICAL RESULTS

In this section, we compare these three admission control algorithms with numerical results obtained for the queueing model shown in Figure 1. We study a P2P live streaming system with the following parameters. Streaming rate r is 3, upload rate c^H of a super user is 7, and upload rate c^L of an ordinary user is 1. According to [11], these three values approximately reflect the actual settings of a typical P2P live streaming system (for example, when the rate unit is 100Kbps). The upload capacity c^S of the streaming server is 14. Limited by the memory space required to measure the state distribution $\mathbb{P}(N^H, N^L)$, we consider a small P2P system with an average of 25 super users and 50 ordinary users by setting $\lambda^H = 50$, $\lambda^L = 100$, and $\mu^H = \mu^L = 2$. If the time unit is 1 hour, this implies that a user stays in the system for an average of 0.5 hours. All results are obtained by simulating the system for 1,000,000 time units.

5.1 Blocking Rate of Ordinary Users

In this group of simulations, we study the blocking rate of each admission control algorithm in order to achieve a required bandwidth guarantee probability δ. We consider a system where both super users and ordinary users have a Poisson arrival process and an exponential lifetime distribution (other types of arrival processes and lifetime distributions are studied in Section 5.3). We simulate each algorithm with its parameter varying from 0 and 1, and then measure the blocking rate of ordinary users (i.e., the ratio of the number of rejected ordinary users to the total number of arrived ordinary users) and the bandwidth guarantee probability (i.e., the probability that the system has sufficient overall bandwidth). Finally, for each algorithm we find the blocking rate corresponding to a given bandwidth guarantee probability δ.

Figure 2 shows the blocking rate of each admission control algorithm for a bandwidth guarantee probability δ. We can see that for the same admission control algorithm, a higher blocking rate is required to achieve a higher bandwidth guarantee probability. We can also see that to achieve the same bandwidth guarantee probability, DUAC has the smallest blocking rate, followed by SSUAC and finally SUAC. For example, in order to achieve 99.9% bandwidth guarantee probability, the blocking rate of DUAC is 19% (with β_{DUAC}=80%), the blocking rate of SSUAC is 41% (with β_{SSUAC}=48%), and the blocking rate of SUAC is 50% (with β_{SUAC}=50%).

In all the following simulations, we set $\beta_{DUAC} = 80\%$, $\beta_{SSUAC} = 48\%$, and β_{SUAC}=50%, so that all three algorithms can achieve the same bandwidth guarantee probability of 99.9% with a Poisson arrival process and an exponential lifetime distribution.

5.2 Retry Robustness

Figure 3 shows the bandwidth guarantee probability achieved by each admission control algorithm when a rejected user retries its admission request. These results are obtained for a system where both super users and ordinary users have a Poisson arrival process and an exponential lifetime distribution. We can see that the bandwidth guarantee probability achieved by both DUAC and SSUAC is always 99.9%, no matter how many times a rejected user retries its admission request. However, the bandwidth guarantee probability achieved by SUAC drops very quickly as a rejected user retries its admission request for more times.

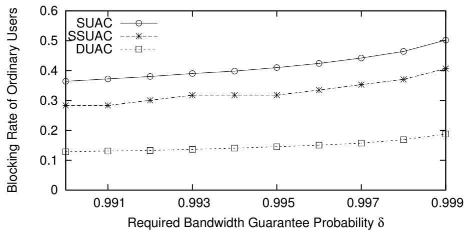

Figure 2: DUAC makes an admission decision based on the current channel state, and then has the smallest blocking rate among all three admission control algorithms in order to achieve a required bandwidth guarantee probability.

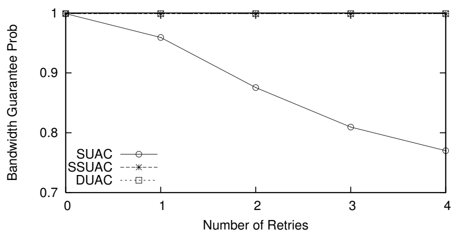

Figure 3: SUAC is not robust in case of user retries. That is, the bandwidth guarantee probability achieved by SUAC highly depends on how many times a rejected user retries its admission request.

5.3 User-Behavior Insensitivity

We first study the insensitivity to the user lifetime distribution. Motivated by the observation [18] that there are a small number of users who stay in the system for a very long time, we simulate a Pareto lifetime distribution. The lifetime of every user follows a Pareto distribution with a fixed mean $1/\mu^H = 1/\mu^L = 0.5$ and a shape parameter k varying from 1.52 to 1.52×8. The arrival process of a super user and an ordinary user is still a Poisson arrival process.

Figure 4 shows the state distribution $\mathbb{P}(N^H, N^L)$ of each admission control algorithm with $N^H = 10$ for $k = 1.52$ and 1.52×8. We can see that the state distribution of SUAC and SSUAC is insensitive to the user lifetime distribution, but the state distribution of DUAC is sensitive (although only slightly). This is consistent with Theorem 1. Figure 5 shows their bandwidth guarantee probabilities. We can see that the bandwidth guarantee probability achieved by SUAC and SSUAC does not depend on the lifetime distribution since they are insensitive to the lifetime distribution. We also observe that the bandwidth guarantee probability achieved by DUAC depends slightly on the lifetime distribution, and this is because the state distribution of DUAC is only slightly sensitive to the lifetime distribution.

Next, we study the insensitivity to the arrival process. Motivated by the observation [9] that the user arrival rates during different time intervals are different (e.g., higher arrival rate at the beginning of a program), we simulate an Interrupted Poisson Process (IPP). An IPP is an ON/OFF process, where both an ON period and an OFF period are exponentially distributed, and users arrive as a Poisson process only during an ON period. A super user arrives as an IPP with an average arrival rate of $\lambda^H = 50$, and an ordinary user arrives as an IPP with an average arrival rate of

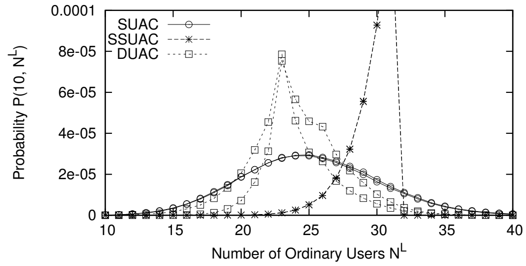

Figure 4: The state distribution of SUAC and SSUAC is insensitive to the lifetime distribution, but that of DUAC is sensitive (although only slightly). This is consistent with Theorem 1.

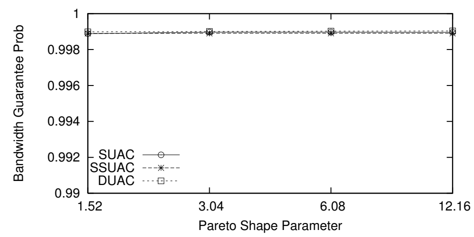

Figure 5: The bandwidth guarantee probability of SUAC and SSUAC does not depend on the lifetime distribution, and that of DUAC depends slightly on the lifetime distribution.

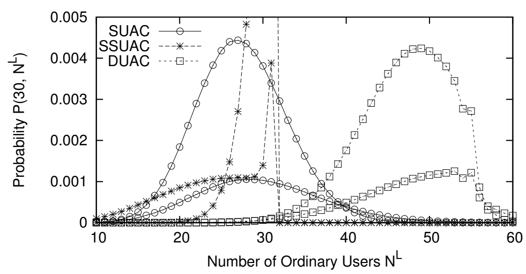

Figure 6: The state distribution of all three algorithms is sensitive to the user arrival processes.

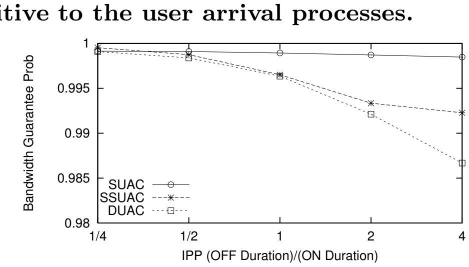

Figure 7: The bandwidth guarantee probability of SUAC slightly depends on the arrival processes, and that of SSUAC and DUAC highly depends on the arrival process.

$\lambda^L = 100$. We set the average ON period to 1/16, which is much shorter than the average user lifetime (i.e., 1/2). We vary the average OFF period from 1/4 to 4 times of the average ON period, and we also adjust the user arrival rate in an ON period accordingly to maintain the same average user arrival rate. Both a super user and an ordinary user have an exponentially distributed lifetime.

Figure 6 shows the state distribution $\mathbb{P}(N^H, N^L)$ of each admission control algorithm with $N^H = 30$ when the average OFF period is 1/4 and 4 times of the average ON period. We can see that the state distribution of all three algorithms is sensitive to the user arrival process. Figure 7 shows their bandwidth guarantee probabilities. We can see that the overall bandwidth guarantee probability of SUAC slightly depends on the user arrival processes, and that of SSUAC and DUAC highly depends on the user arrival process. Note that, when the average OFF period is 1/4 times of the average ON period, an IPP is very similar to a Poisson process, and this is why at that time, all three algorithms achieve 99.9% bandwidth guarantee probability.

Comparing Figure 2, Figure 4, and Figure 7, we can see that there is a tradeoff between the user blocking rate and user-behavior insensitivity. Intuitively, this is because in order to reduce the user blocking rate, an algorithm uses more channel state information, which however makes the algorithm more sensitive to the statistics of user behaviors. Comparing Figure 5 and Figure 7, we can see that the statistical bandwidth guarantee achieved by an algorithm is more sensitive to the distribution changes of user inter-arrival times than to the distribution changes of user lifetimes. The simulation results show that the bandwidth guarantee probability obtained with a Poisson user arrival process can be used as the upper bound of the probability for general user arrival processes. The bandwidth guarantee probability obtained with an exponential user lifetime distribution can be used as a good estimate of the probability for general user lifetime distributions.

6. CONCLUSION

In this paper, we studied a class of admission control algorithms in order to provide statistically guaranteed streaming quality to a P2P live streaming system. In particular, we studied their insensitivity to the fine statistics of user behaviors. In the future, we plan to study admission control algorithms for a P2P live streaming system where a user can simultaneously watch multiple channels, and an admission control algorithm is used to decide whether to accept not only the request of a new user to join the system but also the request of an existing user to watch a new channel.

7. REFERENCES

[1] Ahmed, M. Call admission control in wireless networks: A comprehensive survey. *IEEE Communications Surveys and Tutorials 7*, 1 (2005), 50–69.

[2] Bindal, R., and Cao, P. Can self-organizing P2P file distribution provide QoS guarantees. *ACM Operating Systems Review 40*, 3 (July 2006), 22–30.

[3] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., and Weiss, W. An architecture for differentiated services. *RFC 2475* (December 1998).

[4] Bonald, T., Jonckheere, M., and Proutiere, A. Insensitive load balancing. In *Proceedings of ACM SIGMETRICS/Performance* (New York, NY, June 2004).

[5] Bonald, T., and Proutiere, A. Insensitivity in processor-sharing networks. *Performance Evaluation 49*, 1-4 (September 2002), 193–209.

[6] Bonald, T., and Proutiere, A. Insensitive bandwidth sharing in data networks. *Queueing Systems 44*, 1 (May 2003), 69–100.

[7] Braden, R., Clark, D., and Shenker, S. Integrated services in the Internet architecture: an overview. *RFC 1633* (June 1994).

[8] Crowcroft, J., Hand, S., Mortier, R., Roscoe, T., and Warfield, A. QoS's downfall: At the bottom, or not at all. In *Proceedings of ACM SIGCOMM Workshop on Revisiting IP QoS: What have we learned, why do we care?* (Germany, August 2003).

[9] Hei, X., Liang, C., Liang, J., Liu, Y., and Ross, K. A measurement study of a large-scale P2P IPTV system. *IEEE Transactions on Multimedia 9*, 8 (December 2007), 1672–1687.

[10] Jamin, S., Danzig, P., Shenker, S., and Zhang, L. A measurement-based admission control algorithm for integrated services packet networks. In *Proceedings of ACM SIGCOMM* (Cambridge, MA, August 1995).

[11] Kumar, R., Liu, Y., and Ross, K. Stochastic fluid theory for P2P streaming systems. In *Proceedings of IEEE INFOCOM* (Anchorage, AK, May 2007).

[12] Kung, H., and Wu, C. Differentiated admission for peer-to-peer systems: Incentivizing peers to contribute their resources. In *Proceedings of Workshop on Economics of Peer-to-Peer Systems* (Berkely, CA, June 2003).

[13] Li, B., Qu, Y., Keung, Y., Xie, S., Lin, C., Liu, J., and Zhang, X. Inside the new Coolstreaming: principles, measurements and performance implications. In *Proceedings of IEEE INFOCOM* (Phoenix, AZ, April 2008).

[14] Liu, Y., Guo, Y., and Liang, C. A survey on peer-to-peer video streaming systems. *Journal of Peer-to-Peer Networking and Applications 1*, 1 (March 2008), 18–28.

[15] Perros, H., and Elsayed, K. Call admission control schemes: a review. *IEEE Communications Magazine 34*, 11 (November 1996), 82–91.

[16] Raghuveer, A., Dong, Y., and Du, D. On providing reliability guarantees in live video streaming with collaborative clients. In *Proceedings of MMCN* (San Jose, CA, January 2007).

[17] Siwko, J., and Rubin, I. Call admission control for capacity-varying networks. *Telecommnuication Systems 16*, 1 (2001), 15–40.

[18] Wang, F., Liu, J., and Xiong, Y. Stable peers: Existence, importance, and application in peer-to-peer live video streaming. In *Proceedings of IEEE INFOCOM* (Phoenix, AZ, April 2008).

[19] Wu, C., Li, B., and Zhao, S. Characterizing peer-to-peer streaming flows. *IEEE Journal on Selected Areas in Communications 25*, 9 (December 2007), 1612–1626.

[20] Xu, D., Hefeeda, M., Hambrusch, S., and Bhargava, B. On peer-to-peer media streaming. In *Proceedings of IEEE ICDCS* (Austria, July 2002).

[21] Zhang, M., Zhang, Q., and Yang, S. Understanding the power of pull-based streaming protocol: Can we do better? *IEEE Journal on Selected Areas in Communications 25*, 8 (2007), 1678–1694.