The production of peer-to-peer video-streaming networks

Dafu Lou *
SITE, University of Ottawa, Canada
SEE, Beijing University of Posts & Telecommunication, China
dlou@site.uottawa.ca

Yongyi Mao
SITE, University of Ottawa
Ottawa, Canada
yymao@site.uottawa.ca

Tet H. Yeap
SITE, University of Ottawa
Ottawa, Canada
tet@site.uottawa.ca

ABSTRACT

We introduce the notions of *production* and *saturation time* for peer-to-peer real-time video-streaming networks. Due to the fact that video-streaming is divided into small blocks to transmit, *production*, adopted from economics, is defined as the number of users that have obtained video block m by time t. *Saturation time* refers to the time when the system leaves the state where there have been more users requesting block m than the number of users that the system could have supplied. Based on those two notions, we provide a lower bound on the achievable production functions. Simulation results are provided for a heuristic protocol which appears to confirm the lower bound in terms of production and saturation time. The analysis results have also been confirmed by simulation using a stochastic model.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Communications Applications

General Terms

Theory

Keywords

Peer-to-peer (P2P), Video streaming, Video on-Demand (VoD), Fountain Codes

1. INTRODUCTION

The advantage of peer-to-peer networking architecture and the increasing demand of Internet-based entertainment have made the concept of peer-to-peer online video streaming a heated research topic in both academia and industry, where a great deal of recent literature (see, for example, [1, 2, 4–6, 9, 11, 12, 14]) exists.

Beyond the conventional approaches in networking towards peer-to-peer architectures, it is worth noting that the breakthrough of fountain coding [7, 8] has catalyzed the research activities in this area. In a generic setup of fountain coding, the transmitter encodes an information vector to an infinite sequence of randomly generated packets, and as long as the packets received by the receiver contain in total approximately the same number of the bits as the information vector, the receiver is able to decode the information vector. An elegant property of fountain codes is that they achieve the capacity of the erasure channels without the need to know the statistics of packet loss *a priori*. This property allows fountain codes to be used when multiple transmitters have common information to send to a given receiver. In this setting, the transmitters each simply encode the information independently using a fountain code, and irrespective of the packet-loss patterns, the receiver can decode when sufficient number of packets are received, upon which a rate equal to the sum of the link capacities along all links is achieved. This fact naturally makes fountain codes suitable as an efficient transport protocol for peer-to-peer video streaming.

In [10], the authors propose using fountain codes and more generally "rateless codes" for video transport in a peer-to-peer network and demonstrate the performance advantage of rateless codes in such applications. They also investigate and present distributed algorithms for optimal flow assignment in the network subject to bandwidth constraints.

The starting point of this paper has also been fountain coding, by which one may readily assume that when multiple transmitters communicate a common video block to a common receiver, the communication rate equal to the sum of the their respective link capacities is achievable. Our interest, different from most existing literature, is in characterizing the performance dynamics of the peer-to-peer network *under formation*, namely, the network in the phase when users are arriving. This problem is complicated by the fact that for each user demanding to download a video and watch it in *real time*, he needs to receive the video blocks in the playback sequence. This unique feature of real-time video streaming distinguishes it from file sharing — such as BitTorrent — in a peer-to-peer network. On one hand, the system should allow a newly arriving user to start watching the video as soon as possible. On the other hand, it should allow the users who have started watching the video to continue without interruption. Under limited resource constraints, these two key aspects of quality of service (QoS) (or "wait-

*The research is supported by Bell University Laboratories at University of Ottawa and by the cooperation with Beijing University of Posts & Telecommunication, Beijing, China.

P2P-TV'07, August 31, 2007, Kyoto, Japan.

ing" versus "jittering") clearly conflict with each other. It is therefore of great interest to understand the tradeoff between these two factors. However, when contemplating this problem, we realize that the fundamental quantities allowing one to characterize the system dynamics — as it evolves with time — have not been well studied. Leaving QoS aside, the main purpose of this paper is to establish a basic mathematical concept for real-time peer-to-peer video streaming for the network under formation. We thus introduce the notion of *production* of each video block as a function of time, so as to describe the dynamics of the system. In a nutshell, the production of block m at time t is the number of users who have completed downloading block m. In our analysis, we show that associated with production functions, for each video block there is a *saturation time*, after which every user requesting the block will obtain the block nearly immediately.

We also present a lower bound of the production functions and, correspondingly, an upper bound of the saturation time. The bounds are achieved with a simple transport protocol (Protocol A) based on fountain coding. The protocol is then improved by a modification that incorporates dynamic bandwidth allocation (Protocol B), for which we perform simulations and demonstrate their improved production and saturation time.

Closely related to this work is the work of [13]. In [13], the authors present an analytic framework for video streaming over peer-to-peer networks. Since the results of [13] only hold under fountain coding or similar underlying coding ideas, their results share certain similarity with the bounds presented in this paper. However, the work of [13] ignores the requirement that video content needs to be received and decoded in the playback sequence and treats a video as being equivalent to a large file. Additionally, in [13], the authors set up their problem as finding the rate-saturating point of the system, i.e., finding the time at which the user arrival rate equals the supply rate (or the derivative of production) of the system. When the system does not reject any requesting user (as is in our setting), we argue that the rate-saturating point is not the true saturating point of the system. Nevertheless, we acknowledge that the approach taken in [13] does inspire some of our developments.

To outline the rest of the paper, in Section 2, the system model and the notion of *production* and *saturation time* are introduced; Section 3 presents the lower bound of production and Protocol A; simulation results for both Protocol A and its modification, Protocol B, are presented in Section 4; and the paper is concluded in Section 5.

2. SYSTEM MODEL, PRODUCTION AND SATURATION

The system we consider includes a server $\mathcal{U}_0$ that has a video file and a set of users $\mathcal{U}_1, \mathcal{U}_2, \ldots$. The arrival of users and departure of users are stochastic processes in nature. Every user may arrive or depart at any time. However, we are more interested in the average cases here, not the performance of each individual user. Thus, a deterministic "flow model" — rather than a Poisson model — is used to model the arrival of users. Specifically, we model the number $N(t)$ of users having arrived at the system by time t as a real-valued function $N(t) = \lambda t$, for some positive arrival rate λ. The simulation, in which a Poisson process has been used, suggests that the results validate the "flow model".

On the other hand, to characterize the network formation in close forms, we assume that users will keep serving buffered content to others once they arrive. The assumption is not practical but we argue that the results can be extended to more realistic network settings because the departure of users will just cause the linear solution to become non-linear. The performance of measurement by the notions will still be valuable.

The objective of the streaming system is to provide video streaming service to all users so that shortly after a user arrives in the system and requests the video, it can play the video continuously without interruption.

The video file is divided into M consecutive blocks, each having size A bits. We will use $\{1, 2, \ldots, M\}$ to index these blocks, where block m contains video content preceding block $m+1$, for every $m \in \{1, 2, \ldots, M-1\}$. The playback rate of the video is assumed to be β bits/second, i.e., each block when playing takes A/β seconds to finish. The total size of the video file is then MA, and the total playback duration (or length) of the video is MA/β.

Borrowing a term from economics [3], we now introduce the notion of *production* to characterize the performance of such a system under any given transport protocol.

Definition 1 (Production). *For any given network and under any given video-streaming protocol, the production $N_m(t)$ of block m at time t is defined as the number of users that have obtained block m by time t.*

Due to the stochastic nature of the network, production $N_m(t)$ is in general a random process. However, as we are mostly interested in the average behavior of $N_m(t)$, we treat $N_m(t)$, again, as a real-valued deterministic function.

It is clear that $N_m(t)$ is upper bounded by $N(t)$ for every m. Thus, how rapidly and closely $N_m(t)$ approaches $N(t)$ indicates the system efficiency for distributing the m^{th} block. We note that unlike most other performance metrics in the literature, which characterize the performance of the streaming system under equilibrium conditions, the notion of production proposed here intends to characterize the performance dynamics of the system as it evolves with the equilibrium.

As the users arrive in the system, there is an overlay network, established on top of the IP network, between each user and the server and between every two users. We assume that in the overlay network, a user may acquire a demanded video block either from the server or from any other users. The overlay network is characterized by a homogeneous loss rate γ. More specifically, each logical link between two users or between the server and a user assumes an independent state for each transmitted packet such that the transmitted packet suffers from packet loss with probability γ. We note that the homogeneous assumption of loss rate has no particular significance in the forthcoming analysis. In addition, we assume that the loss rate is unknown to the server and to all users.

Definition 2 (Production Potential). *Under any given video-streaming protocol, the production potential $S_m(t)$ of block m at time t is defined by*

$$S_m(t) := \frac{1-\gamma}{A} \int_0^t \left(s_m(\tau) + b_m(\tau) N_m(\tau)\right) d\tau,$$

where $s_m(t)$ and $b_m(t)$, depending on the protocol, are respectively the server's output bandwidth allocated to transmitting block m at time t and the average user output bandwidth allocated to transmitting block m at time t, where the average is the overall users having obtained block m.

As each user having obtained block m may become a potential supplier of block m to other users, the notion of production potential $S_m(t)$ indicates the possible production of block m up to time t if the production were not limited by the number of users needing block m by time t. In this definition, it has been implicitly assumed that each user has a sufficient buffer so that a user having obtained block m can be a "permanent" supplier of block m.

Now we assume that after a user has obtained block $m-1$, the user will immediately request block m. That is, $N_{m-1}(t)$ is also equal to the number of users needing block $m-1$ at time t. Then the following notions of "saturation" and "unsaturation" are natural.

DEFINITION 3 (m-UNSAT AND SATURATION TIME). *At any time T, the system is said to be block-m-unsaturating, or m-UNSAT, if for any $t \in [0, T]$,*

$$N_{m-1}(t) > S_m(t), \text{ or } N_{m-1}(t) = S_m(t) = 0.$$

where $N_0(t) := N(t)$. The saturation time $\widehat{T}_m$ *of block m is defined as*

$$\widehat{T}_m := \sup\{T : \text{ the system is } m\text{−UNSAT at } T\}.$$

When $N_{m-1}(t)$ is interpreted as the number of users that have requested block m since time 0, the rationale of coining the term "block-m-unsaturating" is to indicate, except for the extreme case of $N_{m-1}(t) = S_m(t) = 0$, the state of the system where there have been more users requesting block m than the number of users that the system could have supplied. It is clear by the definition that the system is m-UNSAT at $t = 0$ for all m. Then the saturation time $\widehat{T}_m$ of block m is the phase-transition point for the system to leave the m-UNSAT state. It is then clear that for a well-behaved protocol, one may want the saturation time to have the following properties.

- Saturation time $\widehat{T}_m$ is finite for every m and as small as possible.
- At time $t > \widehat{T}_m$, $N_m(t)$ is close to λt, namely, every user arriving after time $\widehat{T}_m$ obtains every block very quickly.

For the rest of this paper, we assume that the total output bandwidth of the server is s bits/second and the output bandwidth of each user is b bits/second. We impose no restriction on the downloading bandwidth of each user, which allows for a simplification of analysis. Additionally, we ignore all other possible overheads for example, at every time t, we assume that each user $\mathcal{U}_i$ is instantaneously updated with the information concerning which other users have the video block that user $\mathcal{U}_i$ demands.

3. A LOWER BOUND OF PRODUCTION: PROTOCOL A

We will present a lower bound on achievable production $N_m(t)$, which relies on the following mild assumption:

$$\lambda \widehat{T}_1 > e. \tag{1}$$

This restriction essentially assumes that by the time the system leaves 1-UNSAT phase, at least 3 user have arrived. This assumption evidently holds true for all practical scenarios under any reasonable protocol. In fact, the only cases in which this assumption fail are when the arrival rate is very low and/or the server has sufficient output bandwidth to drive the system out of 1-UNSAT phase quickly; in both of the cases, one however would argue that there is simply no need of using any peer-to-peer approach.

Using $\mathbf{u}(t)$ to denote the unit-step function, the following lower bound can be proved.

THEOREM 1. *Under assumption (1), there exists a transport protocol such that*

1. $N_m(t) \geq \min\left(\lambda t, e^{\frac{b(1-\gamma)}{AM}(t-T_m)}\mathbf{u}(t - T_m)\right)$ *for all* $m \in \{1, 2, \ldots M\}$, *where*
$$T_m = mA/(1-\gamma)s + 1/\lambda;$$
and
2. $\widehat{T}_m$ *is finite for every m.*

3.1 Proof of Theorem 1: Protocol A

We prove Theorem 1 by constructing a protocol, which we call Protocol A, that achieves production functions no lower than the lower bounds in Theorem 1. It will become evident that the essence of the Protocol A is to induce a linear-system dynamics (over a certain time window) so that analytic solutions can be easily derived.

In Protocol A, after the first user $\mathcal{U}_1$ has completely obtained block 1, server $\mathcal{U}_0$ transmits to $\mathcal{U}_1$ block 2. The server continues in this way until $\mathcal{U}_1$ has obtained block M. Then the server terminates transmission and stays idle. Each user $\mathcal{U}_i$ allocates $1/M$ fraction of its output bandwidth for distributing block m for every m. Every user $\mathcal{U}_i$ requests video blocks in sequence, and immediately after receiving a request of the block that it has already obtained, it uses its allocated output bandwidth to transmit the block, coded by its fountain code, to the requesting user. We will use $N_m^A(t), S_m^A(t)$ and $\widehat{T}_m^A$ to denote the corresponding quantities achieved by Protocol A.

We note that in Protocol A the time by which exactly one user has obtained block m completely is T_m, or $mA/(1-\gamma)s + 1/\lambda$ in the theorem. This follows from that for every m it is always $\mathcal{U}_1$ who obtains block m first, that it takes $1/\lambda$ seconds for $\mathcal{U}_1$ to arrive in the system, and that it takes $A/(1-\gamma)s$ seconds for $\mathcal{U}_1$ to obtain each block. The following observations are in order.

1. The system is m-UNSAT at every $t \leq T_m$. This simply follows from that $N_m^A(t) = 0$ at any $t < T_m$ and $S_m^A(t) = 0$ at every $t \leq T_m$.
2. If the system is m-UNSAT at time t, then
$$N_m^A(t) = S_m^A(t). \tag{2}$$
This is due to an idealistic assumption that if the number of users that the system can supply block m up to

time t is more than the number of users that have requested block m up to time t, all output bandwidth allocated to block m will be completely used for supplying users requesting block m. Thus, at m-UNSAT phase, the number of users having received block m by time t is equal to $S_m^A(t)$.

These observations give rise to the following lemma.

LEMMA 1. *If the system is block-m-UNSAT at time t, then*

$$N_m^A(t) = e^{\frac{(1-\gamma)b}{AM}(t-T_m)}\mathbf{u}(t-T_m),$$

Proof: Clearly for $t < T_m$, $N^A(t) = 0$. At $t \geq T_m$, if the system is m-UNSAT, applying (2), we have

$$N_m^A(t) = \frac{(1-\gamma)b}{MA}\int_0^t N_m^A(\tau)d\tau. \quad (3)$$

Solving this equation with initial condition $N_m^A(T_m) = 1$, we get

$$N_m^A(t) = e^{\frac{(1-\gamma)b}{AM}(t-T_m)},$$

for $t \geq T_m$. Combining the case of $t < T_m$ and the case of $t \geq T_m$, the lemma is proved. □

We note that during the time period in which the system is m-UNSAT, $N_m^A(t)$ behaves according to a linear-system dynamics following (3).

LEMMA 2. *For all $m \in \{1, 2, \ldots, M\}$, $\widehat{T}_m^A$ is finite, and*

$$N_m^A(t) = \begin{cases} 0, & t < T_m \\ e^{\frac{(1-\gamma)b}{AM}(t-T_m)}, & T_m \leq t \leq \widehat{T}_m \\ \lambda t, & t > \widehat{T}_m^A(t) \end{cases}$$

Proof: We will prove this result by induction.

First consider the case of $m = 1$. Clearly $T_1 < \widehat{T}_1^A$. During time $[T_1, \widehat{T}_1^A)$, the system is 1-UNSAT. Thus $N_1^A(t) = e^{\frac{(1-\gamma)b}{AM}(t-T_1)}$, during $[T_1, \widehat{T}_1^A)$. In addition, $1 = N_1^A(T_1) < \lambda T_1$, i.e., the exponential curve at $t = T_1$ lies below line λt. Then as t increases, the exponential curve $e^{\frac{(1-\gamma)b}{AM}(t-T_1)}$ and line λt must intersect. Furthermore, by definition of saturation time, $\widehat{T}_1^A$ is precisely the intersecting point of the two curves. Then we conclude $\widehat{T}_1^A$ is finite.

Let $\Delta > 0$ be arbitrarily small. Since it is always true that $N_0^A(t) \geq N_1^A(t)$, $N_1^A(t) = N_0^A(t) = \lambda t$ over interval $[\widehat{T}_1^A, \widehat{T}_1^A + \Delta]$. Then when $t \in [\widehat{T}_1^A, \widehat{T}_1^A + \Delta]$, $S_1^A(t) = \lambda\widehat{T}_1^A + \frac{(1-\gamma)b}{MA}\int_{\widehat{T}_1^A}^t \lambda\tau d\tau$ giving rise to $\frac{dS_1^A}{dt}(t) = \frac{(1-\gamma)b\lambda t}{MA}$ at every $t \in [\widehat{T}_1^A, \widehat{T}_1^A + \Delta]$.

Now it is possible to show that

$$\begin{aligned} \frac{dS_1^A}{dt}(\widehat{T}_1^A + \Delta) &> \frac{dS_1^A}{dt}(\widehat{T}_1^A) = \frac{(1-\gamma)b}{MA}\lambda\widehat{T}_1^A \\ &= \lambda\left(\ln(\lambda\widehat{T}_1^A) + \frac{bA}{sM} + \frac{(1-\gamma)b}{MA\lambda}\right) \end{aligned}$$

when considering $\lambda\widehat{T}_1^A = e^{\frac{(1-\gamma)b}{MA}(\widehat{T}_1^A - T_1)}$, which results in $\frac{(1-\gamma)b}{MA}\widehat{T}_1^A = \ln(\lambda\widehat{T}_1^A) + \frac{(1-\gamma)b}{MA}T_1$, where $T_1 = A/(1-\gamma)s + 1/\lambda$. Then following the assumption of Equation (1), we conclude that

$$\frac{dS_1^A}{dt}(\widehat{T}_1^A + \Delta) > \lambda(\widehat{T}_1^A + \Delta) = \frac{dN_0}{dt}(\widehat{T}_1^A + \Delta) = \lambda.$$

This suggests that after saturation time $\widehat{T}_1^A$, the potential growth rate of the number of users supplied with block 1 always tends to exceed growth rate of the number of users having requested block 1. As consequence, $N_1^A(t)$ is bounded by and stays on $N_0(t)$, or λt.

Now as the inductive assumption, assume the claims of the lemma hold true for a given $m = k$. That is, i.e. $\widehat{T}_k^A$ is finite and

$$N_k^A(t) = \begin{cases} 0, & t < T_k \\ e^{\frac{(1-\gamma)b}{AM}(t-T_k)}, & T_k \leq t \leq \widehat{T}_k^A \\ \lambda t, & t > \widehat{T}_k^A(t) \end{cases}$$

When $m = k+1$, the number of users having requested block-$(k+1)$ is $N_k^A(t)$, which equals λt for $t > \widehat{T}_k^A$. By definition of $\widehat{T}_m^A$, it is clear that the system is $(k+1)$-UNSAT over duration $[0, \widehat{T}_{k+1}^A)$, whether or not $\widehat{T}_{k+1}^A$ is finite. Thus $N_{k+1}^A(t)$ (and $S_{k+1}^A(t)$) takes value 0 when $t < T_{k+1}$ and assumes the exponential curve $e^{\frac{(1-\gamma)b}{AM}(t-T_{k+1})}$, when $T_{k+1} \leq t < \widehat{T}_{k+1}^A$. By noting that exponential curves $e^{\frac{(1-\gamma)b}{AM}(t-T_{k+1})}$ and $e^{\frac{(1-\gamma)b}{AM}(t-T_k)}$ will never intersect after time T_{k+1}, we see that curve $e^{\frac{(1-\gamma)b}{AM}(t-T_{k+1})}$ must intersect with $N_k^A(t)$ when the later assumes the values on line λt, namely, at some time after time $\widehat{T}_k^A$. This intersection point is, by definition of saturation time, at time $\widehat{T}_{k+1}^A$. Thus, $\widehat{T}_{k+1}^A$ is finite, and clearly $\widehat{T}_{k+1}^A > \widehat{T}_k^A$.

By the same argument as in the proof for the case of $m = 1$, it can be shown that after time $\widehat{T}_{k+1}^A$, $N_{k+1}^A(t)$ stays on $N_k^A(t)$, or λt. □

Now we are ready to prove the theorem.

Proof: As Protocol A is only a particular protocol using fountain codes, there exists a protocol that performs no worse than Protocol A in terms of production functions. By identifying the lower bound in claim (1) of the theorem as $N_m^A(t)$, claim (1) is proved. It is also obvious that a protocol satisfying claim (1) of the theorem will also gives rise to claim (2), since all production curves $N_m(t)$ of such a protocol are on the left side of $N_m^A(t)$, giving rise to saturation time $\widehat{T}_m$ no later than $\widehat{T}_m^A$. This proves claim (2). □

Immediately follow the proof of theorem, the following result is implied.

COROLLARY 1. *There exists a fountain-coded transport protocol such that $\widehat{T}_m \leq \widehat{T}_m^A$, for all $m \in \{1, 2, \ldots M\}$.*

We argue that the Theorem 1 gives a lower bound of performance for fountain codes based on transport protocols, as it is clear that Protocol A is not optimal because every user uses a fixed amount of output bandwidth for each block. A fountain-coded transport protocol with a dynamic output bandwidth allocation algorithm should have better performance than Protocol A in terms of production. In next Section, we present Protocol B, which allocates the output bandwidth for each block dynamically.

4. SIMULATION RESULTS AND DISCUSSION

We performed simulations for Protocol A and Protocol B. In our simulations, to meet the setting of the practical system, the user arrival process is modeled as a Poisson process

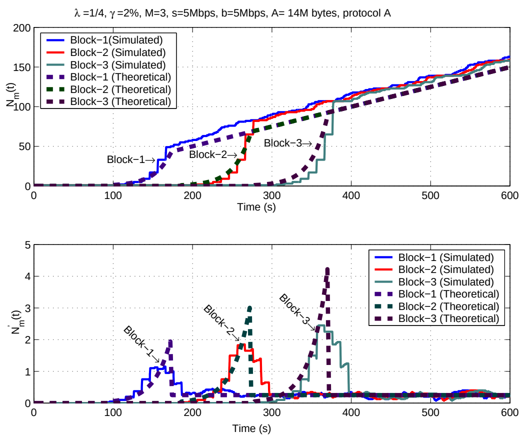

Figure 1: Theoretical and simulated production achieved by Protocol A (top) and the derivative of theoretical and simulated production (bottom).

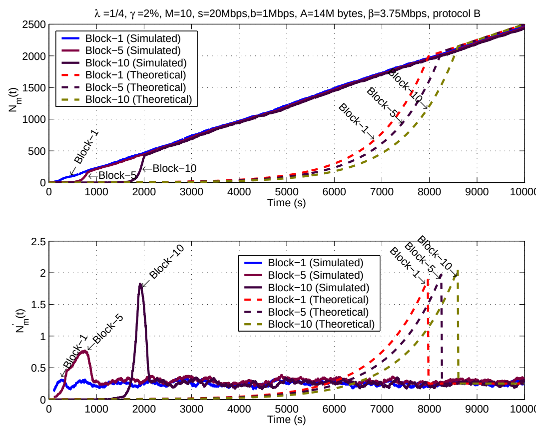

Figure 2: Simulated production achieved by Protocol B in comparison with the lower bounds of Theorem 1 (top); the derivative of simulated production achieved by Protocol B and that of the lower bounds of Theorem 1 (bottom).

with λ users per second. The video server is configured with a high quality of video and can continuously supply to all peers, which performs a video on-demand server. As Protocol A mainly serves as a theoretical construction for bounding the system performance, Protocol A is simulated only for the purpose of confirming the bounds presented in Theorem 1 and validating the flow model used in establishing the theorem. We also designed another heuristic protocol, Protocol B, with a dynamic output bandwidth allocation rather than the fixed bandwidth allocation in Protocol A. Simulation results are presented in terms of production and saturation time.

4.1 Simulation Results of Protocol A

The bounds of Theorem 1 and the simulated production functions achieved by Protocol A are plotted in Figure 1 (top), and their derivatives are plotted in Figure 1 (bottom). The derivative plots in Figure 1 (bottom) serve to demonstrate more sharply the saturation times (corresponding to the sudden drop in the derivative plots) and the instantaneous "supplying capability" of video blocks (equivalent to the instantaneous total output bandwidth, up to scale). From the figure, we see that the theoretical results and simulated results match well, confirming the validity of the theorem and flow model of user arrival. Specifically, the following behaviours are observed. First, during the UNSAT phase, the production grows exponentially until it hits the straight line $N(t)$ characterizing the arrival process whereby the UNSAT phase is terminated and then the production grows according to $N(t)$. Second, uniform spacing between saturation times of consecutive blocks is observed.

4.2 Simulation Results of Protocol B

Similar to Protocol A, Protocol B also uses fountain codes. In Protocol B, each user $\mathcal{U}_i$, as a receiver, may continuously send requests for the block it needs. The user continuously updates a quantity β'_i, the rule of which will be specified shortly. The physical meaning of β'_i is an estimate of the total rate of receiving the block in need. The user compares β'_i with $\beta(1+\delta)$ — the playback rate with a small margin — for some small δ (fixed to 0.1 in our simulations). If $\beta'_i < \beta(1+\delta)$, it sends a request of the block together with its "demanded rate" $\beta(1+\delta) - \beta'_i$.

As a transmitter, each user $\mathcal{U}_j$ that has free output bandwidth randomly select a block number of the blocks that it has in its buffer. Then it randomly chooses a user from the users requesting the selected block. If the chosen user's "demanded rate" is smaller than $\mathcal{U}_j$'s available free bandwidth, it uses a portion of its bandwidth equal to the demanded rate to transmit the selected block to the chosen user. If the chosen user's demanded bandwidth is larger than $\mathcal{U}_j$'s free bandwidth, it uses all its free bandwidth to transmit the selected block to the chosen user. It also informs the chosen user of the bandwidth that it is using to transmit to the chosen user. We call this bandwidth the "promised bandwidth". It repeats this process until there is no free output bandwidth. The server also uses the same algorithm to allocate output bandwidth.

Each user $\mathcal{U}_i$, as a receiver, constantly updates its β'_i as the sum of the promised bandwidths of all users transmitting to it. It then sends the block requests and the "demanded rate".

We note that the proposed Protocol B, although suboptimal, dynamically allocates user output bandwidth to adapt to the evolution of the system. This proposal, to the best of our knowledge, is believed to be the first such effort.

The simulated production achieved by Protocol B and the lower bounds of Theorem 1 are plotted in Figure 2 (top) together with their derivatives (bottom).

First, one observes that the saturation time $\widehat{T}_m$ is much earlier in Protocol B than that in Protocol A. Second, these results suggest that the production for different block m does not follow the same growth curve, demonstrating much more complex dynamics. Third, it is important to realize that the difference between Protocol B and Protocol A is primarily in how each user's output bandwidth is allocated across the video blocks, suggesting the importance of bandwidth allocation in peer-to-peer video-streaming systems. Finally it is worth noting that Protocol B performs quite well and is already a nearly working protocol for high-quality (DVD) video-streaming, particularly when one notices that the parameters chosen are quite close to the realistic settings.

We also consider the realistic network scenario, where the user departs freely with limited buffer size. We found it difficult to analytically derive the closed forms of Theorem 1. We believe that the proposed notions of production and saturation time can be extended to realistic network settings. Most of the characters we have derived under our theoretical model are still suitable for realistic networks. We have performed simulations to measure production and saturation time for the realistic network settings. Length constraint precludes the simulation results here. We simply point out that the results support our argument.

5. CONCLUSION

We study the dynamics of peer-to-peer video streaming over peer-to-peer networks as the users are joining the network, where we introduce the notion of production and saturation time as fundamental measures characterizing the system dynamics in this phase. We present a lower bound of the production functions and, correspondingly, an upper bound on the saturation time. We also propose a heuristic protocol, based on fountain coding, which outperforms the lower bound. The analysis results extended to realistic network settings have also been confirmed by the simulation.

6. REFERENCES

[1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung. Network information flow. *IEEE Trans. Inform. Theory*, 46:1204–1216, Jul. 2000.

[2] S. Deb, M. Medard, and C. Choute. Algebraic gossip: a network coding approach to optimal multiple rumor mongering. *IEEE Trans. Inform. Theory*, 52:2486–2507, June 2006.

[3] M. Fuss and D. L. McFadden. *Production Economics: A Dual Approach to Theory and Applications.* Amsterdam: North-Holland, 1978.

[4] L. Guo, S. Chen, Z. Xiao, E. Tan, X. Ding, and X. Zhang. A performance study of BitTorrent-like peer-to-peer systems. *IEEE J. Select. Areas Commun., Special Issue on Peer-to-Peer Communications and Applications*, January 2007.

[5] A. Habib and J. Chuang. Service differentiated peer selection: An incentive mechanism for peer-to-peer media streaming. *IEEE Trans. Multimedia*, 51:1204–1216, June 2006.

[6] Y. Huang, Y.-F. Chen, R. Jana, B. Wei, M. Rabinovich, and Z. Xiao. Challenges of P2P streaming technologies for IPTV services. *Proc. of IPTV Workshop, International World Wide Web Conference*, May 2006.

[7] M. Luby. LT codes. *FOCS*, page 271, 2002.

[8] A. Shokrollahi. Raptor codes. *IEEE Trans. Inform. Theory*, 52(6):2551–2567, 2006.

[9] D. A. Tran, K. Hua, and T. Do. A peer-to-peer architecture for media streaming. *IEEE J. Select. Areas Commun.*, 22, 2004.

[10] C. Wu and B. Li. rStream: Resilient and optimal peer-to-peer streaming with rateless codes. *Tech. Rep., under submission for journal publication*, 2006.

[11] D. Xu, M. Hefeeda, S. Hambrush, and B.Bhargava. On peer-to-peer media streaming. *Proc. of IEEE ICDCS*, pages 363–371, July 2002.

[12] X. Yang and G. de Veciana. Service capacity of peer to peer networks. *Proc. of IEEE INFOCOM*, 2004.

[13] Y.Tu, J.Sun, M.Hefeeda, and S.Prabhakar. An analytical study of peer-to-peer media streaming systems. *ACM TOMCCAP*, 1:354 – 376, Nov. 2005.

[14] X. Zhang, J. Liu, B. Li, and T.-S. P. Yum. CoolStreaming/DONet: A data-driven overlay. *Proc. of IEEE INFOCOM*, March 2005.