Demand Forecast and Performance Prediction in Peer-Assisted On-Demand Streaming Systems

Di Niu, Zimu Liu, Baochun Li
Department of Electrical and Computer Engineering
University of Toronto
{dniu, zimu, bli}@eecg.toronto.edu

Shuqiao Zhao
Multimedia Development Group
UUSee, Inc.
shuqiao.zhao@gmail.com

***Abstract*—Peer-assisted on-demand video streaming services are extremely large-scale distributed systems on the Internet. Automated demand forecast and performance prediction, if implemented, can help with capacity planning and quality control so that sufficient server bandwidth can always be supplied to each video channel without incurring wastage. In this paper, we use time-series analysis techniques to automatically predict the online population, the peer upload and the server bandwidth demand in each video channel, based on the learning of both human factors and system dynamics from online measurements. The proposed mechanisms are evaluated on a large dataset collected from a commercial Internet video-on-demand system.**

I. Introduction

Video-on-demand (VoD) has become an enormously popular service on the Internet. A successful Internet VoD system usually attracts millions of users, with more than thousands of users concurrently online in a popular video channel at peak times. Following the real-world success of the peer-to-peer (P2P) architecture as a solution to live video streaming (*e.g.*, CoolStreaming [1]), peer assistance has also been introduced into VoD services to increase system scalability and alleviate server bottlenecks [2], [3]. In peer-assisted VoD systems, peers download from both servers and other online peers.

As bandwidth is a scarce resource, it is vital for a VoD service to carefully provision its server capacity, in order to meet user demands while not incurring any resource wastage. Such capacity planning decisions are best to be made proactively prior to demand changes in each video channel, and will thus benefit from an accurate user demand forecast. In this paper, we analyze the operational traces collected from UUSee Inc., one of the leading peer-assisted media content providers in China. Leveraging both the human factors and system dynamics revealed in the traces, we investigate the feasibility of predicting peer population, peer bandwidth contribution and the demand for server bandwidth in the system.

From the real-world traces, we discover that the population evolution in a video channel is highly predictable, as users exhibit periodic viewing behavior and their interest in the video diminishes gradually after it is released. Following the Box-Jenkins method, we introduce seasonal ARIMA models [4] to accurately predict future population given its past observations. Furthermore, popular videos released around the same time of day demonstrate similar population evolution patterns at the beginning due to tractable behaviors of Internet users. This enables us to infer the initial population of a new channel from the statistics of similar channels released earlier. A novel probabilistic model based on the regression of mixtures of gaussians is proposed to account for such a phenomenon. As a peer-assisted service heavily relies on the bandwidth supplies from peers, we also propose to predict the peer upload contribution based on a linear stochastic model using the population evolution as a leading indicator.

Based on the prediction of online population and peer upload, we have made possible a mechanism that can automatically forecast the demand for server bandwidth in each video channel up to 2.5 hours into the future, with the aid of online measurements. Extensive evaluation of the proposed methods is conducted based on the traces of 40 channels (with peak population per channel up to 8000) in a 21-day period spanning the entire 2008 Beijing Olympics.

A. Relation to Prior Work

The importance of demand or popularity estimation in Internet VoD systems has been recognized recently. It is shown that estimating time-varying demands in a large-scale IPTV network can help the system optimally place content on its servers [5]. Toward this goal, the recent history has been used as an estimate of future demand in each video channel [5]. Apparently, this simple method does not yield accurate predictions. In contrast, we introduce a systematic approach of time-series analysis to capture the periodicity, trends and patterns that are unique to VoD system statistics, achieving high prediction accuracy.

The popularity of a new video can be inferred by learning users' preferences on similar videos published earlier, such as using collaborative filtering [6]. In this paper, we propose a novel regression model to infer user demands for new videos, which has a strong physical explanation based on the video release time and diurnal access behavior of Internet users. The new scheme does not require collecting a large amount of user preference information. We also initiate the first attempt to predict peer bandwidth contribution and server bandwidth demand in peer-assisted video streaming services using linear stochastic models.

II. Capacity Planning in VoD Systems

UUSee is one of the leading commercial P2P multimedia solution providers in China, simultaneously broadcasting thousands of on-demand video channels to millions of users dis-

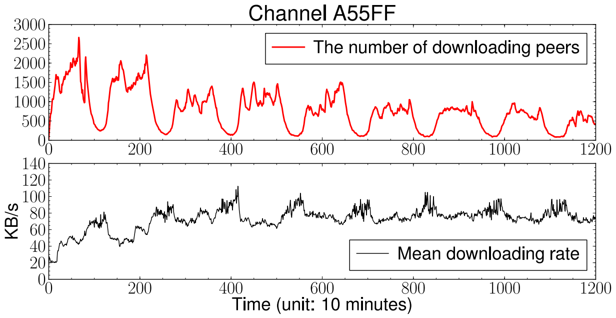


Fig. 1. The number of downloading peers and their mean downloading rate in a typical video channel after the channel was released. The length of a time period is 10 minutes. The video bit rate is 400 kbps.

tributed across over 40 countries in the world. The UUSee on-demand streaming system organizes users that are interested in the same video into a random mesh network. In such a mesh network, there are three types of nodes, namely *media servers*, *downloading peers*, *i.e.*, peers that are currently downloading and watching the video, and *cached peers*, *i.e.*, online peers that have previously watched the video and cached it in their file systems. A downloading peer can download media blocks from all three kinds of peers.

Consider a particular video channel. Let N_t be the online population, *i.e.*, the number of downloading peers in the channel at time t. Denote s_t, u_t, and c_t the average bandwidth each downloading peer receives from the servers, other downloading peers, and cached peers. Hence,

$$r_t = s_t + u_t + c_t \tag{1}$$

is the mean downloading rate of downloading peers in the channel, and $S_t = s_t N_t$ is the aggregate server bandwidth used by the channel.

In VOD services, capacity planning is the process of deciding the amount of server resources, especially bandwidth, that need to be provisioned to each channel in order to meet the user demand. The server bandwidth demanded by each channel varies over time due to time-varying video popularity and peer upload contribution. A discrepancy between the serving capacity and user demands results in inefficiency, either in under-utilized server resources or unfulfilled user demands.

Like most other Internet VoD companies, UUSee over-provisions its server capacity with the hope to maintain a stable quality of service. This causes significant server under-utilization during periods of low demand or high peer upload contribution. However, there still exist unfulfilled demands in the system. For example, the evolution of online population and their mean downloading rate are plotted in Fig. 1 for a typical Olympics video that attracted a flash crowd upon its release. Clearly, there is a performance issue during the first two days (before $t = 300$) , marked by a relatively low mean downloading rate. The reason for such a bandwidth shortage is because the video was not replicated on enough servers and the aggregate server bandwidth to serve the channel was not sufficient to meet the demand.

To minimize the discrepancy between a system's serving capacity and user demands, it is therefore vital for the system to accurately forecast user demands and provisioning servers proactively. Let S_t^r be the aggregate server bandwidth demanded by a channel at time t to achieve a target mean downloading rate of R. To maintain smooth playback, R is usually greater than the video bit rate in order to accommodate the fluctuation in downloading rates. Towards demand forecast, we propose to collect statistics about N_t, u_t and c_t up to time t, and make prediction of S_{t+h}^r at future time $t+h$ required by the channel so that the achieved $r_t \geq R$. We first predict N_{t+h}, u_{t+h} and c_{t+h} to obtain the estimates $\hat{N}_{t+h}$, $\hat{u}_{t+h}$ and $\hat{c}_{t+h}$. The server bandwidth demand S_{t+h}^r can thus be forecasted as

$$\hat{S}_{t+h}^r = \hat{N}_{t+h}(R - \hat{u}_{t+h} - \hat{c}_{t+h}). \tag{2}$$

To facilitate research and analysis, we have implemented detailed measurement capabilities within each UUSee client, which sends its vital statistics to our dedicated logging servers every 10 minutes. (A 10-minute sampling interval proves to be sufficiently fine-grained for capacity planning purposes without incurring overly high computational cost.) The data for validation in this paper feature a set of traces collected from 40 video channels during a 21-day period spanning the entire period of 2008 Summer Olympics.

III. Population Prediction

Among all 40 channels in our traces, 32 were published during the measurement period and attracted flash crowds upon release, which we call *flash-crowd channels*. The other 8 were released earlier and exhibit a steady daily population of a smaller size, which we call *steady-state channels*. We use the Box-Jenkins approach [4] to predict the future population in both kinds of channels and introduce regression methods to infer the initial population evolution of flash-crowd channels.

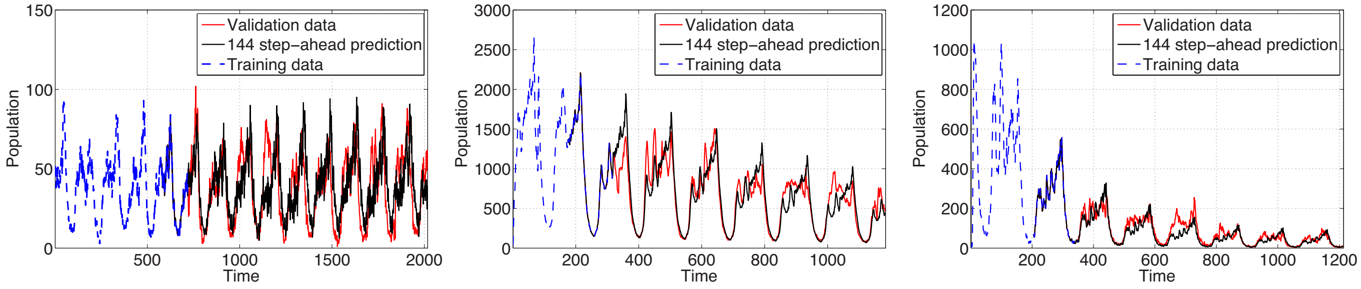
A. Population Prediction: a Box-Jenkins Approach

Given the population time series of a channel in the past few days, we can make fine-grained prediction into its future evolution, leveraging the trend, periodicity and autocorrelation exhibited in its own history. Due to the periodicity (diurnal pattern) in all the channels and the decreasing trend in flash-crowd channels, N_t of any channel is clearly non-stationary [1]. However, following the Box-Jenkins method, we can eliminate both the periodicity and trend via differencing to obtain a stationary yet autocorrelated series, which can then be characterized by models for stationary processes such as ARMA (autoregressive moving-average). Now we briefly outline the so-called seasonal ARIMA (autoregressive integrated moving-average) model [4] for non-stationary population prediction.

Given a time series of interest $\{Y_t\}$, define the backward shift operator B by $BY_t = Y_{t-1}$, the lag-1 difference operator ∇ by $\nabla Y_t = Y_t - Y_{t-1} = (1 - B)Y_t$, and the lag-$d$ difference operator ∇_d by $\nabla_d Y_t = Y_t - Y_{t-d} = (1 - B^d)Y_t$.

For steady-state channels, the population series $\{N_t\}$ has a period of 144 (one day) with no trend. We therefore de-seasonalize $\{N_t\}$ to obtain a stationary series $\tilde{N}_t = \nabla_{144} N_t$,

[1] A process $\{Y_t\}$ is (weakly) stationary if its mean $\mathbf{E}[Y_t]$ and its covariance function $\text{Cov}(Y_{t+h}, Y_t)$ at each lag h are independent of t.



(a) Steady state channel B830

(b) Flash-crowd channel A55F, $d = 0$

(c) Flash-crowd channel 7A40, $d = 1$

Fig. 2. One day-ahead population prediction based on seasonal ARIMA models for three different channels.

which can then be modeled as an ARMA(p, q) process. Equivalently, $\{N_t\}$ is modeled by the seasonal ARIMA model

$$\phi(B)\nabla_{144} N_t = \theta(B) Z_t, \tag{3}$$

where $\{Z_t\} \sim \text{WN}(0, \sigma^2)$ denotes the uncorrelated white noise with zero mean, and $\phi(B) = 1 - \phi_1 B - \ldots - \phi_p B^p$ and $\theta(B) = 1 + \theta_1 B + \ldots + \theta_q B^q$ are polynomial operators in B of degrees p and q.

For flash-crowd channels, however, both trend and seasonality exist in $\{N_t\}$, as shown in Fig. 1. The daily fluctuation in N_t also decreases as t increases. We therefore first apply transformation $\log(\cdot)$ to $\{N_t\}$ to equalize the fluctuation. We then apply transformation ∇_{144} to $\{\log(N_t)\}$ to remove periodicity, and difference $\nabla_{144} \log(N_t)$ for d times to remove the trend, obtaining the stationary series $\tilde{N}(t) = \nabla^d \nabla_{144} \log(N_t)$, which is well explained by an ARMA(p, q) process. The corresponding seasonal ARIMA model for $\{N_t\}$ is thus

$$\phi(B)\nabla^d \nabla_{144} \log(N_t) = \theta(B) Z_t, \quad d \in \{0, 1\}, \tag{4}$$

where $\{Z_t\} \sim \text{WN}(0, \sigma^2)$, $\phi(B) = 1 - \phi_1 B - \ldots - \phi_p B^p$ and $\theta(B) = 1 + \theta_1 B + \ldots + \theta_q B^q$. The difference order d is chosen from $\{0, 1\}$, depending on how fast the daily population decreases in trend.

Once the parameters of the above models are learned from training data, the prediction for N_{t+h} ($h > 0$), denoted $P_t N_{t+h}$, given the values $\{N_1, \ldots, N_t\}$ is performed as follows. To predict N_{t+h}, we first obtain $P_t \tilde{N}_{t+h}$, the minimum mean square error (MMSE) predictor for $\tilde{N}_{t+h}$. $P_t N_{t+h}$ is then obtained by retransforming $P_t \tilde{N}_{t+h}$ using the inverse of the corresponding operators ∇^d, ∇_{144} and $\log(\cdot)$.

As an example, we make one day-ahead (144-step) population prediction, *i.e.*, to predict each N_{t+144} based on $\{N_\tau; 0 \leq \tau \leq t\}$, in three channels, including a steady-state channel and two flash-crowd channels. The data of the first two days are chosen as the training data, *i.e.*, $\{N_t; t_1 < t \leq t_1 + 288\}$, where $t_1 \leq 72$ is to exclude the initial samples that may not comply with the later evolution pattern. Model (3) is fit to the steady-state channel, while model (4) is fit to the flash-crowd channels. The parameter estimates are obtained through a maximum likelihood estimator [4]. As shown in Fig. 2, with $p = 144$, $q = 0$, model (3) is able to tract the slight demand variations from day to day. For a channel with slowly decreasing daily population such as in Fig. 2(b), d is chosen to be 0 by the estimator, while for channels with faster daily population decrease in Fig. 2(c), $d = 1$.

B. Inferring Initial Population with a Mixture of Gaussians

We now infer the initial population evolution in a newly released video channel. Since no past observation is available yet, the inference of initial population cannot make use of seasonal ARIMA models as in Sec. III-A. However, the inference can be done based on the fact that videos released around the same time of day (possibly on different dates) demonstrate a similar population evolution pattern in the first several days. For example, Fig. 3 shows the initial populations of 3 different channels released around 7-9 PM on different dates. They exhibit a similar population evolution pattern: the first population peak was around the midnight after the videos were published, while the second peak was on the second day around noon.

For the $\{N_t\}$ series of each video v, we assume $t = 0$ upon its release. We group all the videos into a number of classes by their release times. We conjecture that the population $\{N_t; 0 \leq t \leq 144n\}$ in a video channel v for the first n days is a realization of the random process

$$N_t = P(v) \sum_{j=1}^{k} \frac{\pi_j}{\sqrt{2\pi\sigma_j^2}} \exp\left(-\frac{(t - \mu_j)^2}{2\sigma_j^2}\right) + Z_t, \tag{5}$$

where $\sum_{j=1}^{k} \pi_j = 1$, Z_t is a zero-mean random noise, $P(v)$ is a popularity index for video v, and

$$G_t := \sum_{j=1}^{k} \frac{\pi_j}{\sqrt{2\pi\sigma_j^2}} \exp\left(-\frac{(t - \mu_j)^2}{2\sigma_j^2}\right)$$

is the same pattern that underlies the population evolution of all the videos in the same class.

There is a probability rationale for this model based on how a user chooses the time $u(v)$ that she watches video v. Assume each user only watches the video once and her viewing span is negligible as compared to $144n$. Let μ_j, $j = 1, 2, \ldots, k$, represent the j-th peak time (usually at midnight or noon) after v is published. In reality, a user will watch v around time $\mu_1 < \mu_2 < \ldots$ with probabilities $\pi_1 > \pi_2 > \ldots$, since a video is more likely to attract audience when it is just published. Conditioned on that a user chooses to watch v around μ_j, $u(v)$ is normally distributed with density $\mathcal{N}(\mu_j, \sigma_j^2)$. Thus,

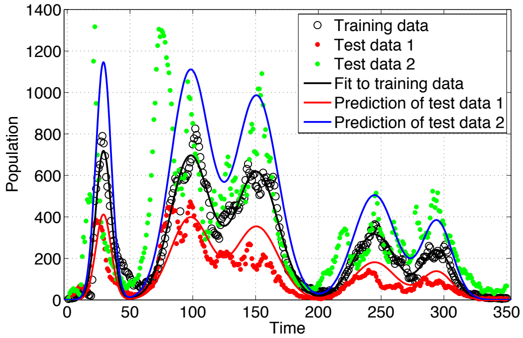


Fig. 3. Inferring initial population. Training data: channel 8331 released on 2008-08-10 19:46:29; test data 1: channel 57F7 released on 2008-08-18 20:16:35; test data 2: channel EDAF released on 2008-08-18 20:49:39. Prediction errors: RMSE$_1$ =74, RMSE$_2$ = 274.

we have

$$\Pr\left(t \le u(v) < t+1\right) \approx \sum_{j=1}^{k} \frac{\pi_j}{\sqrt{2\pi\sigma_j^2}} \exp\left(-\frac{(t-\mu_j)^2}{2\sigma_j^2}\right)\cdot 1,$$

Let $P(v)$ be the total number of potential users interested in video v. When $P(v)$ is large, we have N_t, the number of users that choose to watch video v at time t, given by (5).

In our experiments, we categorize all 32 newly released channels into 7 classes based on their release times. For each class, we train the mixture of Gaussians (5) with $k = 5$ using the EM algorithm [7] based on data $\{N_t; 0 < t \le 144n\}$ from the training channels. For each test channel in the same class, $\{N_t; l < t \le 144n\}$ are inferred with the trained model, given very limited initial observations $\{N_t; 0 \le t \le l\}$ of its own. ($\{N_t; 0 \le t \le l\}$ is used to estimate $P(v)$, *i.e.*, $P(v) = \sum_{t=1}^{l} N_t / \sum_{t=1}^{l} G_t$. We choose $n = 2.5$ and $l = 30$. A prediction example of two channels based on the model trained from only one channel is plotted in Fig. 3. We can see the proposed method achieves satisfactory performance as a best-effort coarse-grained predictor, with relatively small RMSEs compared to the population in the test set.

IV. Predicting Peer Contribution

In order to predict the server bandwidth S^r_{t+h} demanded by a channel at future time $t + h$, besides N_{t+h}, we need to estimate the bandwidth contributions c_{t+h} and u_{t+h} from cached peers and downloading peers. We find that in all the channels, $\{u_t\}$ is a stationary noisy process that can be modeled as a simple low order AR(p) process. In contrast, c_t is much harder to predict. In general, c_t increases from 0 to a relatively stationary value over time, as more downloading peers finish downloading and become cached peers. Since $\{c_t\}$ has a slowly increasing trend, we are tempted to model ∇c_t as an ARMA process. However, the model yields a linear predictor that miss-interprets a large part of the system dynamics as noise and fails to capture the slightly concave trend in c_t as t grows. Therefore, c_t is hardly predictable only based on its own history.

However, the prediction of $\{c_t\}$ is much more accurate if $\{N_t\}$ is used as a leading indicator. We transform both series $\{c_t\}$ and $\{N_t\}$ into stationary series $\{\tilde{c}_t\}$ and $\{\tilde{N}_t\}$ by logarithm transformation and differencing, so that $\{\tilde{c}_t\}$ and $\{\tilde{N}_t\}$ can be linked together through a linear time-invariant (LTI) system of the form

$$\delta(B)\tilde{c}_t = \omega(B)\tilde{N}_t + Z_t, \tag{6}$$

where $\delta(B) = 1 - \delta_1 B - \ldots - \delta_p B^p$, $\omega(B) = \omega_0 + \omega_1 B + \ldots + \omega_q B^q$, and Z_t is white noise. We thus obtain a transfer function model [4] for $\{c_t\}$ and $\{N_t\}$:

$$\begin{cases} \tilde{c}_t = \delta^{-1}(B)\omega(B)\tilde{N}_t + \delta(B)^{-1}Z_t, \\ \tilde{N}_t = \nabla^{d_1}\nabla_{144}\log(N_t) - \mu_1, \quad d_1 \in \{0,1\}, \\ \tilde{c}_t = \nabla^{d_2}\nabla_{144}\log(c_t) - \mu_2, \quad d_2 \in \{0,1\}, \end{cases} \tag{7}$$

where $\{Z_t\} \sim \mathrm{WN}(0, \sigma^2)$, and μ_1 and μ_2 are the means of $\nabla^{d_1}\nabla_{144}\log(N_t)$ and $\nabla^{d_2}\nabla_{144}\log(c_t)$, respectively. By appropriately choosing $d_1, d_2 \in \{0,1\}$, for any flash-crowd channel, the relationship between $\{c_t\}$ and $\{N_t\}$ can be characterized by model (7). Given training data, the maximum likelihood estimates of δ's and ω's in (7) can be obtained by minimizing the conditional sum-of-squares function [4], which is standard in system identification theory.

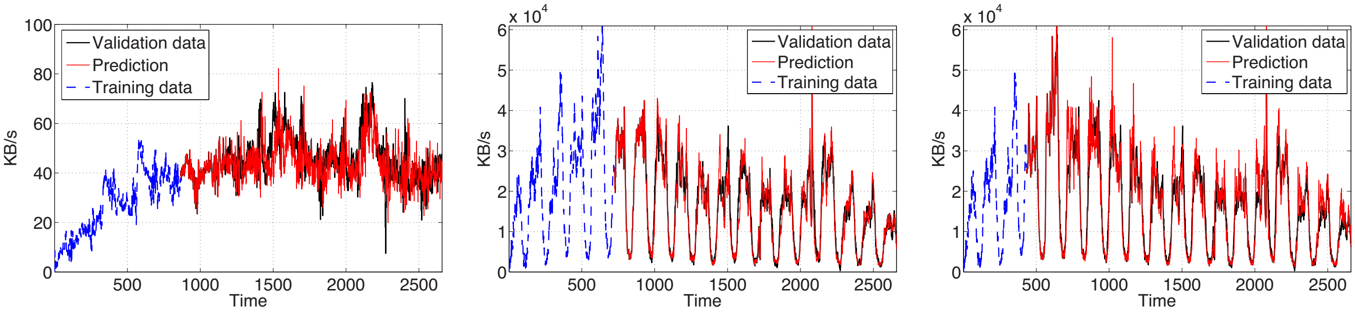
We illustrate the prediction procedure with a typical flash-crowd channel A55F. We choose $d_1 = d_2 = 0$, $p = 54$, $q = 0$, and learn the model parameters using the data of the first few days. For a lead time h, each c_t in the validation data is predicted based on $\{c_{t-h}, c_{t-h-1}, \ldots\}$ and $\{N_{t-h}, N_{t-h-1}, \ldots\}$. From Fig. 4(a), we see that a model trained from the first 6 days' data yields very accurate 2.5 hour-ahead (15-step) prediction for c_t.

We also evaluate the h-step prediction for the total receiving rate from cached peers $c_t N_t$ shown in Fig. 4(b) and Fig. 4(c). $c_t N_t$ is predicted as $\hat{c}_t \hat{N}_t$, where the predictions $\hat{c}_t$ and $\hat{N}_t$ are made using models (7) and (4), respectively, based on observations up to time $t - h$. Fig. 4(b) shows that the proposed method can predict the total receiving rate from cached peers accurately for 2.5 hours into the future. An interesting phenomenon is that although c_t is increasing over time, the peak of $c_t N_t$ is around the 5th day due to a decreasing number of N_t. The proposed method is able to predict such a peak in $c_t N_t$ one hour ahead of time, simply with a training period of the first 3 days, as shown in Fig. 4(c), although with lower accuracy as compared to Fig. 4(b).

V. Demand Forecast

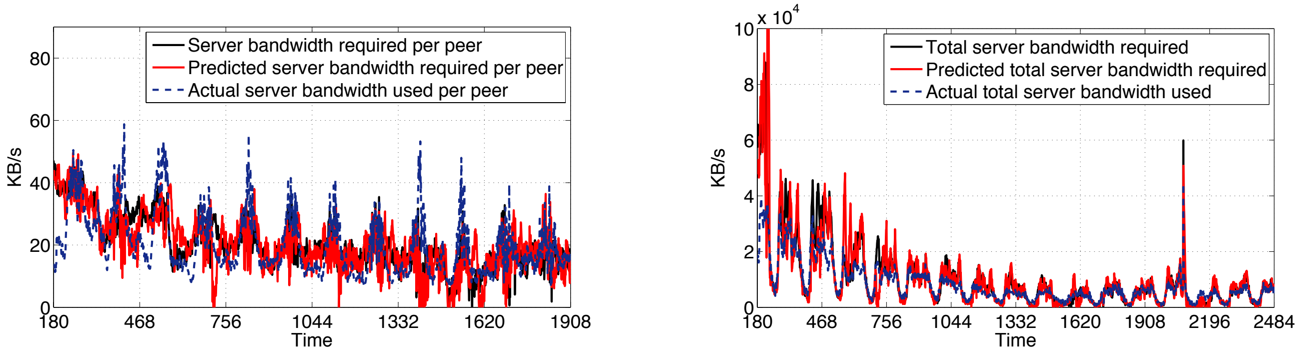
Having elaborated the algorithms for predicting $\{N_t\}$, $\{u_t\}$ and $\{c_t\}$, we are ready to solve the server bandwidth demand forecast problem outlined in Sec. II. Recall that given observations up to time t, the total server bandwidth S^r_{t+h} required by a channel to achieve an $r_t \ge R$ is given by (2). To facilitate application in practical systems, we propose an iterative procedure for progressive model learning and demand prediction, with the aid of online measurement collection.

We illustrate such a procedure in a typical flash-crowd channel A55F. After the release of the channel, we first train the models (4) and (7) for $\{N_t\}$ and $\{c_t\}$ and the autoregressive model for $\{u_t\}$ using the data of the first 1.25



(a) c_t, with a 6-day training period, $h = 15$ (b) $c_t N_t$, with a 5-day training period, $h = 15$ (c) $c_t N_t$, with a 3-day training period, $h = 6$

Fig. 4. The h-step prediction of average receiving rate c_t and total receiving rate $c_t N_t$ from cached peers in channel A55F. $d_1 = d_2 = 0$, $p = 54$, $q = 0$.



(a) Prediction for the server bandwidth required per peer (b) Prediction for the total server bandwidth required by the channel

Fig. 5. Progressive model learning on a daily basis and 2.5 hours ahead (15-step) prediction for the server bandwidth demanded by channel A55F over 16 days. The channel is released on 2008-08-10 10:47:39, and the prediction starts from 2008-08-11 16:47:39 (time period 181).

days (180 time periods), assuming an AR(p) for (4) and (7) with $p = 10$. The models are retrained every 24 hours during the system operation, with a progressively increased model order, based on the measurements collected so far, *i.e.*, p is increased by 10 everyday. The prediction starts at time period 181 and is always made with the newest trained model.

We make 2.5 hours ahead (h=15) prediction for the server bandwidth S^r_{t+h} demanded by the channel and plot the results in Fig. 5. The target mean downloading rate is $R = 80$ KB/s, which can be obtained empirically. We can see that the predicted server bandwidth demand $\hat{S}^r_{t+h}$ is quite close to the real demand S^r_{t+h} in the traces. We also see that the actual server bandwidth used by the channel in the traces is often less than the required minimum amount, especially in the first 3 days. This accounts for the performance issues after the channel is just released, as observed in Fig. 1. Incorporating our prediction mechanism into the system would have forecasted the server bandwidth demand 2.5 hours ahead of time and advised proper server provisioning actions.

In addition, Fig. 5(b) shows that the mechanism is able to predict an abnormal surge in the bandwidth demand at time 2075 because the seasonal ARIMA model (4) accurately predicts a surge in N_t at this point. Moreover, the entire simulation of the above procedure takes less than 90 seconds on a Macbook Pro with 2.26 GHz duo-core processor, indicating reasonable computational cost.

VI. Concluding Remarks

In this paper, we address the issues of demand forecast and performance prediction in peer-assisted VoD services. Through mining the data collected from UUSee Inc., we observe clear patterns in the evolution of online peer population, and propose to predict online population in both existing and new channels making use of the viewing behavior of Internet users. Furthermore, we propose a novel scheme to forecast the instantaneous peer upload contribution with the population evolution as a leading indicator. The prediction of these key statistics in the network enables an accurate forecast of the server bandwidth demand in each channel, which is useful towards efficient and proactive capacity planning and quality control. We also shed lights on how prediction and forecasting can be incorporated in real systems at the runtime with the aid of online monitoring. The effectiveness of our methods is corroborated by operational traces collected from UUSee Inc.

References

[1] X. Zhang, J. Liu, B. Li, and T.-S. P. Yum, "CoolStreaming/DONet: A Data-Driven Overlay Network for Efficient Live Media Streaming," in *Proc. of IEEE INFOCOM*, 2005.

[2] Y. Huang, T. Z. J. Fu, D.-M. Chiu, J. C. Lui, and C. Huang, "Challenges, Design and Analysis of a Large-scale P2P-VoD System," in *Proc. of SIGCOMM'08*, Seattle, Washington, August 2008.

[3] Z. Liu, C. Wu, B. Li, and S. Zhao, "UUSee: Large-Scale Operational On-Demand Streaming with Random Network Coding," in *Proc. of IEEE INFOCOM '10*, San Diego, California, March 2010.

[4] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel, *Time Series Analysis: Forecasting and Control*. WILEY, 2008.

[5] D. Applegate, A. Archer, V. G. S. Lee, and K. Ramakrishnan, "Optimal Content Placement for a Large-Scale VoD System," in *Proc. of ACM CoNEXT*, Philadelphia, USA, November, 2010.

[6] Y. Koren, "Factorization Meets the Neighborhood: a Multifaceted Collaborative Filtering Model," in *Proc. of KDD*, Las Vegas, Nevada, August, 2008.

[7] C. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.