Throughput Prediction-Based Rate Adaptation for Real-Time Video Streaming over UAVs Networks

Tongqing Zhou, Haidong Zhang, Ming Xu, and Yingwen Chen

College of Computer, National University of Defense Technology, China
{zhoutongqing,zhanghaidong,xuming}@nudt.edu.cn, csywchen@gmail.com

Abstract. Real-time video streaming is extensively used in UAVs networks for battlefield surveillance, disaster relief, etc. The available throughput of the multi-hop networks varies a lot with the movement of UAVs. To guarantee the video's quality and continuity, rate adaptation mechanism should be used to choose the appropriate transmission rate according to the varying throughput. In this paper, we propose a novel proactive prediction-based adaptation algorithm to avoid disruptions and provide high quality for real-time streaming over UAVs networks. We show that available throughput varies periodically with UAVs' mission-related movement. Then we set a prediction range with the knowledge of periodicity gained from the measurements of a training. The raw prediction is further calibrated with reactive estimation of buffered video time to precisely guide the adaptation. Simulation results show that our scheme maintains a continuous playback with a high quality and significantly shorten the start-up delay compared with two constant bit-rate schemes.

Keywords: UAVs networks, knowledge-based prediction, rate adaptation.

1 Introduction

Unmanned Aerial Vehicles(UAVs) networks have been playing increasingly important roles in many areas such as disaster relief, wide area sensing and battlefield surveillance [1]. In these scenarios, UAV nodes act as either information collectors or relays, communicating with each other jointly to deliver the critical data to home station with cost efficiency.

Real-time video streaming is especially valuable among all the data collected, because it provides users with explicit and intuitive descriptions of the area they care for. For example, in tactical networks, one can envision the captured video to facilitate mission management. Nevertheless, the stringent end-to-end latency requirement of real-time video is hard to guarantee. In practical, either video continuity or video quality needs to back off, since the wireless connections among UAV nodes are bandlimited and time-varying, considering the inevitable topology changes over time. As video interruptions degrade user experience significantly, a key challenge rises that how we could get rid of video freezes and shorten start-up delay while still guaranteeing high video quality.

Z. Cai et al. (Eds.): WASA 2014, LNCS 8491, pp. 78–89, 2014.

To fulfill this purpose, the bit-rate of video streaming should adapt to the variation of wireless link quality, which demands for the guidance under throughput prediction. Existing works typically perform prediction based on reactive measurements, e.g. Round Trip Time(RTT) [2], Segment Fetch Time(SFT) [3] and simply realtime throughput [4]. For example, [4] predicts the available throughput[1] according to the average throughput achieved in the past several video segments. Though it does make sense when large TCP traffics steadily occupy the same path in a stable environment [5], things are quite different in real-time video streaming scenarios where the achieved throughput is often capped by video generating rate, so that it cannot predict available throughput explicitly. Hence, available bandwidth could be underestimated, which is likely to mislead bit-rate adaption, thus achieving limited video quality. Moreover, the existing schemes are ill-suited to the frequently variation of multi-hop wireless links that UAV works on.

In this paper, we propose a novel scheme to perform video rate adaptation for real-time streaming among UAVs, which enables high utilization of available throughput and makes use of the periodicity knowledge to overcome the frequently throughput variation. Our scheme involves two components, knowledge-based prediction and adaptation with calibration, carried out separately in a training stage and a transmission stage. With a same reconnaissance route, we show that the two stages experience a similar periodic variation, thus making a proactive prediction possible. Our major contributions are:

1. We introduce a proactive training to analyze the variation of available throughput caused by mission-related movement and conclude its periodicity. A prediction range is set based on the several periods' measurements in the training.
2. We propose a method for the estimation of buffered video time, generating an calibration parameter to perform a reactive tweak on the raw prediction.
3. We design an adaptation algorithm based on the prediction to smoothly match the available throughput.

To the best of our knowledge, our prediction method is the first to adopt the periodic throughput variation of UAVs networks to guide video rate adaptation. Moreover, the traditional buffered video time estimation is enhanced in our adaptation algorithm. Experiments show that our scheme successfully maintains an uninterrupted playback with a much shorter start-up delay than two constant bit-rate(CBR) schemes, and achieves a higher video quality.

The rest of the paper is organized as followed: Section 2 describes the relevant work. Knowledge-based throughput prediction and video rate adaptation algorithm are described in Section 3 and Section 4 respectively. Simulation results based on ns3 and discussion are presented in Section 5. We conclude the paper with summary in Section 6.

[1] Throughout the paper we use available throughput to refer to the maximum rate a given flow would achieve at a given point in time.

2 Related Work

Throughput prediction is vital for video rate adaptation as an essential guidance. In [5], throughput predictors for a broad class of applications are analyzed and summarized. And they classify TCP throughput prediction techniques into two categories: Formula-Based (FB) and History-Based (HB). These methods are extensively used in adaptation algorithm. In [2], [3] and [8], FB prediction is presented by analyzing measured parameter RTT, SFT and loss rate. In [4] and [6], HB prediction is proposed to detect fluctuation and take adaptation in real-time. We argue that both the two categories can be regarded as reactive prediction, because the characteristics demanded by FB and previous information demanded by HB are all obtained in real-time. Reactive throughput prediction is often capped by the real-time video's output rate. Thus the capacity would be underestimated which misleads the bit-rate adaptation to a lower level than expected, in other words, video could be transmitted in a higher quality. Our prediction scheme is performed based on the analysis of UAVs' mission-related movement in a training stage before the real-time transmission, which avoids the influence of video output on throughput prediction.

Haakon et al implement a location-based bandwidth-lookup service for bit-rate planning in [7]. They measured bandwidth in real environment, and plan quality adaptation based on the measurement and intermittently collected GPS positional data. In [9], authors propose a location-based model to predict the performance of TCP over a varying ground-to-UAV wireless link. Despite the attractiveness, the estimation process calls for frequently interaction between base station and mobile node which brings down the good throughput, and impacts the real-time transmission, especially for the multi-hop transmission in UAVs networks.

3 Knowledge-Based throughput Prediction

In UAVs networks, a group of mobile nodes jointly communicate with each other to extend communication range. In practical applications, the UAVs play either the role of collector or relay to fulfill a mission. Typically, the collector node would be scheduled to move in a predefined route to obtain acquired data from the ground. The requisite movement result in significantly variation of the available throughput. Ideally, the collector would know the throughput's behavior in advance to take corresponding adaptation for video bit-rate.

Fig. 1 presents a scenario of earthquake surveillance, in which three UAVs facilitate a wireless chain network to transmit the collected real-time video back to the home station. We find that the variation of throughput is periodic with UAVs' repeatedly surveillant movement. Hence, the knowledge of periodicity can be used to benefit the prediction of throughput. A proactive training stage is added before transmission starts to observe and analyze the variation of available throughput in a multi-hop UAV link with the collector regularly moves. Note that the available throughput would experience a similar periodic variation during

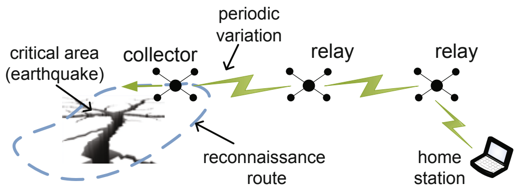

Fig. 1. The collector UAV repeatedly moves in the reconnaissance route to collect real-time information from the critical area. The repeatedly surveillance behavior results in periodic variation of the available throughput provided by the 3-hop-chain network.

video transmission. We introduce a continuous TCP bulk flow from the collector to home station to monitor the variation of available throughput.

We use time series analysis method to prove the existing and conclude the variation period. Let $TP = \{tp_i | 1 \leq i \leq n\}$ be the measured throughput during the t_m long stage, and the sampling interval of throughput is I which equals to t_m/n. Interval I should be appropriately chosen to enable a timely adaptation. The variation frequencies of throughput is organized as

$$F = \{f_i | 0 < f_i \leq 1/I, 1 < i \leq n\} \tag{1}$$

where element f_i denotes a variation frequency. Range of f_i and i decide that the variation period is larger than I and no bigger than t_m. Using Fourier transform, the intensity of each frequency component can be denoted as $P_i = S_i^2 + C_i^2$, where

$$S_i = \sum_{j=1}^{n} (tp_j - tp_m) \cdot \sin(2\pi \cdot k_j \cdot f_i), \tag{2}$$

and

$$C_i = \sum_{j=1}^{n} (tp_j - tp_m) \cdot \cos(2\pi \cdot k_j \cdot f_i) \tag{3}$$

tp_m denotes the mean value of the measured throughput, and $k_j = j$. Theoretically, P_i reflects the probability that throughput varies in frequency f_i. Therefore, frequency f_p related to the peek intensity value P_{max} presents the maximum probability for measured throughput to vary with.

The period of throughput's variation, denoted as T, divides the training stage into $N = \left\lfloor {}^{t_m}\!/_{T} \right\rfloor$ periods, where T equals to $1/f_p$. Hence, there would be T/I intervals in one period. Based on the division, we calculate the average throughput and minimum throughput for each interval in one period. We have,

$$TP_{avg} = \{tp_i^{avg} | tp_i^{avg} = \frac{\sum_{j=1}^{N} tp_{i+T\cdot j}}{N}, i \in [1, T/I)\}, \tag{4}$$

and

$$TP_{\min} = \{tp_i^{\min} | tp_i^{\min} = \min_{0 \leq j \leq N-1} \{tp_{i+T\cdot j}\}, i \in [1, T/I)\} \tag{5}$$

tp_i^{avg} and tp_i^{min} refer to the average and minimum throughput for the i-th interval. During a real-time video transmission, we gain on the average results to seek high video quality, and gain on the minimum ones to avoid the influence of unwanted fluctuations. Hence, TP_{avg} and TP_{min} act as the upper bound and lower bound of the prediction for one period, respectively.

4 Rate Adaptation Algorithm

In this section, an effective video rate adaptation algorithm is proposed in which we adapt the bit-rate to appropriate levels based on prediction repeatedly after a specific interval. By doing this, we aim to (i) minimize the start-up delay while keeping a continuous video playback and (ii) maximize the quality of video. Meanwhile, we estimate the buffered video time on sender-side to tweak the raw prediction results of the training. The flowchart of our adaptation algorithm is shown in Fig. 2. In the flowchart, we use t_{ad} to represent the adaptation interval, while tp_i and tp_{i+1} denote the predicted throughput for the current and next interval respectively. It is important to note that t_{ad} should be appropriately chosen to provide timely adaptation and a relatively smooth quality switch is maintained by taking multi-subsequent-predictions into account at one time. The adaptation process is explicitly described in the subsequent sections.

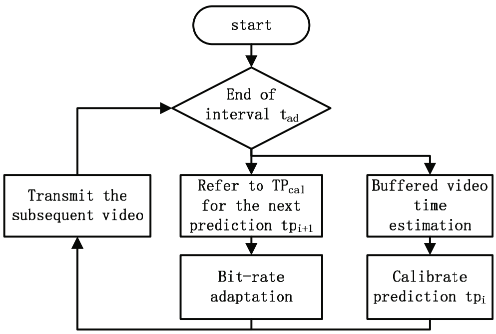

Fig. 2. Flowchart of the proposed prediction-based adaptation algorithm

4.1 Buffered Video Time Estimation

As shown in Fig. 2, we estimate the buffered video time at the end of each adaptation interval t_{ad}. The estimation acts as an evaluation for the prediction performance of the previous interval, and will be provided to the calibration step as a control parameter. Meanwhile, buffer level(buffered video time) is also used to indicate whether the playback is continuous or sometimes interrupted.

We propose a novel scheme to estimate the buffered video time on the sender side instead of collecting the measurement from the receiver side. Overhead on fetching feed-back information is avoided by doing so. During the continuous transmission, we record the sent bytes M_i^s for the i-th sending interval of a period, and calculate the received bytes of receiver M_i^r based on the sequences of TCP ack segments. The value of M_i^s and M_i^r have no direct relationship with each other as throughput varies. If the sending bit-rate is higher than the current throughput, some resident bytes which represents the bytes un-received will generate. We denote this part of data as M_{res}^i. And the value of M_{res}^i decreases as the adaptation algorithm switch the bit-rate down below the available throughput, then we have,

$$M_i^{res} = \begin{cases} M_i^s - M_i^r & i = 1 \\ M_{i-1}^{res} - (M_i^r - M_i^s) & i > 1 \end{cases} \tag{6}$$

At the end of interval i, say $t_i^{(e)}$, the sent bytes is no smaller than the received bytes, so the resident bytes satisfy $M_i^{res} \geq 0$.

Assume that t_i^r is the length of video received during adaptation interval t_{ad}. t_i^r is consist of previously and currently received video pieces which have different bit-rate levels, thus we have,

$$t_{i+1}^r = \frac{M_i^{res}}{br_i} + \frac{M_{i+1}^r - M_i^{res}}{br_{i+1}} \tag{7}$$

where br_i is the selected video bit-rate during interval i. Then, the reference buffered time can be estimated as,

$$t_{i+1}^{buf} = t_i^{buf} - (t_{ad} - t_{i+1}^r) = t_{su} - \sum_{j=1}^{i+1} (t_{ad} - t_j^r) \tag{8}$$

where t_{su} denotes the start-up accumulation(delay), and t_i^{buf} indicates the buffer level at $t_i^{(e)}$. We emphasize that t_{su} is needed for that t_i^r happens to be smaller than the consumed length of time during the transmission.

4.2 Throughput Prediction Calibration

Results of the proactive training set the variation range of throughput prediction as $[TP_{avg}, TP_{min}]$. If we simply take TP_{avg} as the available throughput, a high video quality could be achieved while video freezes would occur. On the contrary, if we carefully adapt the bit-rate based on TP_{min}, no interruption occurs but the video quality stays low. Taking a compromise between the two bounds, we dynamically estimate the available throughput by introducing the reference buffer level as a control parameter, and derive the calibrated prediction $TP_{cal} = \{tp_i^{cal} | 1 \leq i \leq T\}$.

Specifically, we propose a weighted average method for the calibration,

$$tp_i^{cal} = \alpha_i \cdot tp_i^{avg} + (1 - \alpha_i) \cdot tp_i^{\min} \tag{9}$$

where tp_i^{cal} is the prediction for the available throughput during the i-th interval in a period, and α_i denotes the related weight. By tweaking parameter α_i, the value of tp_i^{cal} is dynamically changed in range $[tp_i^{\min}, tp_i^{avg}]$. After an adaptation interval, we increase α_i with an increment Δ_i within the given range unless the buffered time's reduction during that interval goes beyond a threshold level t_f. Then we decrease α_i with Δ_i. The increment Δ_i related to α_i equals to $W/(tp_i^{avg} - tp_i^{\min})$, where W is assumed to be the gap between two bit-rate levels. Thus, every adjustment of α_i is sufficient for a level switch of bit-rate.

Hence, the available throughput can be predicted as a succession of throughput period, in which the prediction performance of last period directly influence current prediction TP_{cal}. The advantage of dynamically calibrating the prediction compared to simply using the average or the minimum version is that it tweaks the proactive prediction with reactive information to adapt to the actual situation.

4.3 Bit-Rate Adaptation

Bit-rate adaptation process is performed based on the calibrated throughput prediction. After each adaptation interval, we refer to TP_{cal} to analyze the relationship between current bit-rate and prediction value for the subsequent interval, which can be categorized into three situations: 1) $tp_{i+1}^{cal} > br_i + W$, 2) $br_i < tp_{i+1}^{cal} < br_i + W$ and 3) $tp_{i+1}^{cal} < br_i$.

We set $br_{i+1} = br_i$ in situation 2) because br_i remains to be the best appropriate level. In situation 1), switch-up operation is expected to promote a high quality. We propose a selective switch-up scheme to adapt the bit-rate, in which the predicted available throughput for the next $r(r \geq 2)$ intervals are all picked for comparison. The scheme avoids instantaneously switch-up by detecting the probably throughput spikes. Thus, a switch-up is adopted only if $tp_{i+2}^{cal},...,tp_{i+r}^{cal}$ also satisfies situation 1). And the corresponding bit-rate is set as

$$br_{i+1} = \max\{br_j^{level}, br_j^{level} \leq \min\{tp_{i+1}^{cal}, ..., tp_{i+r}^{cal}\}\} \tag{10}$$

where br_j^{level} refers to an accomplishable bit-rate level, and note that tp_{i+1}^{cal} would be updated at the end of the subsequent adaptation interval. Otherwise, bit-rate would stay unchanged for the next interval to maintain adaptation stability. For situation 3), switch-down operation should take place to avoid a buffer reduction which may lead to video interruption. In this situation, an aggressive switch-down is preferred to set the bit-rate level with equation 10. The selective process is maintained by taking longer prediction into account to benefit smooth adaptation, while the aggressive action is adopted based on the next interval's prediction. We point out that the proposed algorithm avoids adaptation caused delay by cautiously adapting to the prediction. Thus we only need to calibrate the prediction when unexpected delay is detected.

The minimum throughput version in period T reflects the available throughput related to a specific situation, in which some random interference or loss happen to occur. The algorithm starts by selecting tp_1^{min} as the prediction for available throughput to minimum start-up delay, calibration coefficients are therefore set as 0. For the subsequent periods, we perform selective switch-up and aggressive switch-down as mentioned above to maximize the quality while maintaining a continuous play.

5 Experimental Evaluation

5.1 Experiment Environment

We implemented our knowledge-based prediction scheme and rate adaptation algorithm in ns3 [10]. A chain topology with 3 nodes was used in the simulations. The nodes act as the collector UAV, the relay UAV and the home station, respectively. They fell into a line at the beginning of the simulation, and the initial distance between any two adjacent nodes was set to $100m$. During the mission, the collector moves in a predefined reconnaissance route repeatedly to collect acquired data from assumptive critical area. We used a log distance propagation model to simulate the wireless channel. All the nodes were configured to utilize IEEE 802.11g physical layer.

We performed the simulation process in two stages, that is, a training stage and a video transmission stage. The two stages last for $200s$ and $1000s$, respectively. The BulkSender application in ns3 was adopted to generate a continuous TCP flow during the training stage, in which we analyzed the periodicity of variation and calculated two prediction bounds. Meanwhile, a real-time video streaming application was designed to carry out during the video transmission stage, in which our algorithm was evaluated and two CBR schemes were also performed for comparison.

In the simulation, we set both the sampling interval I and the adaptation interval t_{ad} to $2s$, and the start-up time accumulation t_{su} was set to $5s$. Note that t_{ad} was set to its minimum value to enable a timely adaptation. And the stability is maintained by the adaptation scheme with smooth parameter r set to 2. Calibration coefficient α_i for each interval in a period was initially set to 0. And we denoted the reduction threshold t_f to $0.3s$. Finally, the adopted bit-rates were 500, 550, ... , and $1000kbit/s$ with the level gap W set as 50.

5.2 Result and Discussion

Firstly, we carried out a simulation to evaluate the efficiency of the proposed available throughput prediction scheme. Based on the measurements in proactive training, the prediction scheme firstly analyzes the periodicity, the results is shown in Fig. 3. As mentioned above, the intensity for frequency f_i reflects the probability that throughput varies with f_i. Then we have $T = (1/f_4) \cdot I = 50s$, where f_4 is the frequency related to the maximum intensity. Hence, we can divide the training stage into $200/T$ periods. For each interval in a period, we calculate both its average value and minimum value of all the periods. Fig. 4 shows the throughput prediction results TP_{avg} and TP_{min}, which set the upper and lower bound for latter calibration.

Fig. 5 shows the calibrated throughput prediction with a comparison to the available throughput. During a period, the predicted throughput at a given time

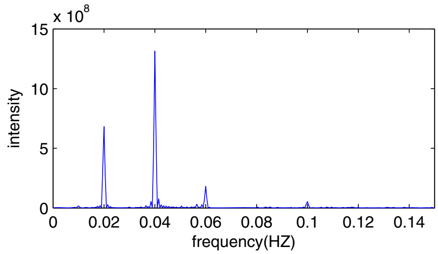

Fig. 3. Intensity as a function of throughput variation frequency

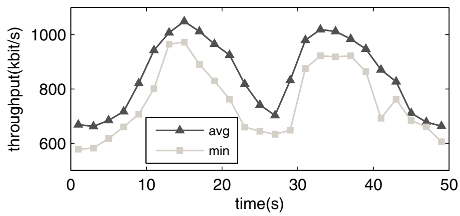

Fig. 4. The prediction bound TP_{avg} and TP_{min} in one period

point i is calibrated to a higher value each time tp_i^{cal} successes to cover the actual capacity. And it decreases when $tp_i^{cal} > tp_i^a$ is satisfied, thus causing the reduction of buffered time to go beyond the threshold t_f. Specifically, the calibration process is performed based on coefficient α, which is periodically tweaked with the buffered time. We take α_{12} and α_{21} related to the 12-th and 21-th interval of the period as examples to present the adjustment of α, the results are shown in Fig. 6. Their increments Δ_{12} and Δ_{21} are 0.3 and 0.2 respectively. We point out that TP_{cal} successfully fits the available throughput by dynamically tweaking between the two prediction bounds TP_{avg} and TP_{min}.

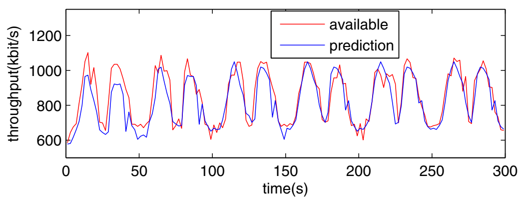

Fig. 5. Calibrated prediction TP_{cal} and available throughput TP_a

Fig. 7 shows the adaptation results of our proposed algorithm, where we only present the first 300s of bit-rate adaptation. During the transmission, adaptation algorithm dynamically choose a bit-rate level below the upper limit set by

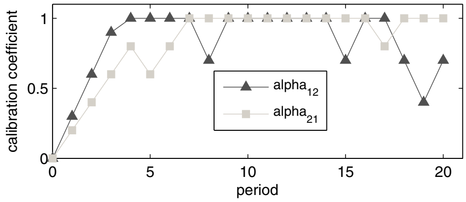

Fig. 6. Adjustment of coefficient α_{12} and α_{21} during the 20 periods long transmission

TP_{cal}. Bit-rate is cautiously switched up to achieve high video quality while maintaining adequate stability by taking next 2 intervals' prediction into account. And it is promptly switched down to avoid adaptation caused delay. The reference buffer level fluctuates as mismatch between available throughput and bit-rate occurs. A behavior of reduction indicates that video bit-rate exceeds the current throughput, which causes the length of received video less than the transmission time. On the other hand, buffer level raises or stays still when the link capacity successfully covers the current bit-rate, thus the new generated bytes and the residual bytes are both received to make up the time consumption. We remark that no buffer underflow occurs, and the minimum buffered time is around $2.5s$.

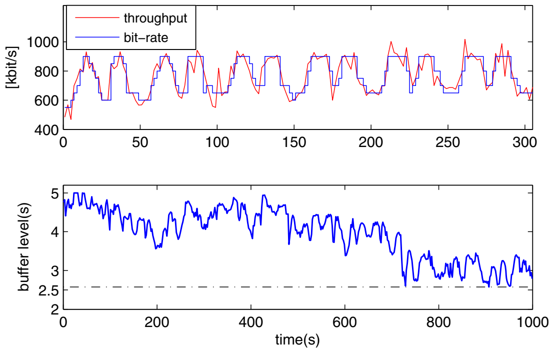

Fig. 7. Measured throughput, video bit-rate and buffer level

Fig. 8 and Fig. 9 show the observation of buffer level for two other runs, which are a high CBR scheme and a low CBR scheme. We set the bit-rate $750kbit/s$ for the former and $650kbit/s$ for the latter, introducing a span of 2 bit-rate levels. Note that both the actual buffer level and the relative buffer level are presented in Fig. 8. The relative level is used here to show the length of surplus video time when the buffer level is above 0, and it reflects the length of video that have suffered a disruption when the buffer exhausts. We point out that

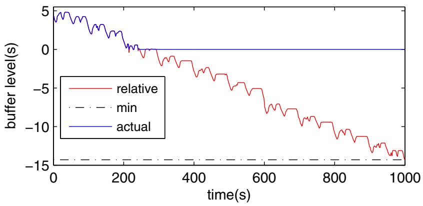

Fig. 8. Buffer level, cbr=750kbit/s

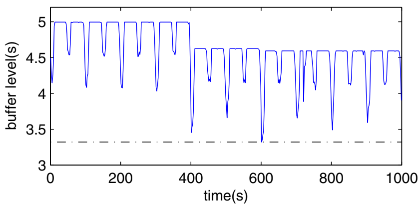

Fig. 9. Buffer level, cbr=650kbit/s

buffer underflows happen because it is unable for an CBR scheme to forecast any variations of available throughput.

Table 1. Average bit-rate and relative delay in 3 schemes

Scheme	Average bit-rate(kbit/s)	Relative delay(s)
1	789	2.5
2	750	19.5
3	650	1.7

The detailed comparisons between our adaptation algorithm and the CBR schemes are presented in Table 1, wherein both the average bit-rate and relative delay are concluded, and scheme 1, 2, 3 refer to the proposed algorithm, the high CBR scheme and the low CBR scheme, respectively. Average bit-rate is used because it represents the average quality of the video streaming. And we use relative delay to represent a specific start-up delay level with which no interruption would happen during the transmission. It is important to note that relative delay fairly reflects one's performance on real-time. As a conversion between start-up accumulation and minimum buffer level, relative delay can be calculated as $t_{su} - t_{min}^{buf}$, where t_{min}^{buf} represents the minimum buffer level experienced(e.g., $2.53s$ for scheme 1). Intuitively, the proposed algorithm achieves a relative delay of $2.5s$, almost 8 times shorter than what scheme 2 exhibits, and promotes an even higher average bit-rate. Compared to scheme 3, scheme 1 brings a rise in bit-rate of almost 3 levels, while maintaining a basically same relative delay (only 0.8s larger). As to continuity, neither the proposed algorithm nor the low CBR scheme incurs an underflow, while the high CBR scheme suffers continuous underflow after around 200s. The proposed algorithm successfully maintains a continuous play of real-time video with a high quality, while introducing the minimum start-up delay.

6 Conclusion

In this paper, we propose a novel throughput prediction method for video bitrate adaptation performed during real-time video streaming. The advantage of the proposed knowledge-based prediction method compared to schemes used in existing rate adaptation works is that our method is carried out during a proactive training to get rid of the video generating rate's throttle on reactive measurements. And we dynamically calibrate the prediction results with the reactive control of sender-side buffer level estimation. Moreover, an aggressive switch-down and selective switch-up scheme is used to avoid interruption, promote video quality and smooth the adaptation. Simulation results show that the proposed algorithm efficiently maintains a continuous play of real-time video with a minimum start-up delay and provides a relatively higher quality.

References

1. Zhao, W., Ammar, M., Zegura, E.: A message ferrying approach for data delivery in sparse mobile ad hoc networks. In: Proc. of ACM International Symposium on Mobile Ad hoc Networking and Computing (2004)
2. Thapliya, R., Hu, C.: AdapComm: a bandwidth allocation methodology for multimedia applications in wireless networks. In: Proc. of ACM SIGCOMM (2013)
3. Liu, C., Bouazizi, I., Gabbouj, M.: Rate adaptation for adaptive HTTP streaming. In: Proc. of the Second Annual ACM Conference on Multimedia Systems (2011)
4. Tian, G., Liu, Y.: Towards agile and smooth video adaptation in dynamic HTTP streaming. In: Proc. of the 8th International Conference on Emerging Networking Experiments and Technologies (2012)
5. He, Q., Dovrolis, C., Ammar, M.: On the predictability of large transfer TCP throughput. In: Proc. of ACM SIGCOMM (2005)
6. Miller, K., Quacchio, E., Gennari, G., et al.: Adaptation algorithm for adaptive streaming over HTTP. In: Proc. of IEEE Packet Video Workshop, PV (2012)
7. Riiser, H., Endestad, T., Vigmostad, P., et al.: Video streaming using a location-based bandwidth-lookup service for bitrate planning. ACM Transactions on Multimedia Computing, Communications, and Applications, TOMCCAP (2012)
8. Wu, D., Hou, Y.T., Zhu, W., et al.: On end-to-end architecture for transporting MPEG-4 video over the Internet. IEEE Transactions on Circuits and Systems for Video Technology (2000)
9. Kung, H.T., Lin, C.K., Lin, T.H., et al.: A location-dependent runs-and-gaps model for predicting TCP performance over a UAV wireless channel. In: Proc. of IEEE MILCOM (2010)
10. Network simulator ns-3, `http://www.nsnam.org/`