Asynchronous Source Clock Frequency Recovery through Aperiodic Packet Streams

Kyeong Soo Kim, *Member, IEEE*

Abstract—We consider the most general case of source clock frequency recovery (SCFR) in packet networks, i.e., asynchronous SCFR through aperiodic packet streams, where there is neither a common reference clock nor any relation between packet generation intervals and a source clock frequency. We formulate the problem of asynchronous SCFR with timestamps as a linear case of regression through the origin (RTO) and propose two schemes, one based on recursive least squares (RLS) method and the other based on simple heuristics of cumulative ratio of interarrival and interdeparture times, which provide better estimates of the source clock frequency with faster convergence than conventional phase-locked loop (PLL)-based schemes.

Index Terms—Source clock frequency recovery, linear regression, least squares, cumulative ratio, packet jitter.

I. INTRODUCTION

REAL-TIME variable bit rate (VBR) services carried over asynchronous packet networks impose serious challenges on the source clock frequency recovery (SCFR) at a receiver. First, there is no common reference clock available at a source and a receiver, with which the source clock frequency can be encoded and transmitted to the receiver for its recovery there as in the synchronous residual time stamp (SRTS) method [1]. Secondly, unlike constant bit rate (CBR) services, packet generation intervals (i.e., interdeparture times) at the source are varying over time and have no relation with the source clock frequency. These, together with unpredictable packet jitter and frequent packet losses, make the asynchronous SCFR quite a difficult task.

A periodic packet stream is a stream of packets generated and transmitted at regular intervals (i.e., constant interdeparture times) at a source, which is typically related with a source clock frequency, while an aperiodic packet stream has no such regular interval. Note that CBR services can result in aperiodic packet streams when the packet size is not fixed. On the other hand, the components of VBR traffic can be periodic like the first packet of each encoded video frame. So we use the term *aperiodic packet streams* in order to clarify the major focus of the asynchronous SCFR schemes studied in this paper. Note that, when there is no periodic time structure in generated packet streams (e.g., voice over IP (VoIP)), these services are more tolerable to the playout delay. The SCFR, however, is still useful even in such a case because the receiver can eliminate the distortion caused by the lengthening or shortening of silence periods between talkspurts with a recovered source clock, which is an inevitable side effect of adaptive receiver buffer algorithms [2].

K. S. Kim is with the College of Engineering, Swansea University, Swansea, SA2 8PP, Wales United Kingdom (e-mail: k.s.kim@swansea.ac.uk).

Existing SCFR schemes extract the information on the source clock frequency from packet interarrival times (e.g., [3], [4], [5]), buffer levels — often combined with packet interarrival times — (e.g., [6], [7]), or timestamps (e.g., [8], [9], [10]). Because the interarrival times and the buffer levels cannot provide the information on the source clock frequency in case of aperiodic packet streams, only timestamp-based schemes can be applicable to the asynchronous SCFR through aperiodic packet streams. To the author's best knowledge, however, SCFR schemes — including timestamp-based ones — have never been applied for and tested with aperiodic packet streams in the literature.

The jitter time-stamp (JTS) approach [9] is close to our work in that it was tested with VBR traffic. It, however, is devised only for on-off type traffic which generates a periodic stream of fixed-size packets during *on* periods. Note that the main purpose of JTS is the recovery of *source rate* defined as the number of packets generated per unit period — rather than the source clock frequency itself — with minimal-length timestamps in the context of AAL2. For this purpose the JTS generates a clock signal used to deliver packets to the application layer with active and inactive periods which are detected from the received on-off type traffic pattern. As for the recovery of source clock frequency, the JTS does not provide any direct way to estimate and/or recover it; it suggests an adaptive adjustment of jitter bias, which may result from changing network conditions and drifting clocks. This procedure, however, is mainly to eliminate any constant bias in dejittering process but not for the effects of clock frequency drift.

In this paper we propose two new asynchronous SCFR schemes for general aperiodic packet streams based on timestamps, which are suitable for the implementation in real systems, and carry out a comparison study with the conventional phase-locked loop (PLL)-based scheme.

II. CLOCK FREQUENCY RATIO ESTIMATION

A. Regression through the Origin (RTO) Model

To recover the source clock frequency without a common reference clock or any reference period in packet streams, we need timestamps in received packets which delivers the information on departure times of packets. The SCFR through periodic packet streams is a special case where the constant packet generation interval, assumed to be known at both the sender and the receiver through service specifications, can be used to extract this information instead of timestamps.

Fig. 1. Illustration of interarrival times, interdeparture times, and their cumulative ratios.

Fig. 2. DCO-based implementation of asynchronous SCFR schemes.

Let $t_s(k)$ and $t_r(k)$ are the kth packet timestamp generated by the source clock whose frequency is f_s and the corresponding arrival time measured with a receiver clock whose frequency is f_r, respectively. As illustrated in Fig. 1, the arrival time $t_r(k)$ includes a packet delay measured in the receiver clock (i.e., $d(k)$) whose true value is unknown.

The arrival times, the timestamps, and the packet delays are modeled by linear regression as follows: For $k \geq 0$,

$$t_r(k) = Rt_s(k) + a + d(k)$$
$$= Rt_s(k) + \alpha + e_k, \quad (1)$$

where R is the ratio of the receiver clock frequency to the source clock frequency, also called *frequency drift*, a is the intercept corresponding to the initial *time offset* between the clocks, $\alpha \triangleq (a + \min_k d(k))$, and $e_k \triangleq + (d(k) - \min_k d(k))$. This model of (1) has been extensively studied in a slightly different context of trend [11] or clock skew [12] estimation where not only frequency ratio but also phase difference between the clocks are to be estimated.

Because we need to estimate only the ratio R for SCFR, however, we can reformulate (1) as a linear *regression through the origin (RTO)* model without a constant term (i.e., α) [13] by subtracting initial values from arrival times, timestamps, and packet delays: For $k \geq 1$,

$$\tilde{t}_r(k) = R\tilde{t}_s(k) + \tilde{d}(k), \quad (2)$$

where $\tilde{t}_r(k) \triangleq t_r(k) - t_r(0)$, $\tilde{t}_s(k) \triangleq t_s(k) - t_s(0)$, and $\tilde{d}(k) \triangleq d(k) - d(0)$. $\tilde{d}(k)$ represents a noise process with a zero mean. From Fig. 1 we can see that $\tilde{t}_r(k)$ and $\tilde{t}_s(k)$ correspond to the sum of interarrival times (i.e., $\sum_{i=1}^{k} iat(i)$) and the sum of interdeparture times (i.e., $\sum_{i=1}^{k} idt(i)$), respectively. For actual implementation, we use these representation by interarrival and interdeparture times for better processing of the wrap-around of counter values.

Fig. 2 shows a digitally-controlled oscillator (DCO)-based implementation of asynchronous SCFR schemes as suggested

in [4], where the estimated clock frequency ratio directly drives a DCO to generate a desired source clock.

B. Least Squares (LS) Estimator

From the linear RTO model of (2), the clock frequency ratio can be estimated recursively using the recursive least squares (RLS) method [14] as follows: For $k \geq 1$,

$$\hat{R}_{LS}(k) = \hat{R}_{LS}(k-1)$$
$$+ P(k)\tilde{t}_s(k)\left(\tilde{t}_r(k) - \tilde{t}_s(k)\hat{R}_{LS}(k-1)\right), \quad (3)$$

$$P(k) = P(k-1) - \frac{P^2(k-1)\tilde{t}_s^2(k)}{1 + P(k-1)\tilde{t}_s^2(k)}. \quad (4)$$

For initialization of the RLS algorithm, we need to set values for $\hat{R}_{LS}(0)$ and $P(0)$ from apriori information like nominal values of clock frequencies [3]. Note that the RLS equations for the linear RTO model are much simpler than typical linear regression models (e.g., those in [8]) and does not depend on any assumption on the noise process.

C. Cumulative Ratio (CR) Estimator

If a packet stream is periodic where $idt(k) = c$ for a constant value c, we can estimate the clock ratio based on the ratio of an interarrival time to an interdeparture time at each packet arrival like many SCFR schemes based on packet interarrival times, i.e.,

$$\frac{iat(k)}{idt(k)} = R + \frac{d(k) - d(k-1)}{c}, \quad k = 1, 2, 3, \ldots \quad (5)$$

In case of an aperiodic packet stream, however, this ratio results in a biased estimate of the clock frequency ratio because the noise component divided by varying interdeparture time, i.e,

$$\frac{d(k) - d(k-1)}{t_s(k) - t_s(k-1)}, \quad (6)$$

has non-zero average in general due to the dependency between the packet delay and the interdeparture time, which has been observed during our simulations.

Fig. 1 suggests an alternative approach for unbiased estimation of the clock frequency ratio for aperiodic packet streams through the ratio of cumulated interarrival times to cumulated interdeparture times. In this case the clock ratio is recursively

estimated at each packet arrival as follows: For $k \geq 1$,

$$\hat{R}_{CR}(k) = \frac{A(k)}{D(k)}, \qquad (7)$$

$$A(k) = A(k-1) + iat(k), \qquad (8)$$

$$D(k) = D(k-1) + idt(k), \qquad (9)$$

where $A(0) = D(0) = 0$. $\hat{R}_{CR}(k)$ can be written as follows:

$$\hat{R}_{CR}(k) = \frac{\sum_{i=1}^{k} iat(i)}{\sum_{i=1}^{k} idt(i)} = R + \frac{\tilde{d}(k)}{\tilde{t}_s(k)}. \qquad (10)$$

Note that the noise component becomes zero as time goes to infinity irrespective of its statistical characteristics. Also note that there are no design parameter values or initial values to set unlike other SCFR schemes.

III. SIMULATION RESULTS

We carried out a comparison study of the proposed SCFR schemes with the conventional PLL-based one [8], [10] using an aperiodic VBR video packet stream from realistic simulation of an access network, where there are 16 subscribers connected through 100-Mb/s user-network interfaces (UNIs) to a shared access network with the same line rate of 100 Mb/s, each of which uses hypertext transport protocol (HTTP), file transfer protocol (FTP) and user datagram protocol (UDP) streaming video traffic with a common application server in the network with a combined traffic rate of about 2 Mb/s per subscriber. The backbone line rate connecting the access network and the server and the end-to-end round-trip time are set to 1 Gb/s and 10 ms, respectively. We use session-level models based on user behaviors for HTTP and FTP traffic. As for UDP streaming video, we generate traffic using a real trace for a common intermediate format (CIF) "Silence of The Lambs", a VBR-coded H.264/advanced video coding (AVC) clip from Arizona state university (ASU) video trace library [15]; 30 frames are generated per second, and transmissions of the packets from a frame are either evenly spread over a frame period (*with frame spreading*) or transmitted at a fixed rate at the beginning of each frame (*without frame spreading*), the latter resulting in on-off type traffic. The average and the variance of the resulting packet interdeparture times in a reference clock (i.e., a global simulation clock) are 5.9026 ms and 0.046159 ms with frame spreading and 5.9051 ms and 0.13472 ms without frame spreading, respectively. For details of the traffic models and their parameter values for the simulation, readers are referred to [16].

The source clock frequency at the server, which is used for generating 32-bit real-time transport protocol (RTP) timestamps for the streaming video, is set to 90.018 kHz with 200-ppm deviation from the nominal value of 90 kHz per RFC 3551 [17]. This clock frequency divided by 50 (i.e., 1.80036 kHz) is also used to transmit packets during active periods when frame spreading is not used; this amounts to the rate of about 21 Mb/s with 1460-byte RTP payload, which is big enough to handle the peak frame rate of the trace (i.e., 19.61616 Mb/s). In case of LS and CR estimators, arrival times are measured using a 48-bit counter driven by the receiver clock with frequency of 15.9968 MHz. For the PLL-based scheme, the free-running frequency of the voltage-controlled oscillator (VCO) is set to 89.982 kHz, and, instead of a counter, floating-point numbers are used to represent arrival times.[1] The local clock frequencies for both the cases amount to the frequency deviation of 200 ppm from the nominal values of 16 MHz and 90 kHz, respectively.[2]

For the LS estimator, we set the initial values of $\hat{R}(0)$ and $P(0)$ to the ratio of nominal clock frequencies (i.e., 16000000/90000) and 10, respectively. For the PLL-based scheme, we assume a critical damping (i.e., damping factor of 0.707) and set the gains of proportional and integral paths to 0.0001 and 0.000001, respectively. Note that the CR estimator has no parameter values to set.

Fig. 3 shows the power spectral densities of the noise components from the simulation, i.e., $\tilde{d}(k)$ in (2) and (10) for the proposed SCFR schemes and $\frac{d(k)-d(k-1)}{t_s(k)-t_s(k-1)}$ in (6) for the PLL-based scheme, which clearly illustrate the effect of frame spreading and resulting traffic pattern: We observe that the use of frame spreading lowers the level of low-frequency components and flattens the overall spectrum, more evidently in the case of $\tilde{d}(k)$ as shown in Fig. 3 (a).

Fig. 4 shows the frequency estimation errors from the three asynchronous SCFR schemes for the UDP streaming video. It is clear from the results that both the proposed schemes outperform the PLL-based one in terms of the speed of convergence and the size of residual errors. Of the proposed schemes, the CR provides nearly equivalent performances with some fluctuations, which is remarkable considering its very low computational complexity (i.e., two storages for $A(k)$ and $D(k)$ and one division and two additions per iteration). Note that the PLL-based scheme is very sensitive to the choice of parameter values, especially the VCO gain; the results shown in Fig. 4 are from the parameter values which produce the best results after many trials.

Comparing the results with and without frame spreading, we found that the frame spreading reduces fluctuations in both the CR estimator and the PLL-based scheme, while the on-off traffic pattern resulting from no frame spreading improves the convergence speed of the LS estimator. As shown in Fig. 3, the frame spreading significantly changes the power spectral densities of noise components, but the results in Fig. 4 show that the proposed schemes are less susceptible to the change in noise component statistics than the PLL-based one.

IV. CONCLUSIONS

Aperiodic packet streams in asynchronous networks brings major challenges to the SCFR; one of them is distortion of packet jitter statistics as discussed with (6), which leads into biased estimation of the clock frequency ratio between the source and the receiver clocks. In this letter we have proposed two asynchronous SCFR schemes for aperiodic

---

[1]The quantization error in phase difference calculation due to the use of a counter makes the PLL never converge to its steady state during the simulation with aperiodic packet streams; this does not happen with periodic packet streams.

[2]The local clock frequency should be high enough (i.e., $f_r \geq 50 f_s$) in order to reduce quantization errors related with the DCO-based clock generation [10]. This is not the case for VCO-based clock generation.

(a)

(b)

Fig. 3. Power spectral density of the noise component in SCFR for UDP streaming video: (a) $\tilde{d}(k)$ in (2) and (10); (b) $\frac{d(k)-d(k-1)}{t_s(k)-t_s(k-1)}$ in (6).

(a)

(b)

Fig. 4. Frequency estimation errors for UDP streaming video: (a) With frame spreading; (b) without frame spreading (i.e., on-off traffic).

packet streams, one based on the RLS method and the other based on simple heuristics of cumulative ratio of interarrival and interdeparture times. Both the schemes are not based on any assumption on the noise process and greatly reduce computational complexity through the formulation of SCFR with timestamps as a linear RTO model. Simulation results demonstrate that the proposed schemes outperform the conventional PLL-based one in terms of the speed of convergence and the size of residual errors in estimating the source clock frequency.

REFERENCES

[1] ITU-T, *ITU-T Recommendation I.363.1, B-ISDN ATM adaptation layer (AAL) specification: Type 1 AAL*, ITU Std., Aug. 1996.

[2] H. Melvin and L. Murphy, "Time synchronization for VoIP quality of service," *IEEE Internet Computing*, vol. 6, no. 3, pp. 57–63, May/Jun. 2002.

[3] K. S. Kim and B. G. Lee, "KALP: A Kalman filter-based adaptive clock method with prefiltering for packet networks use," *IEEE Trans. Commun.*, vol. 48, no. 7, pp. 1217–1225, Jul. 2000.

[4] J. Aweya, D. Y. Montuno, M. Ouellette, and K. Felske, "Clock recovery based on packet inter-arrival time averaging," *Comput. Commun.*, vol. 29, no. 10, pp. 1696–1709, Jun. 2006.

[5] R. V. dos Santos and L. H. A. Monteiro, "A scheme for synchronizing clocks connected by a packet communication network," *Commun. Nonlinear Sci. Numer. Simulat.*, vol. 17, no. 7, pp. 2722–2727, 2012.

[6] R. P. Singh, S.-H. Lee, and C.-K. Kim, "Jitter and clock recovery for periodic traffic in broadband packet networks," *IEEE Trans. Commun.*, vol. 42, no. 5, pp. 2189–2196, May 1994.

[7] H. Bang, J.-O. Choi, S.-R. Lee, and C.-S. Park, "Adaptive clock recovery mechanism having dynamic frequency drift and buffer level control in packet switched networks," *IEEE Commun. Lett.*, vol. 15, no. 5, pp. 575–577, May 2011.

[8] R. Noro and J.-P. Hubaux, "Clock synchronization of MPEG-2 services over packet networks," *Telecommunication Systems*, vol. 11, pp. 3–16, 1999.

[9] W. Su and I. F. Akyildiz, "The jitter time-stamp approach for clock recovery of real-time variable bit-rate traffic," *IEEE/ACM Trans. Netw.*, vol. 9, no. 6, pp. 746–754, Dec. 2001.

[10] J. Aweya, M. Ouellette, D. Y. Montuno, and K. Felske, "Circuit emulation services over Ethernet-Part 1: clock synchronization using timestamps," *Int. J. Network Mgmt.*, vol. 14, no. 1, pp. 29–44, Jan. 2004.

[11] T. Trump, "Maximum likelihood trend estimation in exponential noise," *IEEE Trans. Signal Process.*, vol. 49, no. 9, pp. 2087–2095, Sep. 2001.

[12] S. B. Moon, P. Skelly, and D. Towsley, "Estimation and removal of clock skew from network delay measurements," in *Proc. of INFOCOM'99*, New York, NY, Mar. 1999.

[13] J. G. Eisenhauer, "Regression through the origin," *Teaching Statistics*, vol. 25, no. 3, pp. 76–80, 2003.

[14] L. Ljung, *System identification: Theory for the user*. Englewood Cliffs, New Jersey: P T R Prentice Hall, 1987.

[15] G. V. der Auwera, P. T. David, and M. Reisslein, "Traffic and quality characterization of single-layer video streams encoded with H.264/AVC advanced video coding standard and scalable video coding extension," *IEEE Trans. Broadcast.*, vol. 54, no. 3, pp. 698–718, Sep. 2008.

[16] K. S. Kim, "Effect of ISP traffic shaping on user-perceived performances in broadband access networks," in *Proc. of the 4th International Congress on Ultra Modern Telecommunications and Control Systems (ICUMT 2012)*, Petersburg, Russia, Oct. 2012, pp. 533–538.

[17] H. Schulzrinne and S. L. Casner, "RTP profile for audio and video conferences with minimal control," RFC 3551 (Standards Track), Internet Engineering Task Force, Jul. 2003, obsoletes RFC 1890.