VIDEO STREAMING SYSTEM AND METHOD FOR JITTER BUFFER STATISTICS

Dr. (Mrs.) K. Meena
Former Vice-Chancellor
Bharathidasan University, Trichy , India

J. Thilagavathi
Computer Science Department
Govt. Arts College Karur, India

***Abstract*— In this paper the basic concepts used in the multidimensional QoS inference for video streaming are introduced. An overview of video streaming and the Real-Time Transport Protocol (RTTP) is provided. Next, the main aspect which is related to the jitter buffer in a video context is presented. The notion of machine learning is introduced as well. The concept of dynamic statistics is finally demystified with a brief introduction and some examples.**

Keywords- IP, RTTP, RTCP,SDP,UDP, QOS, Video Streaming, Jitter Buffer

I. INTRODUCTION

Cisco Systems claims in its annual networking index of 2010 that back then 40% of Internet's traffic is related to video [1]. By 2014 Cisco Systems forecast that more than 90% of Internet's traffic will be originating from video services, of which 13% will have its origin in real-time video applications. Cisco Systems also predicts that up to 66% of the world's mobile data traffic by 2014 will be video. Besides the growing demand for VoD, traditional video providers adopt emerging Internet Protocol (IP) technologies for the delivery of video content, which contributes to the growing IP traffic [2]. Today, a wide array of video streaming technologies exists. Both open-source and proprietary solutions are available. Real Time Transport Protocol(RTTP) [3] is one of the most popular open standard streaming techniques. The 3rd Generation Partnership Project (3GPP) Packet-switched Streaming Service (PSS) [4] puts Real Time Transport Protocol forward as the main component for the transport of media streams. Adobe's Hypertext Transport Protocol (HTTP) [5] Dynamic Streaming System is an example of a proprietary streaming solution and it is used in, e.g., Akamai's High Definition (HD) [6] network. Real Networks [7] are pioneers in streaming multimedia and offer proprietary solutions such as real Video. YouTube [8] encodes the videos in a Flash Video (FV) container for playback in Flash enabled browsers and it uses Real Time Transport Protocol (RTTP) for its mobile variant. In the following, the focus will be on Real Time Transport Protocol as it is an open standard and knows wide adoption in all kinds of environments.

The Real-time Transport Protocol is defined in RFC 3550 and it is a common choice to transport multimedia streams over the Internet. This protocol provides loss detection, synchronization and sequencing services as well as security [9]. Designed by the Internet Engineering Task Force (IETF), Real Time Transport Protocol provides end-to-end network transport functions suitable for applications transmitting real-time data (such as audio, video or simulation data), over multicast or unicast network [10]. It does not provide any QoS guarantees however. Each Real Time Transport Protocol(RTTP) stream maintains a queue, the jitter buffer, in which it buffers incoming data. Every Real Time Transport Protocol packet is labeled with a sequence number and timestamp. The incoming data in the queue is arranged such that the data assumes the same sequence as it was originally coded on the streaming server. This is essential for the playback of the media. The jitter buffer is discussed in more detail in section 6.1.1.

The Real-time Transport Control Protocol (RTCP) is used in connection with to create a closed-loop feedback system between the media-player and the streaming server. RTCP disseminates periodic statistical performance measures between all participants in the Real Time Transport Protocol session, including all active media players and the streaming server. The primary function of RTCP is to provide feedback on the quality of the data distribution [10]. Based on this information, the Real Time Transport Protocol (RTTP) streaming server may optimize the performance through adaptive encoding if supported. Other parties can use this feedback to detect and anticipate network problems. The RTCP feedback is also used for congestion and traffic control purposes, although RFC 3550 doesn't state explicitly how to achieve this. Congestion control in Real Time Transport Protocol is not always straightforward as Real Time Transport Protocol streams are often inelastic, i.e., generated at a fixed or controlled rate.

While Real Time Transport Protocol is used to transport time-sensitive data which is stated that the Real-time Transport Streaming Protocol (RTSP) is used to control the on-demand delivery of real-time data. RFC 2326, which describes RTSP, defines request messages such as play, stop, pause, and describe. Most RTSP messages are sent by the client to the RTTP streaming server. RTSP shows overlapping features with the Session Initiation Protocol (SIP) and the Session Portrayal Protocol (SDP). SIP/SDP is however used to establish multimedia sessions whereas Real Time Transport Protocol addresses streaming media systems [11]

The Darwin Streaming Server (DSS) [12] is an open-source streaming server released under the Apple Public License Software that shares the same code base as Apple's proprietary QuickTime Streaming Server (QTSS). DSS supports digital media standards such as 3GPP and Motion Picture Experts Group (MPEG) version 4 for streaming over RTTP/RTSP. DSS provides Reliable User Datagram Protocol (UDP) support. Reliable UDP is an extension to the default UDP that enables Transport Control Protocol (TCP)-like features such as retransmission and congestion control but also properties such as overbuffering1. Reliable UDP is enabled by setting the proper fields in the RTSP SETUP request during client-server negotiation. Major content distribution companies such as Akamai and Apple TV have DSS/QTSS based solutions for their video services. The DSS is available online from Mac OS Forge and it can be installed and launched under Linux and Mac OS.

II. VIDEO STREAMING

Video streaming services is admired among the Internet users all over the world. The reputed Companies such as NetFlix, Youtube and others, have been successful in disseminating video content to any customer, irrespective of the customer's geographical location. Besides VoD, video streaming over the Internet is increasingly used in applications such as remote surveillance, security monitoring, smart home, environmental tracking, battlefield intelligence, distance learning, collaboration and interactive virtual environments [13].

III. JITTER BUFFER

In streaming media applications, the jitter buffer, or play-out buffer, is a queue at the receiver's side that temporarily buffers data frames before these are processed by the media decoder or media player. The motivation behind the deployment of a jitter buffer is to mask the effects of fluctuating network QoS on the delivery of media. The length of the jitter buffer varies depending upon the particular application.

For example, for Voice over IP (VoIP) and video conferencing applications the queue length is minimized to reduce the communication latency. In VoIP settings the jitter buffer length varies around 15 ms with a maximum of 30 ms [14]. For other video streaming applications, e.g., video streaming in browsers such as YouTube, the queue length is the size of the media itself. In such cases, the necessary storage space should be available beforehand. For desktop computers and laptops this is usually not a problem, but hand-held devices are often constrained in memory space. The Real-Time Transport Protocol implementation available in Android 1.6 often allocates a buffer size of around 170 kB [15]. When using the DSS and Android 1.6 together, a buffer usages of about 60% is targeted2. The actual buffer usage shows a standard deviation around 25 KB, peaks larger than this are not unusual. This behavior is inherent for VBR video and fluctuating network resources.

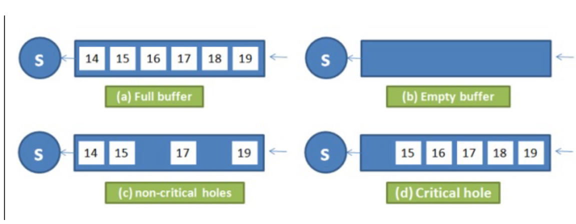

Figure 1: Four Different State of the Jitter buffer

The media player can maintain several jitter buffers in parallel, depending on the amount of channels defined in the media file to be streamed. For a basic video stream, there is usually one channel for audio and a second for video. Each jitter buffer can be considered as a G/D/1 queue. Here the data arrival process is assumed to be a general process, and the frames3 are consumed at deterministic time-intervals. This often resembles the frame-rate of the video or the sample-rate of the audio. Other research assumes M/D/1 queues which include Poison arrival rates [16].

The major objective is to infer the state of the jitter buffers through network (multidimensional) QoS indicators. It is because the state of the jitter buffer is an indicator of the network layer's performance and it has immediate effect on the media player's ability to display streamed media in a timely fashion. Four types of states of the jitter buffer are distinguished: full buffer, empty buffer, critical holes and noncritical holes. An overview is shown in Figure 6.1. A full buffer (Figure 6.1(a)) is defined to be a state when there is at least one media packet in the queue and all sequence numbers starting from the first packet after the last consumed, till the last received packet are present in the buffer. An empty buffer (Figure 6.1(b)) on the other hand doesn't have any packet in the queue. Figure 6.1(c) shows non-critical holes. These are missing sequence numbers, which are not first in the queue. Last, critical holes are shown in Figure 6.1(d). Critical holes manifest when packets that are supposed to be first in the queue are missing. It is clear that an empty buffer will temporarily halt the media playback.

For video streaming the image on the screen could freeze or, in case of audio, the sound could stop. The behavior of the play-backed media in such cases is depended on the used decoder. Similarly, a missing packet, i.e., critical hole, may effect the media quality. Depending on the codec of the streamed media, the media quality may be impaired in different ways, e.g., for video: color changes or blurred pictures, short break. An empty buffer is considered to be more critical for the timely playback than a critical hole.

A non-critical hole can become a critical hole when the non-critical hole progresses to the first place in the queue. The non-critical hole could also turn into a full buffer when the missing sequence numbers arrive at a later point in time. This can happen as a result of reordering effects on the network or because of packet retransmission due to losses or corruptive communication. When the jitter buffer is empty for some time the streaming application may decide to temporarily halt the playback of the media. During this time, the jitter buffer has the opportunity to populate again while no frames are retrieved, this is also called rebuffering. The focus is on the identification of empty buffers as this particular case has the most severe impact on the application layer. Empty buffers affect the temporal playback of the media. Critical holes may mostly affect the special quality of the media. In this context, the notion of empty buffers and critical holes is in line with the Continuity Index (CI) that was initially defined in the work [17]. The CI is interpreted as the percentile of packets, or pieces in BitTorrent terms, that meet their play-out deadline. The CI was previously used in research on the evaluation of swarm video streaming over P2P networks [18] [19].

IV. THE JITTER BUFFER SOLUTION STREAM MODEL

The jitter buffer is modeled as a fluid flow model. For now, the media stream arriving at, and departing from the jitter buffer is considered to be a bit stream. Let Ψ(t) be the amount of data in the jitter buffer at time t. Ψ(t) is the amount of downloaded data, minus the amount of data retrieved by the video decoder up to time t. The rate at which Ψ(t) changes over time is given by

$$\frac{d\Psi(t)}{dt} = \varphi(t) - \zeta(t) \quad (1)$$

Where ζ (t) is the data rate at which the decoder retrieves data from the jitter buffer and φ(t) is the arrival rate of the data to the buffer. φ (t) is a random process equal to the transmission rate and depending on the transmission policy of the streaming server. ζ (t) is a batch process proportional to the frame rate of the video. φ (t) may be affected by the network if it is unable to sustain the transmission. Thus, during transmission φ (t) is modulated by events that can be characterized by QoS parameters.

If φ (t) =ζ (t) then Ψ (t) will be constant, hence the amount of data in the jitter buffer remains stable. When φ (t) > ζ(t), the buffer size grows and for φ (t) < ζ(t) the buffer size shrinks. Depending on the application, φ (t) is aimed to be larger than ζ (t) or equal by controlling the transmission rate. When no constraints exist on memory, in fast streaming scenarios such as watching a video in a browser, φ(t) is often much larger than ζ(t).

Other applications target φ(t) = ζ(t), e.g., for media with unknown length or for resource-constraint devices. In both cases dΨ(t)/dt ≥0 to sustain a seamless media playback from the jitter buffer. dΨ(t)/dt < 0 for a considerable time might indicate unreliable network conditions. This will eventually lead to buffer underflow or exhaustion if the network performance doesn't improve. Thus dΨ(t)/dt indicates the ability of the network to deliver a media stream over a network in a timely fashion. dΨ(t)/dt is defined as a parameter that describes the QoD of a video stream and henceforth referred to as *w*(t):

$$\Psi(t) = \int_0^t \omega(t)dt + c \quad \text{and} \quad \varphi(t) - \zeta(t) \quad (2)$$

Where c is a constant that reflects the initial conditions of the buffer at t = 0. Ψ(0) is usually zero. This implies that no data is available for the decoder at the start of the media streaming. Based on equation 2, a jitter buffer starvation is formally defined as follows.

Definition 1 when streaming a video with length *g*, an empty buffer, or jitter buffer starvation, manifests when the buffer on the receiver's side contains no information:

$$\Psi(t) = 0, \qquad 0 < t < \gamma \quad (3)$$

Note that $\omega(t)$ has a lower bound defined by ζ(t) given that ζ(t) ≥ 0 as long as Ψ(t)≥ 0 and φ(t) is always larger or equal to zero. The Real-Time Transport Protocol streaming server usually streams at the same rate as media is consumed by the media player for real-time streaming. Under ideal conditions ζ (t) ≈ φ (t), hence the $\omega(t)$ is close to zero. This holds for constant bit rate videos. For variable bit rate videos $\omega(t)$ will vary more than for constant bit rate videos. But the average $\omega(t)$ will be close to zero. Under certain circumstances, based on RTCP feedback, the transmission rate φ (t) may be increased or decreased temporarily, e.g., to avoid congestion on the network. Also traffic control is sometimes used to amend the transmission rate in real-time with the target of a constant jitter buffer population. This can be useful when streaming variable bit rate media.

The jitter buffer can starve because data isn't transmitted at a minimal rate to maintain a certain population of the jitter buffer. On the other hand, starvation of the jitter buffer could be the consequence of packet losses on the path to the jitter buffer. These causes should be distinguished. They could potentially take place at the same time.

Also, there is a small lag between the actual start of the streaming and the start of the media consumption from the jitter buffer. This is to allow the jitter buffer to populate before the actual playback of the media starts. It has been shown that for longer start-up latency, the chance of jitter buffer starvation decrease [20] [21].

V. VIDEO STREAMING

A video encoder produces a video at a constant or variable bit rate. The bit rate of a Constant Bit Rate (CBR) video is fixed over time whereas a Variable Bit Rate (VBR) video exhibits a fluctuating bit rate. The produced variable bit rate is dependent on the codec and content of the video, and shows long-range dependent properties [22]. Scenes with lots of motion are often encoded with more bits to maintain the quality level of the images. The minimum bit rate for videophones is 16 kbit/s, HDTV requires 8 to 15 Mbit/s, the Blue-Ray optical disc format has a maximum bit rate of 40 Mbit/s.

Both TCP and UDP preserved to be used in the transport media. Some protocols consented to switch in between the protocols and during the runtime on client's request. TCP provides more features, compared to UDP, such as reliable end-to-end communication and congestion control. Sometimes an application layer process is programmed to provide congestion control for UDP as well as other features, e.g., retransmission. Reliable UDP used by the QuickTime Streaming Server (QTSS) is such example [15].

TCP's congestion control is realized by altering the transmission rate of packets. While streaming media, the application layer has also other alternatives to control the transmission rate. For example, frames can be dropped before the actual transmission; it is referred to as stream thinning. Stream thinning results are in a lower transmission rate without affecting the timeline of the media. This is possible because frames in a video stream maintain a hierarchical structure called the Group of Pictures (GOP).

The structure of a video frame is a combination of I, P, and B frames. I frames are independent reference pictures, fully specified. P, and B frames can be seen as enhancement layers dependent on other frames. Scalable Video Coding (SVC), part of the H.264/MPEG-4 AVC standard, defines a video bit stream to constitute of one or more subsets of bit streams (inclusion of P and B frames). When stream thinning is enabled one or more enhancement layers or bit streams are pruned and, as a result the timeline of the media will not be affected. TCP's adaptation of the packet transmission rate results in a stretch or contraction of the timeline as after all, the whole media file will be send. This may result in jitter buffer exhaustion, or possibly the freeze of the video, if the frames did not arrive before their play out deadline. Due to the use of stream thinning, the streamed media is less prone to Jitter buffer exhaustion compared to congestion control on the transport layer. However, as part of the media content is dropped, the quality of the video may degrade. But the advantage is that the media is less likely to freeze during play-back.

Multiple Bit Rate (MBR) [23] or Adaptive Multi Rate (AMR) encoding is another technique for real-time congestion control. MBR allows for dynamic switching between pre coded videos with different bit rates showing the same content. The 3GPP PSS proposes the use of MBR for adaptive streaming. As with stream thinning, switching between bit rates in MBR encoded videos is useful to adapt to the changing bandwidth constraints on a network. When bandwidth becomes more scarce, the MBR stream server switches to the video with a lower bit rate, resulting in a lower network load.

Similarly, MBR can switch to a higher bandwidth when more network resources are available. Fast streaming is frequently used by web browser players. Fast streaming is a technique where the media file is delivered as fast as possible to the user in order to reduce the startup latency and to protect against negative effects of bandwidth fluctuations. Even if MBR, fast streaming and rate adaption are novel techniques, Guo et al. reported that they tend to over-utilize CPU usage and bandwidth resources to provide the user with a better experience [24]. Effective and efficient quality control of streamed media is not a straightforward task.

VI. QOS PARAMETERS

The term Quality of Service (QoS) has been popular for more than 20 years. Though, there is little consensus of what QoS exactly incorporates. The term QoS can be construed intuitively as a measure for how well a particular service performs. In the context of video transmission QoS is used as the quality quantification of a communication channel in terms of measurable parameters meaningful to a computer network. QoS in the context of computer networks, however, has a wider variety of interpretations. For example, in traffic engineering, QoS also refers to the capability of service guarantee and the provisioning of network resources.
Conventional QoS metrics can be measured and assessed from an end-to-end point of view. In particular, a set of QoS metrics is needed that are applicable and essential in the streaming media context. An overview of common QoS metrics include but is not limited to:

- Latency – also referred to as One-Way Delay (OWD), is the time taken for a data packet to travel from its source to destination. Delay is induced by queues and transmission delays on the path of the packet.
- Delay variation – the OWD of packets commonly varies per packet. The differences in latencies of consecutive packets are referred to as the Inter-packet Delay Variation (IPDV).
- Reordering – a stream of packets sometimes arrives at its destination in a different order than it was initially transmitted.
- Errors – or corruption, occur during transmission when the content of packets on the path changes due to unforeseen circumstances.
- Bandwidth – knows several definitions but in this context bandwidth is defined to be the data rate or capacity of a communication channel. The bandwidth availability of a channel may fluctuate over time.
- Packet loss – some packets may be lost during transmission. This can for example be the result of packet collisions, queue overflows, or physical damage to the transmission medium.

Any of the above phenomena can affect the quality of a multimedia stream when they manifest or fluctuate. Though, not all of the stated QoS metrics are easily measurable. For instance, to measure the OWD accurately, one needs synchronized clocks between sender and receiver. The synchronization is not a simple task, special equipment or protocols need to be deployed to achieve high accuracy. For everyday use, an accurate synchronization system is not feasible for current retail devices.

Measuring the number of erroneous packets is also not straightforward. Error control is done at various layers of the network stack. For example, every Ethernet datagram carries a Cyclic Redundancy Check (CRC) checksum, the Internet Protocol (IP) provides error control for the header, and TCP calculates a checksum for both data and header. UDP provides, similar to TCP, checksum capabilities, but this is not a mandatory feature. Depending on the policy of the protocols, erroneous packets are dropped or retransmitted. This happens transparently to higher layer in the network stack and it is therefore hard to track from the application layer. Consequently, the QoS metrics error and OWD are not viable metrics for our purposes.

Streamed multimedia needs some minimum bandwidth, constant or variable, in order to have a guaranteed quality level. If this bandwidth is not available, the streamed media may show quality deterioration, e.g., in the form of visual impairments. Measuring the bandwidth availability is difficult without perturbing the network. Currently, generating packet trains is the best known methodology to estimate the available bandwidth in high-speed networks. This means that some data sequences are generated from one point in the network to another, to assess the end-to-end network characteristics. In this case, the measurements are called active measurements. This is in contrast to passive a measurement that infers information at a given point in the network.

Additionally, the media arrival rate can be measured. The Packet Arrival Rate (PAR) is a straightforward measure to quantify the arrival rate. PAR can be defined in bytes or packets received per time interval. PAR is however content dependent on the media of concern. A deviation in PAR for CBR media is more likely to be an indicator of changing network conditions than it would be for the VBR case. Though, extreme values for PAR in the VBR case might give insight in degrading network conditions as well.

VII. DYNAMIC STATISTICS

In the conventional linear regression model, the sum of squared residuals is minimized. The standard deviation of the residuals of error terms is assumed to be constant and normally distributed. The treatment of random variables [25] can be represented with the location model:

$$x_i = \mu + u_i \quad i = 1 \ldots\ldots n \tag{4}$$

Where μ is the true value of X and the errors u_i are random variants from u that act additively. U can be considered as noise emanating from, e.g., natural deviations, measurement errors and environmental influences. The aim of statistical methods is to estimate $\hat{\eta} = \hat{\eta}(x1, x2, x3, \ldots, xn) = \hat{\eta}(\mathbf{x})$ such that it is as close to the real value μ as possible.

When a vector of random variates x_i shows equal finite variance, the sequence of x_i is called homoscedastic [26]. Homoscedasticity and normality is the basic assumption for many classical regression models and other statistical methods. The assumption of homoscedasticity and normal distributed errors does, however, not always hold in reality. When this happens, classical statistical methods may yield biased results. Dynamic statistical methods can be used in this case to produce more accurate estimations.

Figure 2 shows an example of the difference in performance between a dynamic and classical statistical method. This is an excerpt from previous work on the application of the M-A1 (M-A) to QoS measurements [27]. The theoretical *c*2 97.5% tolerance ellipse is plotted for the Delay Jitter (D_J) and Zero Throughput Time (T_Z) with a classical method based on the arithmetic mean, and a dynamic statistical method based on the Minimum Covariance Determinant (MCD). The left crosshair is the estimation of location with the dynamic estimator, and the right crosshair is computed via the classical way

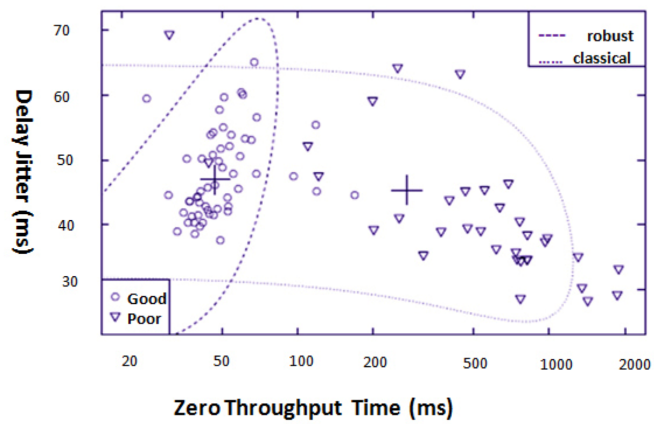

Figure 2: The 97.5% tolerance ellipses for the Zero throughout Time and the Packet Delay Variation computed with classical and dynamic statistical methods. The dynamic ellipse is able to classify the good points better than compared to the classifier based on classical statistical methods.

The ellipses try to capture the variance of the bi variate data. It is observed that the ellipse and the estimation of location for the classical method is inflated towards the outliers. The dynamic ellipse and estimation of location describes the bulk of the point cloud well, regardless of the outliers. The glyphs in the plot are related to the user's perception during the experiment of concern. It is observed that the dynamic ellipse is able to separate the triangular points from the round points in the center of the point cloud with about 95% accuracy. The performance is different from the classical ellipse's classification accuracy. In this context, the dynamic analysis revealed more information than the classical statistical procedure did.

Dynamic statistics can be considered as the stability theory of statistical procedures [28]. Dynamic statistics aims to minimize the influence of outlying data objects and construct models and estimates that well describe the bulk of the point cloud [29]. This is often achieved by down weighting outlying objects to minimize their influence on an estimator. The trimmed mean is a straightforward application of down weighting the n most outlying objects to zero. In general, dynamic procedure not necessarily down weights outlying objects to zero. Dynamic statistics provide objective methods to select the objects to label as outlying and how to down weight them. Objective methods outperform subjective methods for identifying outliers, especially in small sample sets.

Besides poor effectiveness, there are other reasons to avoid subjective outlier detection methods. The classification [25] of good measurement points as a typical must be minimized, outliers inherent to the observed phenomenon should not be removed, and, to assess the statistical behavior of subjective procedures is a challenge.

Dynamic statistical methods are of particular interest for the purpose of multidimensional QoS inference. Understanding the processes that govern the QoS on the network level is partly based on the interpretation and numerical study of QoS data exploration. The goal is to find out QoS indicators that identify the influential measurement points with regard to application layer performance of multimedia streaming. In Previous work on the M-A it was observed that the QoS values most outlying with regards to the measurement's average state were most influential on the quality perception of streamed video. The plot of Figure 6.2 is part of the outcome of this study. To identify these leverage points (influential QoS measurement points), there is a need to apply proper statistical methods. For this purpose dynamic statistical methods are used.

A general introduction to dynamic statistics can be found in the works of Hampel, Hubert and Shurygin. Additionally, Maronna et al. provide an elaborate background and a detailed overview of the latest evolvements in the theory and methods of dynamic statistics.

VIII. DELAY VARIATION

Different definitions of Delay Variation (DV) are circulating and standardized within the research community. DV is sometimes referred to as jitter, inter-arrival jitter, or packet delay variation. RFC 5481 addresses the fuzzy definition of DV and puts forward the DV definition based on the IETF and the International Telecommunication Union (ITU) to be most appropriate for packets when referring to data streams: the quantification of a path's ability to transfer packets with consistent delay. Two definitions of DV are mainly used in industry and academia: the Inter-packet Delay Variation (IPDV), which describes the difference in consecutive OWDs, and the Packet Delay Variation (PDV), which compares the difference in OWD to a fixed reference point.

The actual computations of PDV and IPDV avoid the necessity to measure the OWD with synchronized clocks. Instead, the unsynchronized OWD D_n of packet n is measured and calculated as

$$Dn = T_{R,n} - T_{S,n} \qquad (3.1)$$

Where $T_{S,n}$ is the local departure time of packet n at the sender's side, and $T_{R,n}$ is the local arrival time of corresponding packet at the receiver's side.

The IPDV of packet n is obtained by comparing the unsynchronized OWD of packet n to the preceding packet

$$\begin{aligned} IPDVn &= D_n - D_{(n-1)} && (3.2a)\\ &= (T_{R,n} - T_{S,n}) - (T_{R,(n-1)} - T_{S,(n-1)}) && (3.2b)\\ &= (T_{R,n} - T_{R,(n-1)}) - (T_{S,n} - T_{S,(n-1)}) && (3.2c) \end{aligned}$$

To estimate the delay variation, the real time transport protocol RFC recommends the IPDV definition. The computation of the IPDV with reference to the previous packet (as in equation (3.2c)) can yield biased results. For example, a long lasting increase in OWD is only reflected in two IPDV estimate, at the start and the end. To minimize the bias in PDVs calculation, RFC 5481 recommends not using the previous packet as a reference but instead to use D_{min}, the lowest D measured. Equation 3.2a thus becomes:

$$PDV = D_n - D_{min} \qquad (3.3)$$

In real-time environments D_{min} is however not always known in advance. Given that D_n might be available, e.g., at the start of a measurement, D_{min} may change over time. To address this problem the PDV can be computed over a given time interval as the standard deviation D_J of D_n with reference to the first unsynchronized OWD in that particular time interval.

D_J is then defined as

$$\sqrt{(1/(N-1))\ \sum_{(n-1)}^{N} (D_{d}-\bar{D})^2} \qquad (3.4)$$

Where D is the average unsynchronized OWD of the time interval, D_d is the PDV with reference to the first packet of the time interval: $D_d = D_n - D_1$, where D_n is the unsynchronized OWD of packet n. N is the total number of packets within the time interval. Equation (3.4) in its current form requires to store all values of D_d in order to compute $(D_{S,n} - D)^2$ at the end of the time interval. This might not be favorable or feasible for memory constrained devices.

By rearranging the symbols under the root of equation (3.4) the following derivation is obtained

$$D_J = \sqrt{(1/(N-1))\ \sum_{(n-1)}^{N} D^2_{d} - \frac{N}{N-1} D^2} \qquad (3.5)$$

As a result, for the calculation of D_J, one needs to maintain only three variables: N, $\sum D^2_d$ and D. Clearly, this is more advantageous than maintaining an array of a priorly unknown length for the computation of D_J (as in equation (3.4)).

The D_J calculation from equation (3.5) describes a Ring Buffer (RB) algorithm. Non-RB calculation algorithms store historical values in a database and compute the estimates afterward. The RB algorithm does not need to store any historical timestamps or sequence numbers. Thus a RB is more efficient and desirable in resource constraint environments.

In the study by Verdoy six different delay jitter methods were studied. The study pointed out that the D_J from equation (3.5) is the optimal method with regard to packet inconsistencies, memory requirements and ease of deployment. An overview of other DV definitions is available in Verdoy's work and RFC 3393.

IX. CONCLUSION

In this paper Dynamic statistics is been discussed. It has been introduced as a tool to improve the reliability of the measurement data. Dynamic statistical methods are expected to help in identifying problematic network conditions. The jitter buffer was described along with an overview of streaming video techniques. Four states of the jitter buffer have been identified which will be matched against QoS measurements in the experiments.

REFERENCES

[1] Cisco Systems Inc. Cisco Visual Networking Index: Forecast and Methodology, White Paper, pp. 2009-2014, (June 2010).

[2] Norton, W. B. Video internet: The next wave of massive disruption to the u.s. peering ecosystem, In Asia Pacific Regional Internet Conference on Operational Technologies (APRICOT), (February 2007).

[3] Guo, L., Tan, E., Chen, S., Xiao, Z., Spatscheck, O., and Zhang, X, "Delving into internet streaming media delivery: a quality and resource utilization perspective", In Proceedings of the 6th ACM SIGCOMM conference on Internet measurement (New York, USA), IMC '06, ACM, pp. 217–230, (2006).

[4] 3GPP TSG-SSA, Transparent end-to-end Packet-switched Streaming Service (PSS); protocols and codes (release 9), TS 26.234 V9.4.0, (2010).

[5] Adobe. HTTP dynamic streaming on the Adobe Flash platform. White Paper, (2010).

[6] S. Shunmuga Krishnan, and Ramesh K. Sitaraman, "The Impact of Video Stream Quality on Viewer Behavior", Proceedings of the Internet Measurement Conference 2011 Boston, Massachusetts, November 14-16, (2011).

[7] RealNetworks, "Minimize Rebuffering and Maximize Media Encoding Rate Delivered: Helix Mobile Server Rate Control for Mobile Networks", Real Networks White Paper Series. Version 2.1, pp. 7-14, 32-36, (January 2010).

[8] Hacettepe Üniversity, Eğitim Fakültesi Dergisi, Journal of the Faculty of Educational Sciences of Hacettepe University, Tsuji, Yohko, pp. 131-140, (2011).

[9] Hanssen, F., and Jansen, P. G. Real-time communication protocols: an overview. University of Twente CTIT technical report series 03-49, (2003).

[10] Schulzrinne, H., Casner, S., Frederick, R., and Jacobson, V. RTP: A Transport Protocol for Real-Time Applications, RFC 3550 (Standard), Updated by RFCs 5506, 5761, 6051, (July 2003).

[11] Lindquist, J., Maenpaa, J., Rajagopal, P., and Marjou, X. SIP/SDP overlap with RTSP. Informational Internet-Draft, MMUSIC WG, (Dec. 2009).

[12] Apple Computer, Inc. QuickTime streaming server modules programming guide, pp. 106–107, (2011).

[13] He, Z., and Xiong, H, "Transmission distortion analysis for real-time video encoding and streaming over wireless networks", IEEE Trans. Circuits Syst. Video Technology, Vol. 16, no. 9, pp. 1051–1062, (2006).

[14] Small C. G, "A survey of multidimensional medians. International Statistical Review 58", pp. 263–277, (1990).

[15] Apple Computer, Inc. Quicktime streaming server modules programming guide, pp. 106–107, (2011).

[16] ParandehGheibi, A., Médard, M., Shakkottai, S., and Ozdaglar, A, Avoiding interruptions - QoE trade-offs in block-coded streaming media applications. The Computing Research Repository abs/1001.1937, (2010).

[17] J. Peltotalo, J. Harju, M. Saukko, L. Vaatamoinen, I. Bouazizi, I. Curcio, and J. van Gassel, "A real-time peer-to-peer streaming system for mobile networking environment", in INFOCOM Workshops, (2009).

[18] Erman, D. On BitTorrent Media Distribution. Ph.D thesis, Blekinge Institute of Technology, (2008).

[19] Erman, D.and Yao, Y., De Vogeleer, K., and Popescu, Evaluation of swarm video streaming. In International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT) (Moscow, Russia), IEEE, pp. 1172 –1179, (2010).

[20] Liang, G., and Liang B, "Balancing interruption frequency and buffering penalties in VBR video streaming", In 26th IEEE International Conference on Computer Communications, pp. 1406 –1414, (May 2007).

[21] ParandehGheibi, A., Médard, M., Shakkottai, S., and Ozdaglar, A. E, Avoiding interruptions - QoE trade-offs in block-coded streaming media applications, (2010).

[22] Beran, J., Sherman, R., Taqqu, M., and Willinger, W, Long-range dependence in variable-bit-rate video traffic. IEEE Transactions on Communications vol. 43, no. 234, pp. 1566–1579, (April 1995).

[23] Sun,W., Lu, Y., andWu, F, Bit-stream switching in multiple bit-rate video streaming using wyner-ziv coding. In IEEE International Conference on Multimedia and Expo, pp. 1665–1668, (2006).

[24] Guo, L., Tan, E., Chen, S., Xiao, Z., Spatscheck, O., and Zhang, X, Delving into internet streaming media delivery: a quality and resource utilization perspective. In Proceedings of the 6th ACM SIGCOMM conference on Internet measurement (New York, NY, USA), IMC '06, ACM, pp. 217–230, (2006).

[25] Maronna, R., Martin, D., and Yohai, V, Robust Statistics: Theory and Methods. John Wiley & Sons, New York, pp. 1694–1711, (2006).

[26] Wilcox, R, "Introduction to robust estimation and hypothesis testing - Statistical modeling and decision science", Elsevier/Academic Press, pp. 327-336, (2005).

[27] De Vogeleer, K., ˙Içkin, S., Fiedler, M., Erman, D., and Popescu, A, Estimation of quality of experience in 3G networks with the M-A. In The Fourth International Conference on Communication Theory, Reliability, and Quality of Service (CTRQ 2011) (Budapest, Hungary), pp. 72–77, (2011).

[28] Hampel, F. Robust statistics: a brief introduction and overview. In Invited talk in the Symposium on Robust Statistics and Fuzzy Techniques in Geodesy and GIS, (Mar. 2001).

[29] Daszykowski, M., Kaczmarek, K., Heyden, Y. V., and Walczak, B, Robust statistics in data analysis – a review: Basic concepts. Chemo metrics and Intelligent Laboratory Systems vol. 85, no. 2, pp. 203 – 219, (2007).