Scalable Wireless Video Streaming over Real-Time Publish Subscribe Protocol (RTPS)

Basem Al-madani, Mohammed Al-saeedi, Anas A. Al-Roubaiey  
Department of Computer Engineering  
King Fahd University of Petroleum and Minerals  
Dhahran, 31261, Saudi Arabia  
{mbasem,g201003900,roubaiey}@kfupm.edu.sa

Abstract—Enabling Real-Time video streaming over wireless networks faces challenges of time-varying channel conditions and the limited network resources. The instability of wireless networks lead to problems such as limited and time-varying bandwidth and traffic congestion when transmitting a burst of video streams. The time-varying wireless channel conditions expose the transmitted video packets to be delayed or dropped. However, in Real-Time video streaming each frame must be delivered and decoded by its playback time. As a result, efficient Real-Time video streaming requires an efficient video quality of service (QoS) transmission control mechanism to adapt to the network changes. Recently, layer coding (LC) enables Real-Time and scalable video streaming to clients of heterogeneous capabilities by dropping upper enhancement layers without the need of re-encoding and with less bit rate. However, layer coding still facing unfair layer protection problem in which packets from the base or lower layers might be dropped while there is a chance to drop packets from the upper enhancement layers. Loosing packets from the base layer can significantly affect the delivered video quality and sometimes lead to an interruption especially in error-prone networks as wireless networks. Architectural solutions at the middleware level introduce higher flexibility, more efficiency in development time and more QoS control. In this paper, we investigate the behaviour of video streaming over Real-Time publish-subscribe based middleware. We propose and develop an unequal layer protection mechanism for Real-Time video streaming based on the Data Distribution Service (DDS) middleware, and show the performance of our approach over IEEE 802.11g WLAN networks. Our approach shows a graceful degradation of video quality while maintaining a robust video streaming free of visible error or interruptions.

I. INTRODUCTION

The flexibility and low infrastructure requirements that wireless networks offer to customers increase its popularity among users. With this popularity, more demands have been recently turned into delivering Real-Time video over wireless networks. However, Real-Time video streaming over wireless networks has been addressed with delivery and communication challenges. The time-varying changes in the wireless channel conditions that can occur due to interference, fading and mobility make video streaming over such networks is a challenging task [1]. Therefore, provisioning of video streaming end-to-end Quality of Service (QoS) is required for maintaining a continuous video playback in Real-Time multimedia applications, e.g., video conferencing. Video streaming is often described as bursty since video is basically a collection of frames sequence transmitted in a particular frame rate. The video frame cannot be decoded or played out at the receiver side until all or most of its transmitted packets are received on time [2]. In Real-Time video streaming, each frame must be delivered and decoded by its playback time. Therefore, any packet that is retransmitted due to lost in transmission or late arriving is considered useless if its decoding and display deadline is too late to be displayed. Real-Time video streaming over wireless networks is difficult because such networks exist to time-varying bandwidth, delay jitter, or high packet loss rate. Reliable delivery of high-quality video over wireless networks which are dealing with unknown and dynamic bandwidth, delay jitter and loss rate is a wide research area in Real-Time video streaming.

The bandwidth available between two points in wireless networks is generally unknown and time-varying. If the sender transmits faster than the available bandwidth then congestion occurs, packets are lost, and there is a severe drop in video quality. If the sender transmits slower than the available bandwidth then the receiver produces sub-optimal video quality. The end-to-end delay that a packet experiences may fluctuate from packet to packet leading to a delay jitter problem. However, the receiver must receive, decode, and display frames at a constant rate, and any late frames resulting from the delay jitter can produce problems in the reconstructed video, e.g., jerks in the video. This problem is typically addressed by including a playout buffer at the receiver. Although, the playout buffer can compensate for the delay jitter, it also introduces additional delay which is not proper for Real-Time application. In addition to the delay jitter, problem as packet loss considers as a fundamental problem in Real-Time video streaming over error-prone networks. Wireless channels are typically afflicted by bit errors or burst errors. The packet-loss problem may lead to serious video quality degradation, which not only affects the quality of current frame, but also leads to error propagation to subsequent frames due to use of the Motion Compensation Prediction (MCP) mechanism [3], [4]. Moreover, a single bit error in a video bit stream with a variable-length coding (VLC) may cause the decoder to lose the synchronization, and consequently the successive correctly received bits become useless [4].

Overcome the time-varying bandwidth problem in Real-Time video streaming requires to accurately estimate the available bandwidth and meanwhile adapt the transmitted encoded video bit rate to the estimated channel bandwidth, and to solve this problem in a multicast situation where a single sender streams data to multiple receivers where each may have a different available bandwidth. Although, several schemes have been proposed to adapt video
encoder bit rate to the available wireless channel bandwidth, or to adapt the wireless channel architecture and protocols to the generated video encoder bit rate, the ability of these schemes to deliver a high quality Real-Time video streaming is limited. Traditional approaches to Real-Time video streaming focused in adapting the video encoder bit rate to the available network resources. Other approaches focused at the level of network protocols and link layer adaptation (e.g., cross-layer approach) to the source video streaming rate. Recently, layer coding (LC) or as formally called Scalable Video Coding (SVC) [5], [6] enables Real-Time and scalable video streaming to clients of heterogeneous capabilities by dropping upper enhancement layers without the need of re-encoding and with a less generated bit rate. However, layer coding still facing unfair layer protection problem in which packets from the base or lower layers might be dropped while there is a chance to drop packets from the upper enhancement layers. Moreover, bit-stream layers are not fully independent since a particular layer requires the presence of all lower layers and the reception of the bit-stream’s base layer is always required for at least decoding the base quality. Thus losing packets from the base layer can significantly affect the received video quality and sometimes lead to an interruption especially in the error-prone networks. Architectural solutions at the middleware level introduce higher flexibility, more efficiency in development time and more QoS control. In this paper, we investigate the behaviour of video streaming over Real-Time publish-subscribe based middleware. We propose and develop an unequal layer protection mechanism for Real-Time video streaming based on the Data Distribution Service (DDS) middleware [7], and show the performance of our approach over IEEE 802.11g WLAN networks. Our approach shows a graceful degradation of video quality while maintaining a robust video streaming free of visible error or interruptions. The rest of the paper is organized as follows. Section II presents an overview background about publish-subscribe communication model. Section III presents the literature review. Section IV presents our proposed scalable RTPS-based video streaming approach. Section V demonstrates the experimental and evaluation results. Finally, our conclusion is drawn in Section VII.

II. REAL-TIME PUBLISH-SUBSCRIBE COMMUNICATION MODEL

The Publish-Subscribe architecture is a data centric design permitting direct control of information exchange among different nodes in the architecture [8]. It connects anonymous information producers with information consumers. The property of decoupling publisher and subscriber in time (data when you want it), in location (publisher and subscriber can be located anywhere) and in platform (connect any set of systems) makes the publish-subscribe communication model more appropriate for large scale and loosely coupled distributed Real-Time systems than the traditional client-server model [9]. Client-server communication drawbacks, e.g., server bottleneck, single points of failure and high bandwidth load in many-to-many communication are resolved by publish-subscribe communication model [10]. Data in publish-subscribe communication model is pushed by the publishers to the subscribers immediately after the data is produced without the need of request, and thus subscribers can access the data in Real-Time. In addition, publish-subscribe architecture frees the data sender (publisher) from waiting for an acknowledgement by the receiver (subscriber), as a result publisher can quickly move on to the next receiver within deterministic time without any synchronous operations which is desirable for a large scale distributed Real-Time systems [9]. Due to these features, publish-subscribe communication model becomes popular in different middlewares, e.g., Java Message Service (JMS), Microsoft Component Object (COM+) and Data Distribution Service (DDS).

DDS is a high performance middlewares standardized by the Object Management Group (OMG) for QoS-enabled publish-subscribe communication aimed at distributed Real-Time and embedded systems [7]. At the core of DDS is the Data-Centric Publish-Subscribe (DCPS) layer that is targeted towards the efficient delivery of the proper information to the proper recipients for applications running on heterogeneous platforms [7]. DCPS builds on a Global Data Space (GDS) by which application’s participants running on heterogeneous platforms can share information by publishing data under one or more topics of interest in the same domain. On the other hand, application’s participants can use the global data space to declare their intent to become subscribers and access data of interested topics. Each topic represents a logical channel for connecting publishers to all interested subscribers. Figure 1 represents the publish-subscribe dissemination of data from one or more publishers to interested subscribers in the DDS middleware. Moreover, DDS is a publish-subscribe based standard with a diverse set of Quality of service (QoS) that guarantee a high performance and low delay of data transmission [7].

III. LITERATURE REVIEW

Real-Time video streaming over wireless networks gains a great amount of research since wireless channel conditions change rapidly due to changing distance between the enabled devices, signal fading, noise interference, and network congestion, leading to time-varying packet loss rate and fluctuating effective bandwidth. Therefore, the provisioning of end-to-end QoS in wireless networks is a very challenging problem. On one hand, the literatures of Real-Time video streaming have concentrated on the avoidance of network congestion since it severely affects the performance of Real-Time video streaming. This is performed by adapting the source video bit rate to the channel bit rate. On the other hand, at the receiver side, if frames packet loss occurs due to unavoidable channel error, many literatures have proposed error concealment and resilience mechanisms for preventing decoders from discarding the entire frame and breaking of the continuity of the video
streams. In this section, we classified the literatures that have been conducted in the Real-Time video streaming area into four major approaches; video rate adaptation approach, error resilient approach, cross-layer design approach, and middleware approach. Video rate adaptation and middleware approaches are going to be discussed since they are more related to our approach.

A. Video Rate Adaptation Approach

Congestion is a common phenomenon in communication networks that occurs when the offered load exceeds the designed limit, causing degradation in network performance such as throughput. Useful throughput can be decreased for a number of reasons. For example, it can be caused by collision in multiple access networks, or by increased number of retransmissions in reliable systems. Besides a decrease in useful throughput, congestion may expose the network traffic to other problems which include packet losses, higher delay and delay jitter. To avoid such undesirable consequences of congestion, control procedures are often employed to limit the amount of network load. Such control procedures are called rate adaptation or congestion control.

In video streaming over wireless networks, a proposal of adapting the bit rate of the encoded video based on verified network status (e.g., available bandwidth) has been widely adopted. Five main approaches have been proposed under the concept of rate adaptation, i.e., rate control, transcoding, bit-stream switching, packet pruning, and scalable coding. Transcoding and bit-stream switching [11] approaches might not be suitable for Real-Time live video streaming. It requires pre-encoded video frames at different rates and quality levels which introduce more delay and not practical for live video show. Transcoding refers to the conversion of one encoding data to another when the receiver does not support the format or has limited storage capacity. However, transcoding is commonly a lossy process; introducing video quality loss. It requires some computational cost and in some cases it might be necessary to decode and then re-encode the content according to end-user restrictions. Some literatures have proposed error resilient transcoding approaches [12], [13], [14] to solve the loss problem in order to be suitable for Real-Time video streaming, but still restricted to the case of prestored videos not a live one.

Unlike transcoding, the idea of the rate adaptation approach is to change the bit rate of the video encoder without changing video formats. This change is according to the negative feedback of the available network resources by using some QoS indicators, e.g., packet loss, packet deadline, etc., while maintaining a reasonable video quality and avoiding any modifications to the network infrastructure. On of the early and widely accepted rate control approaches is the TCP-Friendly Rate Control (TFRC) [15]. TFRC is an equation-based congestion control for unicast multimedia traffic based on the TCP Reno's throughput equation. In TFRC, the sender adjusts its sending rate as a function of the measured rate loss, where a loss consists of one or more packets dropped within a single round-trip time. However, TFRC is mainly proposed for wired networks especially for the Internet and when applied to the wireless networks, it suffers from performance degradation [16]. This is because TFRC method assumes perfect link quality and considers the network congestion as the only packet loss reason while most of packet loss in wireless networks is due to error at physical layer. As a result, literatures such as [16], [17], [18], [19], [20], [21], [22] proposed a new optimized TFRC-based mechanisms to support error-prone networks as wireless networks. In addition, authors in [23], [24], [1], [25], [26], [27], [28] proposed different rate control schemes to adapt video stream to the available network resources. As instance [1] proposed an algorithm for verifying network status by using MAC-layer parameters implying PHY-layer information and then correspondingly adjusting the target bit rate. Despite the success of the rate control approach in Real-Time video streaming, it may fail under the multicast video streaming scenarios over multi-rate wireless LANs. In multicast video streaming, rate control mechanism requires to send a specific video transmission rate for every user of different wireless network channel conditions. This may involve the estimation of every users channel resources and the generation of multiple video transmission bit rates which introduce an overhead.

Recently, the idea of multi-layer scalable video stream start to be dominant in the field of video streaming. Unlike the single-layer rate adaptation approaches (e.g., rate control, transcoding, bit-stream switching), multi-layer video stream consists of a base layer and other enhancement layers which are independent of each other. That is, dropping an enhancement bit-stream layer will not severally affect the whole quality of the decoded video. Scalable video streaming overwhelms non-scalable (single-layer) video streaming in heterogeneous and error-prone networks. It enables data rate adaptation without re-encoding or transcoding, but only by dropping bit-stream packets. This property eliminates the overhead of transcoding and bit-stream switching to adapt video rate to the available network resources. The Scalable Video Coding (SVC) H.264/SVC is an implementation of multi-layer scalable video stream. H.264/SVC provides network-friendly scalability at a bit-stream level with a moderate increase in decoding. Scalability in video streaming is supposed to provide functionalities such as bit rate, format, and power adaptation to the varying terminal capabilities or network conditions. In H.264/SVC, scalability refers to the removal of parts of the video bit stream in order to adapt it to the various preferences of end users as well as to varying terminal capabilities or network conditions [29]. However, scalable video coding as H.264/SVC still requires end-to-end QoS for maintaining the priority of which frames packets of which layer are supposed to be dropped first in order to control video traffic congestion and deliver better video quality [30]. Forward Error Correction (FEC) mechanism also proposed for the scalable video streaming by allocating different amount of FEC codes to different layers according to their priority to achieve graceful degradation [31], [32], [33]. In [32] an unequal layer error protection has been proposed in a DVB-H transmission of layered video on response to packet losses. A research work as in [33] proposed an unequal error protection for SVC base and enhancement layers while considering the transmission of on-demand scalable variable-bit-rate (VBR) video and the existence of receiver playback video buffer. However, our research work is different in which it consider the Real-Time video transmission. On the other hand, a good amount of researches have been conducted in a cross-layer schemes as a proposed solution for delivery of scalable video over multirate wireless networks as 802.16 or IEEE 802.11 [34], [35], [36], [37], [38]. As instance in [34], authors have proposed a cross-layer design to optimize the
link adaptation scheme that configures the PHY and MAC layers, and treat SVC enhancement layers differently in a way that the highest possible video quality is achieved by avoiding dropping layers and without adding to the traffic load of the WLAN. Link adaptation optimization is used to determine the number of video layers permitted, and the PHY transmission mode assigned to each video layer. However, such proposed approaches are complex and only exclusive for those wireless networks which employ a variable rate PHY and a link adaptation mechanism as in 802.11n multi-input multi-output (MIMO), 802.11a and 802.16. Some other literatures proposed a Real-Time and scalable delivery of SVC-based video over MIMO networks [39], [40].

B. Middleware Approach

There is a deficiency in the literature related to video streaming over middlewares. In this section we summarize the most related existing works. Detti et al., [41] evaluated and demonstrated a technique for streaming H.264 SVC video over a DDS middleware. The structure of the DDS data unit designed by them was able to carry H.264 SVC video units. Also they designed a receiver-driven rate-control mechanism based on the DDS data unit and exploiting specific DDS functionality. Finally, they implemented and showed the effectiveness of their mechanism in an 802.11 wireless scenario, comparing their proposal with other solutions. Clavijo proposed that a CORBA middleware implementation can be used to offer Real-Time video streaming [42]. Furthermore, in his paper [42], Karr et al stated that a CORBA based platform was introduced to respond to changing resource requirements in video applications using video streaming service over CORBA-based solution which has been the one projected for Real-Time environment. CORBA is a very complete technology that introduces a big number of interfaces for almost any type of middleware functionality; however, CORBA is a complex architecture that introduces implementation overheads, in particular if compared with other lighter weight technologies such as ICE (Internet Communications Engine) [43], DDS (Data Distribution Service for Real-Time systems) [7], or some specific Real-Time Java based solutions [44]. Therefore, existing approaches can be improved to offer appropriate support to the real time nature of video transmission with guarantee. In addition, using new standard middleware introduces flexibility for video transmission at two approaches. First, compared to direct implementation over the network level, the utilization of a middleware is already more flexible. Second, utilizing middleware solution offers QoS management which allows to appropriately initiating Real-Time support to video transmission.

IV. DESIGN & IMPLEMENTATION OF SCALABLE RTPS-BASED VIDEO STREAMING

This section covers the architecture and implementation of the proposed scalable video streaming system over Real-Time Publish Subscribe Protocol (RTPS). The RTI Data Distribution Service (RTI-DDS) [7] is used for the implementation of the system.

Fig. 2. Scalable RTPS-Based Video Streaming Architecture.
from DataWriters to DataReaders but with the cost of no guarantee to receive data. Best effort delivery is suitable for Real-Time video streaming over loosely network such wireless since it is more efficient and losing some video frames will affect the quality of the encoded video but not corrupting the whole video data.

- **History**: controls how the system manages frames payload sent by a publisher’s DataWriter or received by a subscriber’s DataReader. It help tunes the reliability between publishers and subscribers. In Real-Time video streaming reliability is not a matter, so no need for keeping history of the recent published data for retransmission. The history QoS is set to KEEP_LAST value with one picture for publisher’s DataWriters and KEEP_LAST value with one Group of Pictures (GOP) of depth for the subscriber’s DataWriters. The decoder at the subscriber side some times needs previous frames for decoding reference frames.

- **Durability**: controls whether or not new subscribers get data which was published by publisher’s DataWriters previously to increase system tolerance to failure conditions. It is obvious that in live video streaming, the new joining participants should follow the live show while the previous show events are useless. However, decoding frames like reference P or B frames are based on prediction compensation algorithms. For example, P frame can only be decoded with reference information from previous I or P frames. In addition, B frame can only be decoded with reference information from the previous and successive I or P frames. Due to this fact, durability QoS is set to TRANSIENT_LOCAL value which means that the frame which has been already sent may be relevant to late-joining subscribers (subject to any history depth, lifespan, and content or time-based filters defined). Data will be cached with the DataWriter that originally produced it.

- **Partition**: provides another way to control which DataWriters will match and communicate with which DataReader. Normally, DataWriters are matched to DataReaders of the same Topic. However, by using the partition QoS policy, additional criteria is used to decide if a DataWriter’s data is allowed to be sent to a DataReader. One or more strings can be added to the DataWriter’s publisher or DataReader’s subscriber parent in such case the DataWriter is only matched to a DataReader for the same Topic only if their publisher and subscribers have a common partition.

Partition QoS has some key features that play a main role in our proposed RTPS-based Real-Time video streaming architecture. First, subscription to a certain video partition be dynamically changed at run-time. This is used in our proposed scalable architecture to quickly control the subscriber’s DataReader to begin receive lower video quality from another DataWriter when a network degradation is introduced. Second, partition QoS policy can dynamically configure the connection topology without stopping/starting or destroying/re-creating publishers, subscribers or even a participant. As a result, there is no spawning and killing of threads or allocation and deallocation of memory when publishers and subscribers add or remove themselves from partitions. This property is appropriate to Real-Time application since keeping a low latency is a critical issue. In our proposed scalable architecture, every sublayer video is assigned to a certain video partition. By default, the video subscriber subscribes to the highest video partition, so it reads video stream from the matched DataWriter of a high quality video. It adaptively subscribes to a lower video partition (lower video quality) when network degradation is notified (e.g., low bandwidth, congestion, etc.). Thus, subscriber’s DataReader immediately received video frames from the matched and proper publisher’s DataWriter.

- **Deadline**: deadline period is set to a specific value that estimates when a frame should be received at the subscriber side. It is also used as an indicator of the network performance degradation. When frames fail to reach within the estimated deadline period, it means that the system is performing improperly and the subscriber should switch to begin receive video of lower quality from another publisher DataWriter’s partition.

<table>
<thead>
<tr>
<th>Publisher</th>
<th>Reliability</th>
<th>History (depth = 1)</th>
<th>Durability</th>
<th>Partition</th>
<th>Time Based Filtering</th>
<th>Lifespan</th>
<th>Presentation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BEST_EFFORT</td>
<td>KEEP_LAST (depth = 1)</td>
<td>TRANSIENT_LOCAL</td>
<td>&quot;video.L&quot;/&quot;video.L&quot;</td>
<td>Not applicable/Not applicable</td>
<td>Playback deadline/Not applicable</td>
<td>Publish order</td>
</tr>
<tr>
<td>Subscriber</td>
<td>BEST_EFFORT</td>
<td>KEEP_LAST (depth = GOP)</td>
<td>TRANSIENT_LOCAL</td>
<td>&quot;video.L&quot;/&quot;video.L&quot;</td>
<td>&lt;Playback deadline/&gt;Not applicable</td>
<td>N/A/N/A</td>
<td>N/A/Publish order</td>
</tr>
</tbody>
</table>

| R × O | Y | N | Y | N | N/A | N/A | Y |

- **Time Based Filtering**: time based filtering QoS policy controls the rate of data samples that should be delivered to a DataReader within the permitted deadline. Data samples for a DataReader can be filtered out using the TIME_BASED_FILTER QoS by setting the minimum separation time. Once a data sample for an instance has been received, the middleware will accept but drop any new data samples for the same instance that arrives within the time specified by minimum_separation. Minimum separation time should be less than the deadline time. Time based filter QoS allows receiving the data samples within a period of time (deadline) but after the time specified by minimum_separation, see Figure 3. In our proposed architecture time based filter QoS is used to optimize resource usage (CPU and possibly network bandwidth) by only delivering the required amount of video samples (NALs) to different DataReaders, and filtering out samples that arrive faster than a specified rate (period of time between video samples arrival).

- **Lifespan**: lifespan QoS Specifies how long the system
should consider data sent by a publisher to be valid. It is used to timestamp all data sent and received. In our proposed video streaming system, lifespan QoS is set to a period equal the frame playback time. The DataReader’s receive queue will be checked out to see how long the frame’s packet/s has been stored before playout by comparing their timestamp to the current time. Video sample that has exceeded its lifespan duration will be removed from the DataReader’s receive queue. Therefore, ensure the system doesn’t receive or act on video data that are too old and have expired.

- **Presentation:** In video streaming applications, frames or video samples should be retrieved in the same order as originally have been sent. The Presentation QoS is used in our proposed RTPS-based Real-Time video streaming architecture to guarantee that the video NAL units are retrieved by the publisher’s DataReader as originally were sent by the publisher’s DataWriter.

### B. System Scalability and Behaviour

Scalability in our proposed approach means the ability of the system to adaptively serve different video subscribers (clients) with the appropriate video quality that is proportional to the time-varying wireless channel conditions (e.g., time-varying bandwidth), or limited computational resources. The approach uses the Data Distribution Service (DDS) [7] middleware which contains a built-in Real-Time Publish Subscribe Protocol (RTPS) in order to stream Real-Time video. We adopted the passive or non-intrusive technique for estimating the available bandwidth and user’s capabilities. Based on the existence traffic in the network we use the rich DDS data QoS to estimate the potential congestion and packet loss occurrence and thus control the transmitted video NAL units packets. Two scalability architectures are proposed in our work: one for the single-layer video streaming AVC and the other for multi-layer video streaming SVC. Re-encoding, retransmission, pre-encoded bit-stream switching and transcoding are not considered since our scope is the scalable streaming of live video through an error-prone loosely networks as IEEE 802.11. Our proposed scalability mechanism is based on unequal packet jitter and loss protection by dropping the less important video NAL’s packets depending on the frame type and layers for which they are belong to. Single-layer coded video stream as in H.264/AVC can be treated as a temporal scalable video stream, if the encoded bit-stream has the correct properties. For example, have a correct temporal scalable video picture sequence. That is, reference pictures as I and P frames both are considered as the base layer while the subsequence of the hierarchical B frames are considered as the next enhancement layers. Figure 4 shows our proposed temporal scalability for single-layer video stream using the DDS middleware over wireless networks. Every encoded frame is assigned in the publisher side to one partition P or more depending on the temporal available scalable layers (see partition QoS). By default the subscriber side subscribes to the whole temporal layers which are assigned to the higher partition $P_n$ of topic V. Every partition in the publisher side has a DataWriter (DW) with a certain QoS. All DWs have the same Best-Effort reliability QoS to maintain a fast transmission for Real-Time video streaming. In the subscriber side, only one built-in DataReader is considered and using the partitioning QoS to switch among different temporal streams. When the subscriber side DR’s deadline Qos detects NAL’s packets exceeding their playout deadline in a history buffer of GOP size, it directly switches its subscription to another partition $P = P_{n-1}$ without the need to send a feedback traffic. The only partition traffic that is supposed to be transmitted is the one that the subscriber is subscribed to its matched topic plus the partition name. Figure 5 shows an example of encoded single-layer video with a temporal scalability property. A source video of a CIF resolution (352×288) is encoded to AVC stream of $(IBBBBBBBI...)$ GOP sequence. In this example, the group of picture size is 8 and the bit-stream consists of hierarchial B-pictures. The lowest temporal $T^0$ which only contains I-frames $(II....)$ can be considered as the temporal base layer and assigned to the lowest partition $P^0$ while the current temporal with the next one which contain I and P frames $(IPI....)$ are considered as the next enhancement layer and assigned to partition $P^{0+1}$ till the last partition $P^3$ which contains the whole frames $(BBBBBBBBI...)$... The result will be four partitions for streaming four different temporal streams. The same idea for scalable (multi-layer) video stream SVC but with supporting various scalabilities, i.e., temporal, spatial, and quality scalabilities. Beside the scalability nature of the SVC video stream, our proposed RTPS-based video streaming scalability gives more control one which layer should be unsubscribed first using the DDS rich QoS without the need for relay node or dropping packets at the receiver.

![Fig. 3. Time Based Filter QoS [46].](Image)

![Fig. 4. RTPS-based video streaming of single-layer video stream with temporal scalability of H.264/AVC.](Image)

![Fig. 5. Single-layer video stream partitioning example.](Image)
Fig. 6. RTPS-based video streaming of multi-layer video stream with combined scalability of H.264/SVC.

Fig. 7. Multi-layer video stream partitioning example.

side. As shown in Figure 6, every encoded SVC sublayers are assigned to a certain partition. Figure 7 shows an example of multi-layer encoded video with a combined (temporal, spatial and quality) scalability property. In this example, the lowest two sublayer are both with the same temporal and spatial (352×288) properties while the quality is changed from $Q^0$ to $Q^1$. Also, the next two sublayers are both with the same temporal and spatial (176×144) properties but differ than the first two sublayers and with different qualities $Q^0$ to $Q^1$ same as the first two sublayers.

V. EXPERIMENTAL SETUP & PERFORMANCE EVALUATION

In this section we present performance and scalability results obtained from transmitting a Real-Time video stream over the Data Distribution Service (DDS) middleware by using our proposed scalable RTPS-based Real-Time video streaming system architecture through 802.11g WLAN.

A. Experimental Setup

The experimental setup involves five nodes A to E with a wireless adapter of 54Mbps in each one connected with an access point of 54Mbps channel capacity, see Figure 8. The experiment setup is performed indoor to study the affect of transmitting live video in such ordinary condition. A source video sample encoded by a single-layer video encoder (H.264/AVC) is transmitted over our proposed scalable RTPS-based Real-Time video streaming system architecture through 802.11g WLAN.

TABLE II. SOURCE VIDEO SAMPLES.

<table>
<thead>
<tr>
<th>Video sample name</th>
<th>Format</th>
<th>Number of frames</th>
</tr>
</thead>
<tbody>
<tr>
<td>foreman</td>
<td>CIF (352 × 288)</td>
<td>300</td>
</tr>
<tr>
<td>foreman</td>
<td>QCIF (176 × 144)</td>
<td>300</td>
</tr>
</tbody>
</table>

addition to the proper video streaming and coding/decoding tools.

1) Evaluation Framework: The Evaluation Video (EvalVid) tool-set [47] is used for evaluating our proposed scalable RTPS-based video streaming architecture. EvalVid enables networking operatives to evaluate the effects of real video streams on proposed network protocols. It basically evaluates the delivery quality of the video transmission in a real or simulated network environment. One of the drawbacks of EvalVid is that it only supports single layer video codec like H.264/AVC. That is, a scalable video codec like H.264/SVC is not supported. Fortunately, EvalSVC [48] comes to fill this gap and enabling the evaluation of a scalable video coding. It is capable of evaluating the enhanced features such as: spatial, temporal, SNR, and combined scalability of SVC bit-streams transmitting over real or simulated networks. Thus, we could use both evaluation tool sets to evaluate our proposed scalable RTPS-based video streaming system.

2) Source video sample: The commonly used foreman video test sequence in the 4:2:0 format is used to evaluate our scalable RTPS-based video streaming system. Two foreman video sequences of different Common Intermediate Format (CIF) and frame size are used as shown in Table II.

3) Performance measurement metrics: Different QoS measurement metrics of the network such as end-to-end delay, jitter, loss rate, sender’s and receiver’s bit-rate are going to be measured in order to see the performance of our proposed system for streaming video over 802.11g WLAN networks. In addition, the Peak Signal Noise Ration (PSNR) video quality measurement metric is going to be measured.

Frame end-to-end delay involves the one-way delay at the source encoder, channel transmission and propagation delay, and source decoder at the receiver endpoint. In our experiments the encoding and decoding delay (processing delay) have been excluded since they are out of our scope. Encoding and decoding processes are performed separately and in non-realtime. Therefore, the measured end-to-end delay is only for the channel transmission and propagation delay of every successive transmitted frame within its playback time constraint from the publisher (sender) endpoint to the subscriber (receiver) endpoint.
Frame jitter refers to the variability over time of a series of frame one-way frame latency measurements across the network caused by network congestion, time varying network bandwidth, interferences, etc. Frame jitter severely affects the quality of streaming video. A network with constant latency has no variation (or jitter). Packet jitter is expressed as an average of the deviation from the network mean latency. Packet loss can be caused by a number of factors including signal degradation over the network channel due to multi-path fading, packet drop because of channel congestion. Frame loss rate refers to the percentage loss of frames that are dropped by such factors or intentionally by the proposed solution for scalability purpose. PSNR stands for Peak Signal Noise Ration. The PSNR computes the peak signal-to-noise ratio, in decibels, between two images. This ratio is often used as a quality measurement between the original and a compressed image. The higher the PSNR, the better the quality of the compressed, or re-constructed image.

B. Performance Evaluation

For evaluating our scalable RTPS-based video streaming system over error-prone wireless networks, the H.264/AVC encoder is used for producing a temporal single-layer scalable video bit-stream and then transmits this stream by both RTP/UDP and RTPS/UDP protocols separately in real-time to show the performance and scalability of our proposed system. The source video file (foreman) of YUV format in CIF resolution (352×288) and 300 frames is encoded by using X264 [49] encoder into 30 frames per second and a GOP length of 8 frames. The first frame is intra-coded IDR frame and represents a special GOP while every other GOP consists of a key frame followed by a hierarchically predicted B-frames. The number of temporal scalability levels that can be generated is dependent on the specified GOP size. In this experiment, the generated bit-stream provides 4 temporal scalability levels as shown in Table III. The encoded bit-stream does not provide several spatial resolutions or several bit-rates for a specific spatio-temporal resolution. The encoded video bit-stream is hinted by MP4 container using MP4Box tool of GPAC [50] framework in order to packetize the frames for the transport with RTP. The maximum transmission packet unit is assigned to 1024 byte.

For transmitting the encoded video bit-stream over RTP/UDP, the mp4trace tool from EvalVid is used to send the hinted mp4 file to a unicast and multicast destination IP in three different scenarios; one publisher to one subscriber, 6 subscribers and 12 subscribers. On the other hand, our RTPS-based video streaming system is used to transmit the same encoded and hinted video sample over RTPS/UDP protocol using the same scenarios.

<table>
<thead>
<tr>
<th>Layer</th>
<th>Resolution</th>
<th>Frame Rate (fps)</th>
<th>Bit Rate</th>
<th>DTQ</th>
<th>In Partition</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>352×288</td>
<td>3.7500</td>
<td>182.4971</td>
<td>(0.0,0)</td>
<td>P₀, P₁, P₂, P₃</td>
</tr>
<tr>
<td>1</td>
<td>352×288</td>
<td>7.5000</td>
<td>230.3008</td>
<td>(0.1,0)</td>
<td>P₁, P₂, P₃</td>
</tr>
<tr>
<td>2</td>
<td>352×288</td>
<td>15.0000</td>
<td>276.8912</td>
<td>(0.2,0)</td>
<td>P₂, P₃</td>
</tr>
<tr>
<td>3</td>
<td>352×288</td>
<td>30.0000</td>
<td>327.0816</td>
<td>(0.3,0)</td>
<td>P₃</td>
</tr>
</tbody>
</table>

In each scenario, the tcpdump network monitoring tool is used to trace the IP packets at the sender and receiver during the transmission process. The result are sender and receiver trace files to be used later by EvalVid. In every experimental scenario five main files is used for the evaluation purpose; the YUV source file before and after the encoding, the encoded and encapsulated mp4 video file, the sender and receiver UDP packets trace file, and the transmission sender trace file which contains information about the frame type, packet size, packet space (segmentation), and transmission timestamp in milliseconds.

To obtain the PSNR values we compare the encoded video at the sender side with PSNR values of the receiver side. The results show that RTP and RTPS have nearly the same performance in one-to-one unicast scenario as shown in Figure 9. However, RTPS shows a lower but stable (no interruptions) video quality when the numbers of receivers (subscribers in RTPS) increased as shown in Figures 10 and 11. This is because subscribers are dynamically switching among video partitions when network degradation has been detected in our scalable RTPS-based video streaming system. Figure 16 shows that RTPS has a better frame end-to-end delay performance than RTP especially when the number of participants are increased. In Figures 15 and 17 cumulative frame jitter of our RTPS-based system also shows a better performance; the frame end-to-end variations are very small in comparing to RTP. The
Fig. 12. Frame End-to-End delay (1 to 1).

Fig. 13. Cumulative jitter (1 to 1).

Fig. 14. Frame End-to-End delay (1 to 6).

Fig. 15. Cumulative jitter (1 to 6).

Fig. 16. Frame End-to-End delay (1 to 12).

Fig. 17. Cumulative jitter (1 to 12).

VI. CONCLUSION

Real-Time video streaming over wireless networks faces challenges of time-varying packet loss rate and fluctuating bandwidth. Frames must be delivered and decoded by its playback time. Recently, layer coding (LC) enables Real-Time and scalable video streaming to clients of heterogeneous capabilities by dropping upper enhancement layers without the need of re-encoding and with less bit rate. However, layer coding still facing unfair layer protection problem in which packets from the base or lower layers might be dropped while there is a chance to drop packets from the upper enhancement layers. Loosing packets from the base layer can significantly affect the delivered video quality and sometimes lead to an interruption especially in error-prone networks as wireless networks. In this paper, we investigate the behaviour of video streaming over Real-Time publish-subscribe based middleware. We propose and develop an unequal layer protection mechanism for Real-Time video streaming based on the Data Distribution Service (DDS) middleware, and show the performance of our approach over IEEE 802.11g WLAN networks. Our approach shows a graceful degradation of video quality while maintaining a robust video streaming free of visible error or interruptions.

ACKNOWLEDGMENT

The authors would like to acknowledge the support provided by the King Fahd University of Petroleum and Minerals (KFUPM).

REFERENCES


reconstructed video is almost clear of significant jerks.


