Hindawi Publishing Corporation
Advances in Multimedia
Volume 2014, Article ID 164940, 11 pages
http://dx.doi.org/10.1155/2014/164940

Research Article

Real-Time QoS-Aware Video Streaming: A Comparative and Experimental Study

Basem Al-Madani, Anas Al-Roubaiey, and Zubair A. Baig

Department of Computer Engineering, King Fahd University of Petroleum & Minerals, Eastern Province, Dhahran 31261, Saudi Arabia

Correspondence should be addressed to Anas Al-Roubaiey; roubaiey@kfupm.edu.sa

Received 23 September 2013; Accepted 10 December 2013; Published 16 January 2014

Academic Editor: Costas Kotropoulos

Due to its flexibility, scalability, real-time, and rich QoS features, Data Distribution Service (DDS) middleware provides seamless integration with high-performance, real-time, and mission-critical networks. Unlike traditional client-server communication models, DDS is based on the publish/subscribe communication model. DDS improves video streaming quality through its efficient and high-performance data delivery mechanism. This paper studies and investigates how DDS is suitable for streaming real-time full-motion video over a communication network. Experimental studies are conducted to compare video streaming using a the VLC player with the DDS overlay. Our results depict the superiority of DDS in provisioning quality video streams at the cost of low network bandwidth. The results also show that DDS is more scalable and flexible and is a promised technology for video distribution over IP networks where it uses much less bandwidth while maintaining high quality video stream delivery.

1. Introduction

Video streaming applications are experiencing fast growth and demand for diverse business needs. Applications of video streaming include, for example, commercial applications such as e-learning, video conferencing, stored-video streaming; and military applications such as video surveillance of targeted field or specific objects. Video traffic is resource intensive and consumes a lot of network bandwidth; therefore it is challenging issue to stream video over limited-bandwidth networks, for example, WSN or Bluetooth. In many cases, bandwidth usage implies direct cost on end-users. In this work, we try to enhance the end-user experience both in terms of quality and cost, through the deployment of the DDS middleware.

1.1. DDS Overview and Video QoS Polices. DDS stands for Data Distribution Service. It is a set of specifications standardized by the Object Management Group (OMG). The DDS middleware is a known standard with built-in data-structures and attributes specified by meta-information called topics. Every topic describes a set of associated data-samples with the same data-property and data-structure. For example, a topic named "temperature" can be used to store samples of temperature monitored by a distributed set of sensors [1].

The entities that write and read the data-samples using a DDS-based middleware are the publishers and the subscribers. A publisher consists of a set of data writer modules, each of which is used to write information on a particular topic. On the other hand, a subscriber reads the data samples of topics by using its data reader modules. A topic is qualified by a wide set of Quality of Service (QoS) parameters that manage a number of aspects of the distribution of linked data-samples.

For instance, the "lifespan" QoS parameter computes the maximum time a data-sample can stay within the system from time of inception of writing. The "history" QoS specifies the maximum number of data-samples that can be stored in the middleware; if this maximum number is reached, then the newest data-sample substitutes the oldest one. When an application requires data-samples on a particular topic, it simply feeds the DDS interface with the name of the topic. The DDS middleware does the setup of the underlying networking resources for data delivery.

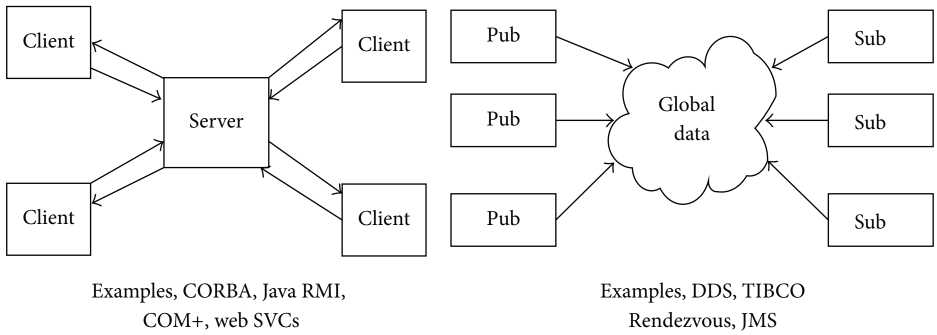

Figure 1: Publish subscribe versus client server architecture.

This buffering function of History QoS is beneficial for video streaming where the late joining participants can still view previously delivered video. It is also worth mentioning that an application can determine a filtering-condition correlated to the content of data-samples, for example, temperature measurement less than 20 degrees. In this case, the DDS transfers only data-samples complying with the filtering condition. This filtering is very useful in many video transmission scenarios; for example, for safe web browsing, we may use content based filters to delete undesired frames. Another important QoS is the RELIABILITY QoS, which has two main values RELIABLE and BEST_EFFORT. For real-time applications such as video transmission, RELIABILITY QoS is set to BEST_EFFORT specially in case of real-time video streaming where retransmission significantly affects the video playback. DDS also supports reliable QoS for data sensitive applications such as FTP through the RELIABLE. In video streaming, presumably new values for the samples are generated often enough that it is not necessary to resend or acknowledge any samples [2].

There are two approaches for establishing communication between heterogeneous systems. The first one, *publish/subscribe*, is a messaging pattern where senders of messages, called publishers, do not program the messages to be sent directly to specific receivers, called subscribers. Instead, published messages are classified into several topics with unique identifications, without knowledge of what, if any, subscribers there are. Similarly, subscribers express interest in one or more classes (topics) and only receive messages that are of interest, without knowledge of what, if any, publishers there are; see Figure 1. Due to its loosely coupled property, *publish/subscribe* architecture is more flexible and scalable for distributed systems; in our work, this architecture is represented by DDS standard.

The second approach is the traditional *client/server* pattern, which is a tightly coupled pattern where the programmer here should specify the clients and servers addresses and they have to work at the same time. That makes it less scalable than publish/subscribe pattern. In brief, Corsro [3] summarizes the advantages of publish/subscribe over client/server architecture as publish/subscribe is plug and play, loosely coupled, fault resilient, and inherently many-to-many architecture, whereas client/server architecture is complex in development, tight coupling, fragile to fault, and inherently one-to-one architecture. The client-server pattern is represented in this paper using VLC player.

1.2. Contributions. The main contribution of this paper is to examine the behavior of real-time video streaming over both publish subscribe and client server architectures using the DDS middleware and the VLC player. To the best of our knowledge, this is the first attempt to examine and contrast these scenarios where we concentrate on examining the total bandwidth consumed and video quality of video traffic by evaluating network throughput, packet loss, and jitter with different network load, number of subscribers. Furthermore we demonstrate the most important quality of service performance parameters of DDS and describe how these can be configured to improve video transmission quality over networks.

Our results prove that DDS is very much applicable and a promising technology for video streaming. Its key features such as platform independent, reliability, and scalability help it significantly improve the quality of stored video streaming over heterogeneous platforms. In addition, we also discuss the utilization of DDS QoS to maximize application performance.

1.3. Paper Organization. The paper is organized as follows. Section 2 presents the literature review. Section 3 gives a background on DDS QoS and VLC and describes how DDS QoS can be used to improve networked video streaming. In Section 4, we demonstrate the experimental work and conduct a results analysis. Finally, conclusions and future work are discussed in Section 5.

2. Literature Review

In this section, we summarize the previous work in the literature for enhancing video streaming over wireless networks.

Detti et al. [1] 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 [4] video-units. Also they designed a receiver-driven rate-control mechanism based on the DDS data unit, which exploited 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 et al. proposed that a CORBA middleware implementation can be used to offer real-time video streaming [5]. Furthermore, in [6], Kaff et al. introduced a CORBA based platform to respond to changing resource requirements in video applications using video streaming service. CORBA is a very complete technology that introduces a big number of interfaces for almost any type of required middleware functionality; however, it is a complex architecture that introduces implementation overheads, particularly when compared with other lighter weight technologies such as ICE (Internet Communications Engine) [7], DDS (Data Distribution Service for real-time systems) [2], or some specific real-time Java based solutions [8]. Therefore, existing approaches can be improved to facilitate real-time video transmission with guaranteed QoS. In addition, using new standard middleware introduces flexibility for video transmission in two ways. First, compared to direct implementation over the network level, the utilization of a middleware is already more flexible. Second, utilizing middleware solutions provides QoS management to appropriately initiate real-time and QoS-aware support for video transmission.

Vora and Brown [9] studied DDS deployment for the newer 802.11n standard. Their performance metrics were throughput, delay, and jitter when video streaming is brought in a network carrying merely data traffic. They also studied the approximate number of users streaming high rate videos that can be supported over various network configurations. In [10], the authors analyzed and evaluated the performance of H.264-based video streaming over multihop wireless local area networks (WLANs). Guidance was provided on how to achieve the optimal balance for a given scenario, which is important when deploying end-to-end video streaming services with quality of service guarantees. For WLANs, we have conducted a previous study to examine DDS over WLANs [11], but the video that we used was very low motion video and the codec bit-rate was 128 kbps which is much less than what is used in this paper. That adjustment was done to meet the limited WLAN bandwidth; the results showed that consumed bandwidth was nearly twice less than that in the proposed work.

Chen and Zakhor proposed several TFRC connections as an end-to-end rate control solution for wireless video streaming. They showed that this approach not only avoids modifications to the network infrastructure or network protocols, but also results in full utilization of the wireless channel [12]. Stockhammer et al. proposed that the separation between a delay jitter buffer and a decoder buffer is in general suboptimal for video transmitted over VBR channels [13]. They specified the minimum initial delay and the minimum required buffer for a given video stream and a deterministic VBR channel.

In [14], Nasser proposes QoS adaptive multimedia service models for controlling the traffic in multimedia wireless networks (MWN) for cellular networks. The suggested framework is designed to take advantage of the adaptive bandwidth allocation (ABA) algorithm with new calls in order to improve the system usage and blocking probability of new calls. Simulation results showed that the QoS adaptive multimedia service framework outperforms the existing framework in terms of new call blocking probability, handoff call dropping probability, and bandwidth utilization.

Li and Pan [15], through their study in a WDS-based multihop wireless environment, found out that it is likely for multihop wireless networks to increase coverage and sustain improved video streaming performance at the same time. When they analyzed the throughput of IEEE 802.11 multihop wireless networks, they proposed a complete two-dimensional Markov-chain model in their paper. The model considered the retry bound and post-back off step into account to better capture the performance of IEEE 802.11 MAC protocols in a non-ideal channel and with non-persistent traffic. The throughput analysis is validated by network simulation with extended lower and upper-layer simulation modules. The achievable throughput gives an upper bound of the video streaming performance, which is further validated by our H.264-based video streaming simulation with application-layer performance metrics (provided in subsequent sections). The results correspond to the observation they had on the multihop test bed. Another study [16] highlights that since the advent of ad hoc networks, it has been viewed as a potential multiapplication technology. This paper presents a comparative study of multicasting of video and video-like data using two different ad hoc routing protocols, namely, OLSR and PUMA. Their NS2 simulations show that OLSR produces higher throughput and lower latency.

A cross-layer solution for video streaming QoS support has been proposed recently in [17]. This work focused on low bandwidth networks. The authors evaluated the feasibility of transmitting streaming video flows by using the Contention Free Period (CFP), considering the necessary coordination between the CFP period of the IEEE 802.15.4 MAC standard and the real features of the wireless medium as well as the limitations of the sensor electronics and their power-consumption. This coordination is performed through two steps. (1) The first one is the generation of safety time gaps in the MAC frame. These gaps are created for ensuring that delayed frames arrive at the sink on time and avoiding collisions with other frames; (2) The second one is the design of a distributed protocol, developed at the application layer, that allows to measure and calculate several metrics (QoS parameters) such as frames or images delay, video throughput, and the subjective impression perceived by the users when they receive the video sequence. The subjective perception is measured by the Peak Signal-to-Noise Ratio (PSNR) and Mean Opinion Score (MOS) values. From the analysis of these metrics, the proposed protocol controls the optimum MAC gap sizes, the available video transmission rate, and the minimum power consumption of the WPAN network nodes. This protocol is denoted as Cross-Layer Multimedia Guaranteed Time Slot (CL-MGTS), since it uses application-level QoS parameters to tune the MAC and physical layers. However, their solution is tightly coupled to the IEEE 802.15.4 MAC layer protocol which makes it importable; DDS has a resource management QoS that could be used to control such types of networks.

One of the latest proposals on enhancing QoS support for video transmission is by Huang et al. [18]. They improved the quality of video transmission by proposing a multipath technique that extends the Datagram Congestion Control Protocol (DCCP), which is an unreliable transport layer protocol with a congestion control mechanism used for multimedia streaming. In order to use multiple network interfaces to transmit streaming data smoothly, a Multi-Path Datagram Congestion Control Protocol (MP-DCCP) is proposed and presented in this work. Video streaming transmission through multipath faces three problems: (1) out-of-ordering packets at the receiver side, (2) conditions of paths that are changed anytime, and (3) the importance of frames/packets that is different. The first problem may let video streaming data be delivered to the application layer too late, especially for the live streaming data. The second and third problems may let a packet be scheduled to an unsuitable path for transmission. Since the importance of video streaming frames/packets is different, it should consider how to schedule the transmission of frames/packets properly. For example, an important frame/packet should be transmitted through the more reliable path. In order to resolve these three problems, a QoS-aware Order Prediction Scheduling (QOPS) scheme for MP-DCCP is also proposed. QOPS estimates packets' arrival orders at the receiver side through these multiple paths before packets are scheduled into the transmission at the sender side. From simulation results, the authors show that the out-of-ordering problem of packets in MPDCCP can be countered using the proposed QOPS scheme.

Another recently proposed technique to improve video transmission quality using tightly coupled solutions is found in [19]. The authors proposed a new multirate H.264 scalable video multicast in lossy networks using network coding. They first prioritize video layers based on its effect on the end-to-end video quality. Each video layer is routed via the path obtained from the optimization framework under the constraints on QoS guarantees. Different destinations may receive different number of video layers that depended on their max flows. The bottleneck in the network is resolved by using network coding to ensure that all destinations receive the rate equaling their max flows. The network coding is only applied within the same layer. Simulation and numerical results under randomly generated networks show the advantage of the proposed scheme in terms of objective and subject qualities of the end-to-end video.

Although most of the previous research focused on the video streaming QoS in terms of delay, throughput, and quality, DDS in addition to taking delay and performance in consideration because it is originally for real-time distributed systems also adds more QoS mechanisms that were not existent in previous approaches, such as content-based filter, time-based filter, and resource management; see Section 3. Furthermore, allowing each participant to tune his QoS parameters independently adds significant improvement to the application and makes it more flexible and portable. Subsequent sections of this paper focus on merging these capabilities and how they affect the video streaming performance.

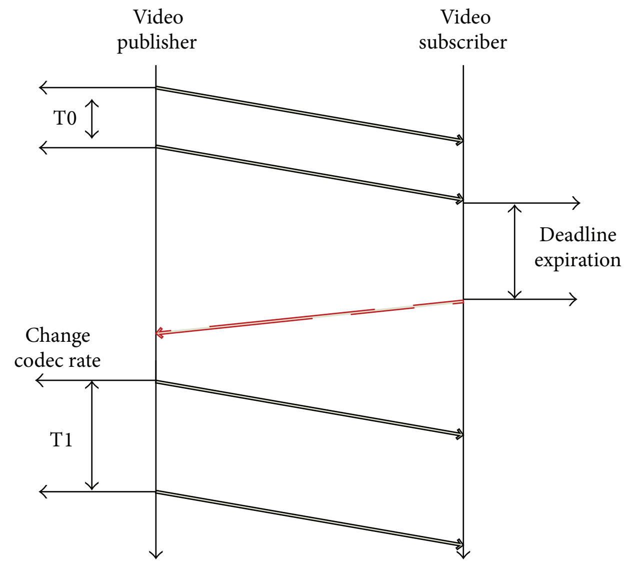

Figure 2: Deadline QoS and congestion control.

3. QoS Architecture of Video Streaming

In this section we review and discuss the QoS of DDS that can be adapted to improve video streaming and reduce the effect of network congestion. Also, we provided an analysis on the use of DDS with the VLC video streaming application.

3.1. DDS QoS Polices for Video Streaming Support. Many QoS policies are used by DDS middleware to support smooth video transmission over networks and also to minimize the required bandwidth, this is very important for many companies and institutions that have to pay for bandwidth usage. A proof of concept study from Granada University [20] is performed to proof the suitability of video streaming over DDS; they stated some of the QoS policies that affect video streaming. In this section, we investigate these polices and show how they can be used to support video streaming applications.

(i) *Deadline and congestion control*, network congestion occurs when a link or node is overloaded and as a consequence it results in packet loss, increased delays, and at times blocking of connections. A lot of research has been done for mitigating network congestion. In the middleware layer, a deadline QoS policy can be used for congestion detection and control, as illustrated in Figure 2. If the subscriber waiting time for the next packet exceeds a certain predefined deadline, it sends a notification to the publisher who will start minimizing the codec rate to avoid congestion on the subsequent streams. When the congestion is overcome, the previous status is recovered.

(ii) *Time-Based Filtering* is the minimum separation time between two successive packets received at the subscriber side. This QoS policy used in video applications is to reduce application load (receiving rate) at

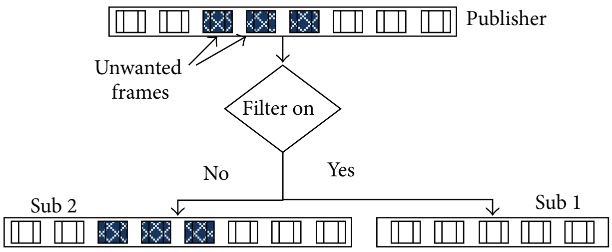

Figure 3: Content-filtered topic.

the subscriber side. For instance, suppose that the publisher is a server and subscribers are different devices that have different capabilities, for example, laptop, PDA, cell phones, or even sensors in WSNs, each one of them has to adapt the receiving rate based on its available resources using such policy. Note that because the deadline is the maximum wait time for data update on the subscriber side, the time-based filter value must be less than the deadline value.

(iii) *Lifespan* avoids delivering stale data, where each packet has its expiration date that will be examined on the subscriber side before playing it back. In video transmission, it can be used to drop the stale received packets because video application is only interested in data with short delays; this QoS is very useful on live video streaming to keep a consistent playback.

(iv) *Best Effort and Presentation* are related to each other; the presentation QoS is used to assert that subscribers will receive data in the order in which it was sent by the publisher, where video samples should be retrieved in the same order. In the best effort QoS, the video frames are delivered with minimum delay; thus it is useful in real-time video transmission where time is more sensitive than packet loss; this QoS policy uses the presentation QoS to assert ordered packet delivery.

(v) *Content-Filtered Topic* is a very useful feature if you want to filter data received by the subscriber. It also helps to control network and CPU usage on the subscriber's side because only data that is of interest to the subscriber is sent. In video transmission, this feature can be used to filter the received video such that each subscriber will just receive only relevant data. Figure 3 depicts the behavior of this QoS policy.

3.2. H.264 in DDS. In order to keep this paper self-contained, we describe briefly the H.264/AVC video compression codec used by DDS. For more detailed information about H.264/AVC, the reader is referred to the standard [21] or corresponding overview papers [22–25].

For video coding, DDS video streaming tool has been integrated with the H.264/AVC (advanced video coding) standard [21]. H.264/AVC is the latest video coding standard of the ITU-T Video Coding Experts Group and the ISO/IEC Moving Picture Experts Group. The main goals of the H.264/AVC standardization effort have been enhanced compression performance and provision of a "network-friendly" video representation addressing "conversational" (video telephony) and "nonconversational" (storage, broadcast, or streaming) applications. H.264/AVC has achieved a significant improvement in rate distortion efficiency relative to existing standards. However, one of the open research issues is to improve this tool by using H.264/SVC (Scalable Video Coding) [4]. SVC enables the transmission and decoding of partial bit streams to provide video services with lower temporal or spatial resolutions or reduced fidelity while retaining a reconstruction quality that is high relative to the rate of the partial bit streams. Hence, SVC provides functionalities such as graceful degradation in lossy transmission environments as well as bit-rate, format, and power adaptation.

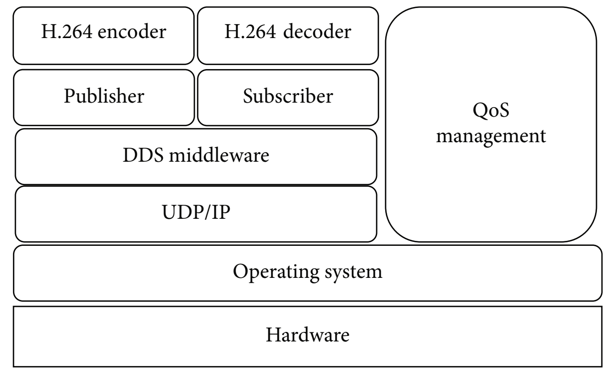

Figure 4: Architecture of video streaming over DDS.

An H.264 stream is a sequence of NALUs (network adaptive layer units). A NALU is formed by a header and a payload carrying the actual encoded video frame. The NALU header contains information about the NALU type and its relevance in the decoding process. From the information reported in the NALU header, we are specifically interested in the three parameters called dependency id (DID), temporal id (TID), and quality id (QID). Each parameter determines a specific scalability facility. DID allows coarse grain scalability, TID allows Temporal Scalability and QID allows Medium Grain Scalability.

The NALUs are represented by DDS using NALU-topic which used to deliver NALUs containing video frames. The structure of the data-sample of the NALU-Topic contains: a H.264 NALU, the ssid and a marker-bit. Both ssid and marker-bit are used for rate-control purpose [1]. The video-publisher is the sender of the video: it executes the software logic interacting with the DDS facility and hosts the data writers. The video-publisher is fed by H.264 NALUs coming from the encoder and, by parsing entering NALUs, builds data-samples and sends them to data writers (DWs). Figure 4 shows the architecture that we have used in this work. The video-subscriber is the module used to receive the video, executes the software logic interacting with the DDS facility, and hosts the Data Readers (DRs).

3.3. VideoLAN VLC Media Player. VLC stands for Video LAN Client, but since VLC is no longer a simple client, this abbreviation is not applicable. VLC is a highly portable free and open source media player and streaming media server written by the VideoLAN project [26]. Video streaming on VLC is based on the client server architecture and thus is an ideal streamed player for our evaluation. VideoLAN is a group of people, who produce and distribute free and open source software for video and multimedia purposes, released under Open Source licenses. It started as a student project at the French École Centrale Paris but is now a worldwide project with developers from everywhere and dozens of millions of people using VideoLAN's software.

Taking these QoS parameters and architectures into consideration, the next section focuses on examining the performance and QoS effect of DDS on video streaming and compares it with the VLC video player.

4. Experimental Work

In this section, we experimentally evaluated the performance of stored video streaming over LAN using both DDS middleware and VLC player.

4.1. Hardware and Software Specifications. The experiment was carried using hardware and software tools; the measurement and monitoring tools and hardware platform specifications that were used are described in Tables 1 and 2, respectively.

4.2. Experimental Setup and Performance Metrics. As shown in Figure 5, the experiment test-bed was composed of three HP computers that are connected using speed-touch hub with 100 Mbps speed; those computers are provided with measurement and monitoring tools that are shown in Table 1. The two technologies, DDS and VLC, are examined by transmitting a full motion video clip of 72 seconds of length, 640 × 480 resolution, and 600 codec bit-rate at 25 fps for each, using H.264 decoder. This video clip was taken from the RTI DDS video streaming tool that we used in our experimental evaluation, for vehicle traffic that shows high motion of cars, making the comparison more accurate.

The QoS parameters are adjusted to meet the existing network link specification; for example, the deadline is adjusted to infinite, lifespan is also infinite, and reliability is the best effort. These parameters are suitable for dedicated and fast networks such as Ethernet LAN because they are congestion free, fast, and reliable. One computer represents a publisher and the others represent the subscribers. In our experiment, we examined the network with different numbers of subscribers (3, 6, 9, 12, 15); however, the effect was not clear in the visual frames; therefore, we used the background traffic (generated by Jperf) to make the comparison of DDS and VLC more visible. Since it was very difficult to examine the technology scalability by increasing number of subscribers or clients, we used Jperf to generate background traffic to make our experiments more realistic. The background traffic was 75% of the available bandwidth which is nearly 98 Mbps (measured by wire-shark). This percentage of background traffic is specified after performing intensive experiments until we observed the effect of traffic on the video quality.

Table 1: Tools and programs.

Tool	Version	Purpose
VLC player	1.0.6	Video streaming
RTI DDS	4.5	Video streaming
Wire-shark	1.2.7	Measure BW, PKT loss, Jitter
Jperf	2.0.2	Generate background traffic
RTI Analyzer	4.5	QoS monitoring and network debugging

The consumed bandwidth is a very important metric for performance evaluation because consumers have to pay for used bandwidth; also reducing the used bandwidth increases the network performance in terms of delay, jitter, and packet loss. Thus, our performance metrics concentrate onbandwidth and in addition we also study packet loss, and jitter (Packet Delay Variation). Besides these objective measurements, we added a subjective measurement [27] which makes it easier for human eye evaluation of screen shoots during simulations.

4.3. Results and Analysis. The results were collected after repeating the experiments several times and then averaging out. Figure 6 shows the effect of background traffic on video traffic from both DDS and VLC. The frames have been taken during playback at the subscriber side in three cases, with 3, 9, and 15 subscribers. And for those cases where there was some distortion in the frames, we selected those frames where the damage was visible. We start examining the effect of background traffic from 25% and 50%, but no effect was visible, then we adjusted it to 75% and then the effect began to appear from the case with 9 subscribers in VLC and from the case with 15 subscribers in the DDS setup. In general, the figure shows that DDS outperforms VLC, where in the case of 15 clients in VLC, the system was unstable and the picture was very choppy, whereas in DDS, the picture started behaving intermittently for case of 15 subscribers, and it was quite choppy. This indicates the effectiveness and scalability of DDS video streaming over VLC.

In Figure 7, the consumed bandwidth is considered as the comparison performance metric, measured without adding the background traffic. From this figure, you can see that the consumed bandwidth is almost the same for the same technology both with background traffic and without it since the video traffic is the same in both cases. The slight increase in VLC is due to the increase of control packets to mitigate the overloaded network. Intuitively larger number of subscribers lead to more bandwidth consumption; but it is clear that VLC increases its rate at a higher rate than DDS where the slope of the line is less and more stable (linear). The figure shows that DDS clearly consumes much less bandwidth than VLC and moreover the difference is even more evident with increasing number of subscribers, where it was nearly 2 Mbps in case of 3 subscribers and it reaches about 5 Mbps in case of

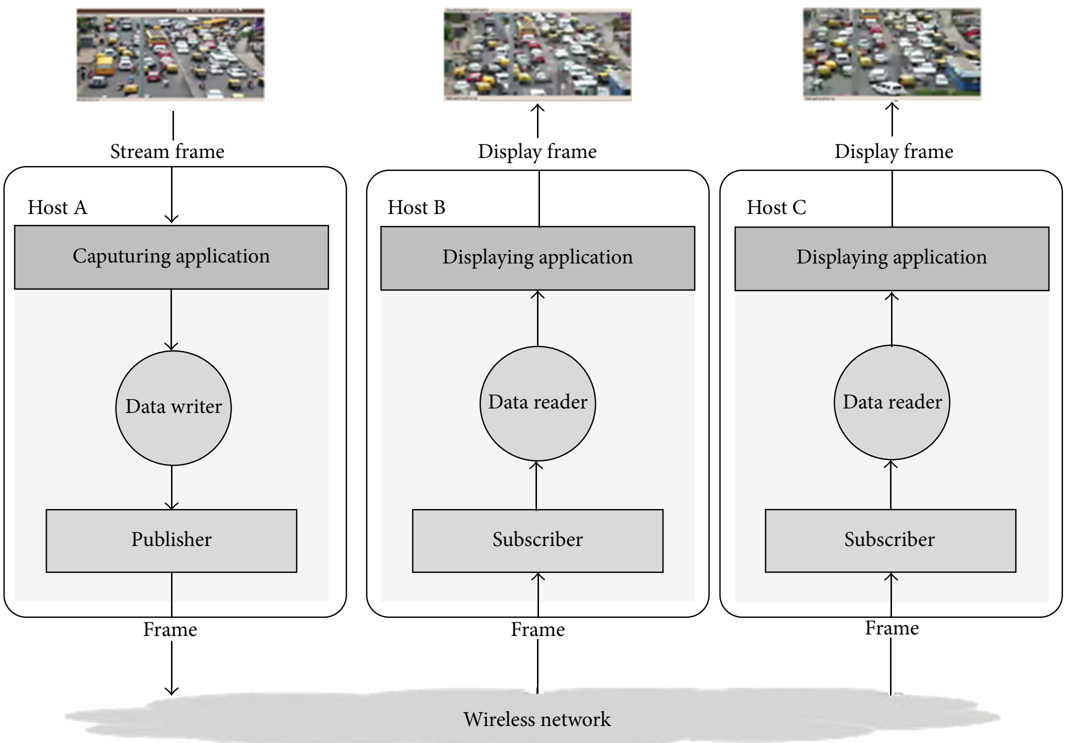

Figure 5: Experiment test-bed.

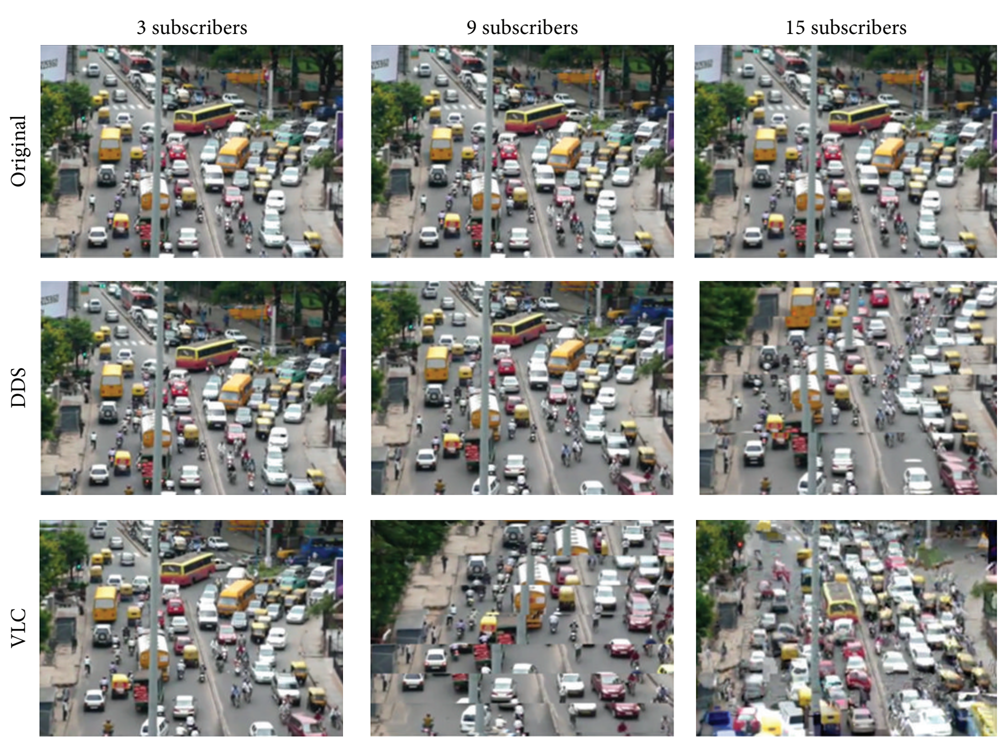

Figure 6: Visual comparison with background traffic.

Table 2: Platform specifications.

	Publisher	Subscriber A	Subscriber B
CPU	Intel(R) Core(TM) i5 2.40 GHZ	Intel Core (TM) i5 2.40 GHZ	Intel(R) Core(TM) i5 2.40 GHZ
Memory	1.8 GiB	1.8 GiB	1.8 GiB
OS	Ubuntu (lucid) Release 10.04 LTS Kernel Linux 2.6.32-37-generic GNOME 2.30.2	Ubuntu (lucid) Release 10.04 LTS Kernel Linux 2.6.32-37-generic GNOME 2.30.2	Ubuntu (lucid) Release 10.04 LTS Kernel Linux 2.6.32-37-generic GNOME 2.30.2
Network connection	Ethernet 100 Mbps	Ethernet 100 Mbps	Ethernet 100 Mbps

Table 3: BW consumption by video streams using DDS and VLC.

No. of sub.	Without BG traffic		With BG traffic	
	DDS	VLC	DDS	VLC
3	1.5%	2.8%	1.4%	2.6%
6	3%	4.4%	2.7%	4.6%
9	4.4%	7%	4.1%	7.2%
12	5.6%	10.8%	5.4%	10.7%
15	7.1%	11.5%	6.1%	12%

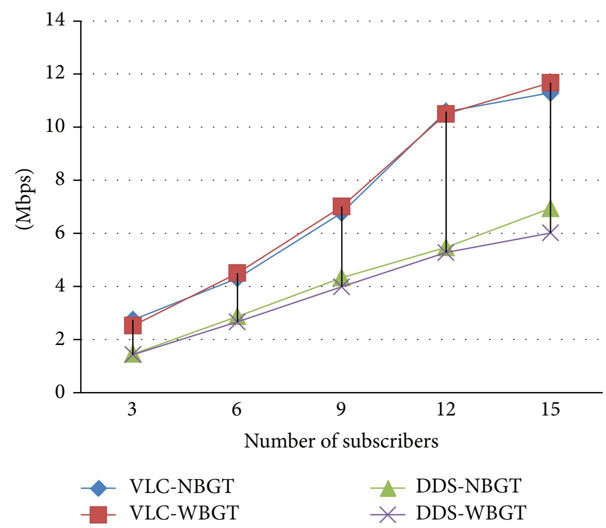

Figure 7: Consumed bandwidth with and without background traffic.

15 subscribers. Table 3 shows exactly how much bandwidth percentage is consumed for both technologies.

In contrast to Figure 7, Figure 8 illustrates that VLC and DDS had similar number of dropped packets both with and without background traffic. This is because the packet dropping of DDS in case of no background traffic is very different than when background traffic is present, whereas, for bandwidth, the background traffic has a significant effect. In case of no background traffic, both mechanisms had acceptable performance. As can be seen in the figure, the worst case result was with 15 subscribers and 5000 VLC packets, which represent less than 3% of the total packets sent, thus nullifying the effect of background traffic. The packet loss effect was clear on the other cases with background traffic, where it was very visible especially in case of VLC that the frames were very choppy and even the color was variable, with high failure of certain clients during the video streaming process.

Figure 9 shows the jitter, that is, packet delay variation, both with and without background traffic for DDS and VLC. It is clear that DDS outperforms VLC, where the difference is about 40 ms. The stream is mostly affected by the packet loss factor, because jitter is unaffected by this loss, whereas, for delays greater than 100 ms, the jitter effect is more visible.

In Figure 10, we examine the content-based filter QoS; we use a scenario of tracking an object location by the publisher and sending the coordinates to the subscribers who show interest in a specific region. Each subscriber can specify in his contract a subarea of interest to avoid bothering itself with irrelevant data. As can be seen from the figure, the filter size of the y axis ranges from 25 to 100%. The filter size illustrates the area percentage covered by a given publisher. As the filter size narrows, the throughput at the subscriber's side decreases nonlinearly (because of the object movement randomization). Therefore, the figure shows that in case of 25%, the throughput almost halves as in case of 100%. Likewise, time-based filter QoS can also be used to control the receiving rate of the published data, for example, to avoid overwhelming the limited resource devices at the subscriber side.

Figure 11 also examines two main QoS parameters used in DDS, both being presented at the transport layer by TCP and UDP protocols. However, because DDS uses UDP in the transport layer, it supports reliable transmission by adding reliable and best effort QoS to the application layer (middleware layer). Reliable and best effort QoS are examined using data readings (nonheavy traffic) in Figure 11. The consumed bandwidth appears to be quite low with respect to video traffic. As can be seen from the figure, the reliable scenario uses more bandwidth as the number of subscribers increase. Similar to TCP, DDS also uses acknowledgment packets to assure reliability over UDP; intuitively, these extra packets increase as number of subscribers increase.

In Figure 12, we compare the data readings traffic and video camera traffic. The video camera traffic is derived indoors, and thus very low traffic (surveillance with almost no movement) is observed. Thereby, we examine the DDS middleware with three types of traffic; data readings, low

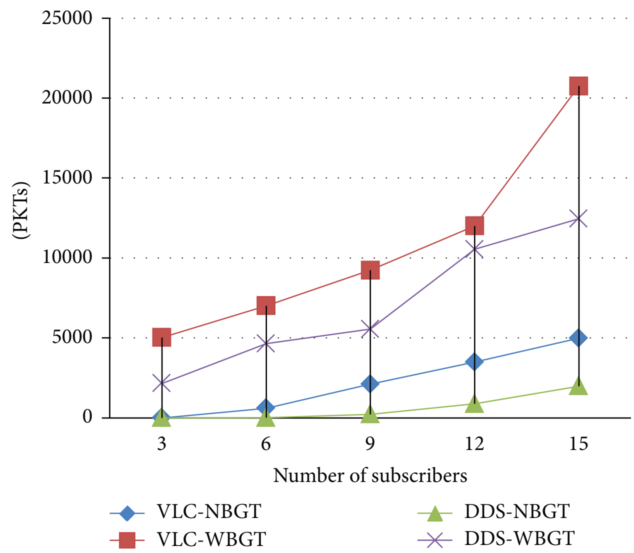

Figure 8: Dropped packets with and without background traffic.

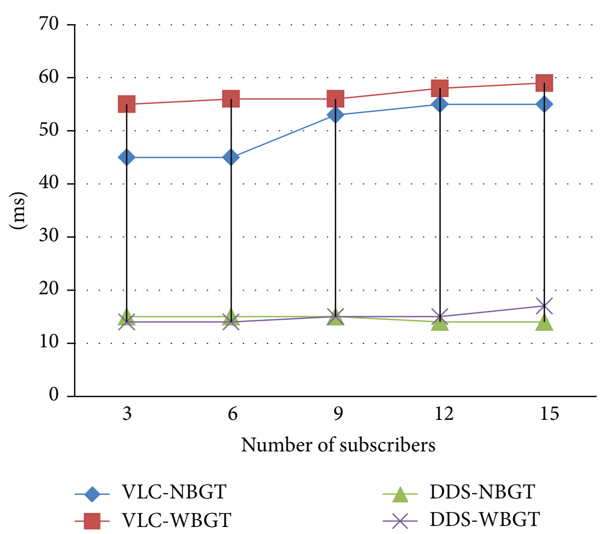

Figure 9: Jitter with and without background traffic.

traffic video surveillance, and high traffic video streaming. In video camera traffic, the throughput reaches almost 1.8 Mbps in case of 10 subscribers which makes it suitable as a low price choice for video surveillance applications. Moreover, Figure 13 illustrates the impact of interference on DDS performance (in WLAN); it shows that the consumed bandwidth increases due to control packets used for mitigating packet dropping and congestions; reaching 4 Mbps in case of 10 subscribers.

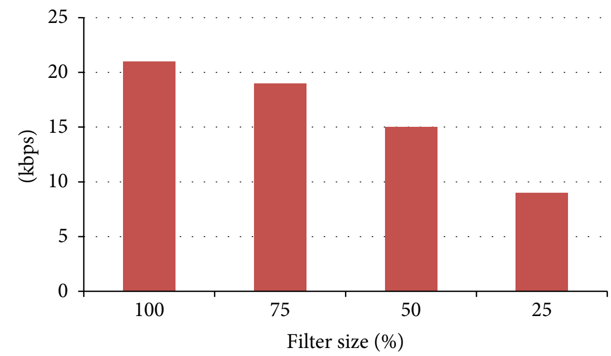

Figure 10: Impact of filter size in content-based filter QoS on DDS middleware traffic.

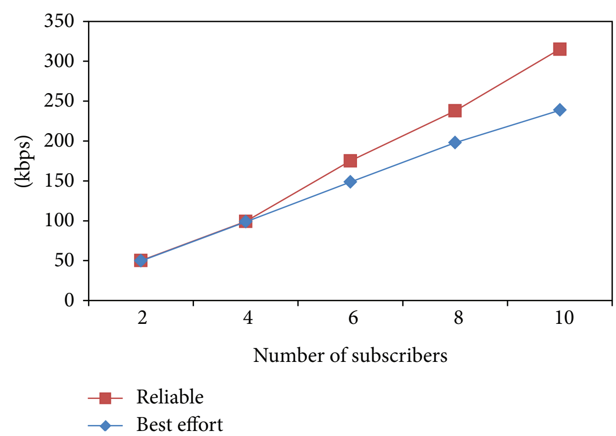

Figure 11: Reliable and best effort QoSs comparison in terms of consumed bandwidth.

5. Conclusions and Future Work

This paper introduced a performance evaluation for video transmission over LAN using Data Distributed Service (DDS). To the best of our knowledge, this is the first study examining the real effect of video distribution using DDS on the network bandwidth and jitter, while comparing it with VLC video streaming player. From our results we conclude that this technology is a promising technology for distributing video over networks, since it consumes low bandwidth, has low jitter, and causes lesser packet loss. Furthermore, it gives more control on video streaming through the use of a rich set of QoS polices that are provided by the DDS middleware.

DDS is designed for large-scale distributed systems; however, in our experiments, we considered only 1 publisher and 15 subscribers, whereas, in real-life distributed applications, this number is quite small. This limitation was because of the lack of DDS simulators and limited number of machines that we used in our experiments. Also, the examination of DDS implementation over indoor dedicated WLAN makes the implementation easier because no mechanisms are built to adapt the video streaming to the time-varying bandwidth of the error-prone wireless channels. Thus, this implementation still lakes mechanisms that leverage the DDS QoS support for adaptively streaming the video frames according to the available time-varying network bandwidth. Practically, DDS-based solution, however, is still applicable because it is compared with VLC video streaming tool which is a practical and well-known player in current market.

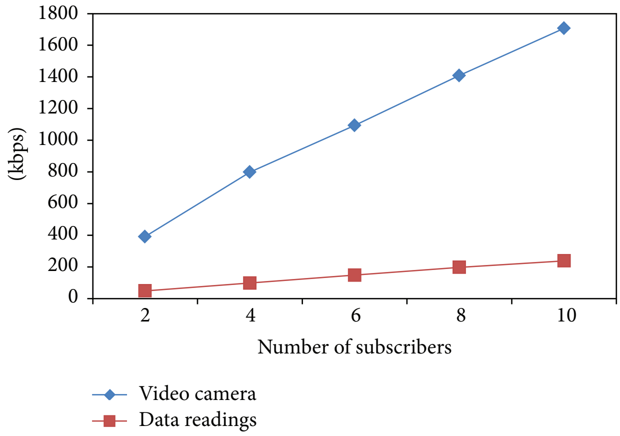

Figure 12: Video surveillance versus data readings traffic.

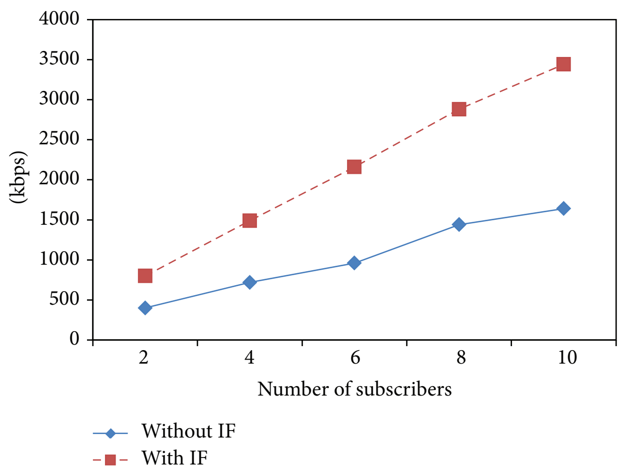

Figure 13: Impact of interference on video traffic over DDS.

As future work, we intend to do this study on more restricted networks such as Bluetooth personal area networks, and examine the QoS parameters to come up with the best configuration for specific conditions. The QoS parameters, also, can be used to solve perennial network problems such as network congestion. Furthermore, scalable video streaming over DDS QoS policies is to be proposed. This mechanism adaptively is supporting video streaming over time-varying bandwidth and error-prone networks.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors would like to acknowledge RTI Ltd. Also, they would like to thank King Fahd University of Petroleum & Minerals for their support for conducting this research work.

References

[1] A. Detti, P. Loreti, N. Blefari-Melazzi, and F. Fedi, "Streaming H.264 scalable video over data distribution service in a wireless environment," in *Proceedings of the IEEE International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM '10)*, June 2010.

[2] OMG, "Data Distribution Service for Real-time systems," Object Management Group, 1. 2 formal/07-01-01 edition, January 2007.

[3] A. Corsro, *The OMG Data Distribution Service for Real-Time Systems*, Opensplice DDS, PrismTech, 2008.

[4] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1103–1120, 2007.

[5] J. A. Clavijo, M. J. Segarra, C. Baeza et al., "Real-time video for distributed control systems," *Control Engineering Practice*, vol. 9, no. 4, pp. 459–466, 2001.

[6] D. A. Kaff, C. Rodrigues, Y. Krishnamurthy, I. Pyarali, and D. C. Schmidt, "Application of the QuO quality-of-service framework to a distributed video application," in *Proceedings of the 3rd International Symposium on Distributed Objects and Applications (DOA '01)*, pp. 299–308, 2001.

[7] ZeroC, "Distributed Programming with ICE," Zero Company, version 3. 3. 1., March 2009.

[8] P. B. Val, M. García-Valls, and I. Estévez-Ayres, "Simple asynchronous remote invocations for distributed real-time java," *IEEE Transactions on Industrial Informatics*, vol. 5, no. 3, pp. 289–298, 2009.

[9] V. Vora and T. Brown, "High rate video streaming over 802. 11nin dense Wi-Fi environments," in *Proceedings of the 5th IEEE Workshop on Network Measurements (WNM '10)*, Denver, Colo, USA, 2010.

[10] D. Li and J. Pan, "Performance analysis and evaluation of H.264 video streaming over multi-hop wireless networks," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '08)*, pp. 5003–5007, December 2008.

[11] B. Al-madani, A. Al-Roubaiey, and T. Al-shehari, "Wireless video streaming over data distribution service middleware," in *Proceedings of the IEEE 3rd International Conference on Software Engineering and Service Science (ICSESS '12)*, pp. 263–266, June 2012.

[12] M. Chen and A. Zakhor, "Rate control for streaming video over wireless," in *Proceedings of the 23th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '04)*, pp. 1181–1190, March 2004.

[13] T. Stockhammer, H. Jenkač, and G. Kuhn, "Streaming video over variable bit-rate wireless channels," *IEEE Transactions on Multimedia*, vol. 6, no. 2, pp. 268–277, 2004.

[14] N. Nasser, "Service adaptability in multimedia wireless networks," *IEEE Transactions on Multimedia*, vol. 11, no. 4, pp. 786–792, 2009.

[15] D. Li and J. Pan, "Performance evaluation of video streaming over multi-hop wireless local area networks," *IEEE Transactions on Wireless Communications*, vol. 9, no. 1, pp. 338–347, 2010.

[16] P. Manjul, V. Balasubramanian, Y. Li, and J. Xu, "Real-time video streaming over multi-hop ad-hoc networks," in *Proceedings of the 2nd International Conference on Networking and Distributed Computing*, vol. 4, pp. 7695–4427, 2011.

[17] A.-J. Garcia-Sanchez, F. Garcia-Sanchez, J. Garcia-Haro, and F. Losilla, "A cross-layer solution for enabling real-time video

transmission over IEEE 802.15.4 networks," *Multimedia Tools and Applications*, vol. 51, no. 3, pp. 1069–1104, 2011.

[18] C.-M. Huang, Y.-C. Chen, and S.-Y. Lin, "The QoS-Aware Order Prediction Scheduling (QOPS) Scheme for Video Streaming Using the Multi-path Datagram Congestion Control Protocol (MP-DCCP)," in *Proceedings of the 15th International Conference on Network-Based Information Systems (NBiS '12)*, pp. 276–283, September 2012.

[19] S. Jina, S. Tarnoi, and W. Kumwilaisak, "QoS-aware multi-rate H. 264 scalable video multicast with network coding in lossy networks," in *Proceedings of the 9th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON '12)*, pp. 1–4, May 2012.

[20] J. M. Lopez-Vega, J. Sanchez-Monedero, J. Povedano-Molina, and J. M. Lopez-Soler, "QoS policies for audio/video distribution over DDS middleware," in *Proceedings of the Real-time and Embedded Systems Workshop*, 2008.

[21] ITU-T and ISO/IEC JTC 1, "Advanced video coding for generic audiovisual services," *ITU-T Recommendation H. 264 and ISO/IEC, 14496-10 (MPEG-4 AVC)*, Version 8 (including SVC extension): Consented in July 2007.

[22] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, 2003.

[23] G. J. Sullivan and T. Wiegand, "Video compression-from concepts to the H.264/AVC standard," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 18–31, 2005.

[24] D. Marpe, T. Wiegand, and G. J. Sullivan, "The H.264/MPEG4 advanced video coding standard and its applications," *IEEE Communications Magazine*, vol. 44, no. 8, pp. 134–142, 2006.

[25] G. J. Sullivan, H. Yu, S.-I. Sekiguchi et al., "New standardized extensions of MPEG4-AVC/H.264 for professional-quality video applications," in *Proceedings of the 14th IEEE International Conference on Image Processing (ICIP '07)*, pp. I13–I16, San Antonio, Tex, USA, September 2007.

[26] J.-B. Kempf, VideoLAN project, http://www.videolan.org/.

[27] J.-S. Lee, F. De Simone, T. Ebrahimi, N. Ramzan, and E. Izquierdo, "Quality assessment of multidimensional video scalability," *IEEE Communications Magazine*, vol. 50, no. 4, pp. 38–46, 2012.