Video Transmission Over Lossy Wireless Networks: A Cross-Layer Perspective

Scott Pudlewski, *Member, IEEE*, Nan Cen, *Student Member, IEEE*, Zhangyu Guan, *Member, IEEE*, and Tommaso Melodia, *Member, IEEE*

***Abstract*—Video content currently makes up nearly half of the "fixed" Internet traffic and more than a third of the mobile traffic in North America, with most other regions showing similar trends. As mobile data rates continue to increase and more people rely on 802.11 wireless for home and commercial Internet access, the amount of video transmitted over at least one wireless hop will likely continue to increase. In addition, as cameras continue to become smaller and cheaper, the demand for video services in sensor and MANET networks will also increase. In this paper, we examine the state of the art of wireless video communication at each layer of the networking stack. We consider both existing and emerging technologies at each layer of the protocol stack as well as cross-layer designs, and discuss how these solutions can increase the video experience for the end user.**

***Index Terms*—Video encoder/decoder, wireless video streaming, cross-layer design, compressive sampling, cloud computing.**

I. Introduction

In 2013, it was reported [1] that 28% of all peak-time Internet traffic was attributed to Netflix [2], while YouTube [3] accounted for another 17%. To put this in perspective, all http traffic only accounted for 9% of traffic [1]. In addition, as available data rates increase, interactive video services [4] are becoming more profitable and therefore more prevalent. As low priced video-enabled mobile devices (such as smart phones) become more common, video creation and consumption on mobile devices is increasing dramatically. Most of these advances are due (at least in part) to advances in predictive video encoding algorithms (for example, H.264/AVC [5] and the recently finalized HEVC [6]), which enable extreme compression of video traffic with very little distortion.

However, restrictions in data rate, computational complexity, battery life, cost, and channel quality still limit the performance of wireless video streaming on resource-constrained devices. The current practice of focusing predominantly on rate-distortion performance with little consideration for these restrictions has led to the development of encoders that often perform poorly in non-ideal network conditions. Current wirelessly networked streaming systems are therefore affected by the following limitations.

A. Data Rate Constraints

While the exact data rate required for a video stream is highly variable and dependent on specific parameters of the video encoder, video streaming traditionally requires high throughput. To use commercial systems as an example, the YouTube streaming rate is usually around 285 kbit/s, and it can reach rates of up to 1005 kbit/s [7]. Similarly, Netflix data rates range from 100 kbit/s to 1750 kbit/s for standard definition, and 2350 kbit/s to 3600 kbit/s for high definition, with the application choosing the rate based on network conditions [8]. Because these streaming systems are designed to fully take advantage of any wireless capacity, they will often force the network to operate in a *near-congested* regime, forcing the application to adapt the video rate to prevent congestion as other flows enter the network.

B. Complexity Constraints

While high-end mobile devices have recently become commercially available (i.e., smartphones, tablets), these devices are for the most part battery-powered and resource-constrained. While they are capable of implementing complex video encoding algorithms, the real-time execution of such algorithms will drain the battery of the device very quickly [9]. This leads to a tradeoff between data rate requirements and power constraints. Reducing the complexity of the compression algorithm to conserve energy at the encoder decreases the resulting encoding efficiency, resulting in larger amounts of data that need to be transmitted.

C. Channel Conditions

Compensating for lossy wireless channels is a major challenge. It is well known that predictively-encoded video is susceptible to bit errors. This is due in part to the use of variable-length coding (i.e., Huffman coding) in which a single bit error can cause the loss of entire blocks of data. In data networks, bit errors are usually dealt with using some form of error correction scheme which generally has an all-or-nothing approach to error correction, in that a received packet is either entirely correct or is discarded and must be retransmitted. However, while too many errors can cause significant distortion to the end user, guaranteeing error-free reception is often unnecessary [10]. While the quality does decrease sharply when the BER increases beyond some threshold, for low levels of BER there is typically *no perceptually observable decrease in video quality*.

Manuscript received November 22, 2013; revised April 10, 2014; accepted June 05, 2014. Date of publication July 23, 2014; date of current version January 20, 2015. This work was supported in part by the U.S. Office of Naval Research under Grant N00014-11-1-0848 and by the U.S. National Science Foundation under Grant CNS1117121. The guest editor coordinating the review of this manuscript and approving it for publication was Prof. Béatrice Pesquet-Popescu.

S. Pudlewski is with the Massachusetts Institute of Technology (MIT) Lincoln Laboratory, Lexington, MA 02420 USA (e-mail: scott.pudlewski@ll.mit.edu).

N. Cen, Z. Guan, and T. Melodia are with the Department of Electrical Engineering, The State University of New York, Buffalo, NY 14260 USA (e-mail: nancen@buffalo.edu; zguan2@buffalo.edu; tmelodia@buffalo.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSTSP.2014.2342202

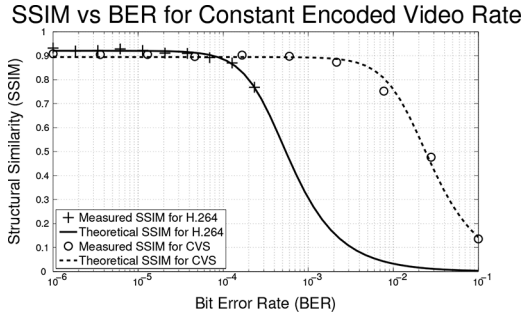


Fig. 1. SSIM [11] vs BER for H.264 and CVS Encoders.

This leads to an obvious tradeoff between the quality of the received video and the techniques used to reduce the BER. This is shown in Fig. 1, which presents the performance of compressive video sensing (CVS) [9] and H.264/AVC [5] for increasing bit error rates (BER). CVS is a video encoder based on compressed sensing principles, which will be discussed in detail in Section III-D. In these tests, we see that there is little or no effect in the SSIM quality measurement of the received video for BER rates of up to 10^{-4} for H.264 [5] or for 10^{-3} for CVS [9]. This shows that, while channel conditions can be highly variable, in many cases we can *simply ignore* errors. When we look at a system implementation, this intuitively means that even with low-quality and low-power transceivers such as in wireless sensor network (WSN) systems [12], [9], it is still possible to achieve high-quality video by i) *avoiding the use* of packet-level error detection schemes such as CRC, and ii) assuring that the BER is "low enough". Below, we will discuss techniques at both the video encoder and PHY layer that can potentially accomplish this.

D. Network Constraints

These tradeoffs get tightly coupled and become much more difficult to analyze when streaming video over large scale networks, e.g., mesh, sensor networks, and vehicular networks. First, it is by no means easy for the video source to acquire exactly the dynamic state information of the entire network, which may comprise multiple hops, multiple paths, and be characterized by mobility and heterogeneity of devices. Second, information-centric (or content-aware) transmission is not easy at intermediate nodes that, unlike the original source, may not be aware of all the details of the videos to be transmitted. Instead, only limited video information (e.g., roughly-classified priority level, delay requirements) may be available, while the packetized video content is typically not decoded at intermediate nodes. To achieve a good tradeoff between in-network processing and transmission, cross-layer design may be needed that jointly considers application-specific Quality of Experience (QoE) requirements, end-to-end rate control, path and link maintenance, available radio resources, and available energy budget of the nodes. It is still a challenge to dynamically and adaptively find the optimal paths under application-specific constraints to improve the end users' QoE. Some approaches designed to address these challenges will be discussed in the remainder of this paper.

E. Summary

In this work, we will discuss how these challenges are being addressed by state-of-the-art solutions at all layers of the networking protocol stack. We will examine both the current state-of-the-art as well as emerging trends in wireless video streaming. We will discuss recent work being performed at each layer to enhance video communications, as well as discuss open problems in wireless video networking and comment on the way forward.

The remainder of this paper is structured as follows. In Section II, we will discuss the system models and network scenarios that will be considered throughout the article. Section III discusses challenges in the design of modern video encoders for wireless networks. We will then discuss solutions at each of the layers of the networking protocol stack. We will concentrate on transport layer in Section IV, network layer in Section V, data link layer in Section VI, physical layer in Section VII, and on cross layer solutions in Section VIII. Finally, in Section IX we will draw the main conclusions.

II. Wireless Video Transmission Overview

In this section, we briefly introduce the system models and network scenarios that will be considered as a reference in the remainder of this article. We first discuss a number of "typical" wireless paradigms, and discuss for each of them advantages and challenges for video streaming. We then discuss each layer of the networking protocol stack, and discuss their roles and influence on wireless video transmission. The goal is to demonstrate how the received quality of wireless video is influenced by control decisions taken at each and every layer of the protocol stack.

A. Video Network Paradigms

Not surprisingly, the performance of wirelessly-transmitted video is highly dependent on the type of wireless network considered. In general, it is more challenging to transmit video over networks with multiple wireless hops. While the solutions we present here attempt to be independent of any specific wireless networking protocol, wireless networks typically fall into one or more of the following broad categories.

- **Cellular networks.** As available data rates in cellular networks increase, the number of users watching or creating video on mobile phones is increasing dramatically. Current-generation cellular networks are wireless only at the first and last hop, leading to what is essentially a wired network with a wireless "last-mile" link[1]. This wired backbone allows for much higher data rates than are possible in fully wireless networks. However, the devices themselves are usually resource constrained. A typical cellular network architecture is shown in Fig. 2(a).
- **Wireless local area network (WLAN).** Similar to cellular networks, a home or commercial wireless local area network generally relies on existing wired infrastructure for the majority of the link, and some version of the IEEE 802.11 [13] standard to bridge the network from the fixed infrastructure to an end user. Similar to a cellular network, since there is only a single wireless hop, the data rate is typically limited by the rate achievable on the wireless link. The network architecture of a WLAN is shown in Fig. 2(b).
- **MANET, and tactical MANET.** A mobile ad-hoc network (MANET) is a self-organizing, self-configuring infrastructureless multi-hop wireless network. While MANETs have

[1]It is worth noting that LTE-advanced, as well as next-generation standards may take advantage of multiple wireless hops. However, the assertion that the majority of the link between two wireless cellular nodes relies on wired infrastructure is still valid.

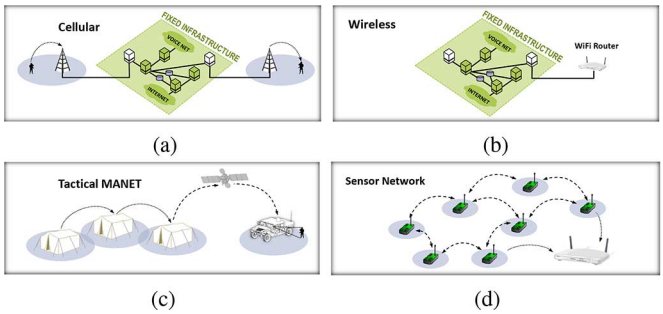


Fig. 2. Network Categories. (a) Cellular networks. (b) Wireless networks. (c) Tactical MANET networks. (d) Sensor networks.

seen limited commercial use, they are fundamental to military tactical networking solutions, and can include highly heterogeneous links. Because they rely on multiple wireless hops, data rates are often very limited. An example of such a network is shown in Fig. 2(c).

- **Sensor networks.** Sensor networks are similar to MANET networks in that they also form an infrastructureless multi-hop wireless network. However, sensor networks are generally designed to be low cost, single purpose networks designed to sense and relay their local environment. This can be through the use of any number of sensing modalities, from simple light and temperature sensors to more complex audio and video collection [14]. In most cases, the sensor networks either take advantage of existing device's networks (such as using existing smartphones to monitor an urban environment), or deploy specific single-use devices. Often in the latter case, the wireless devices are deployed where accessing them is very difficult, and the devices are treated as disposable. An example of a sensor network attached to a wireless router is shown in Fig. 2(d).

B. Video Traffic Paradigms

Along with the network topology, the performance of wireless video is also influenced by the nature of the video application. The specific application influences the tolerance to latency, distortion, and bandwidth, as well as constraints on privacy or security restrictions and format requirements. Below, we will introduce a number of common traffic paradigms.

- **Real-time video.** Real-time video is a video traffic paradigm in which the video is being used for some real-time application such as video telephony. Because of the *real time* requirement, low latency is essential. Even small delays can have a significant impact on the quality of the video communication experience. Such services can be viewed as a system that attempts to minimize the delay between the content being captured at the source and the content being displayed by the receiver. Because of this emphasis on timely delivery, such systems may be willing to sacrifice quality for latency.
- **Video gaming.** Video gaming applications are similar to real-time video applications in that the latency of the delivered video is the primary concern. However, unlike video telephony, which can be viewed as a small number of independent video streams, video gaming applications tend to be highly interactive. Such systems add additional low-latency interactive requirements to the network.
- **Video on demand.** Unlike real-time video services, video on demand services transmit pre-recorded content based on the demands of the end user. Such services generally take advantage of a relatively large buffer as well as the availability of the entire video stream to deliver much higher quality video than more time-sensitive applications.
- **Interactive video.** By its traditional definition, interactive video is essentially video on demand in which the user can *interact* with the playback of the video. This includes traditional playback commands such as pause, fast forward, and rewind. With the huge demand for Internet video services such as Netflix and YouTube, interactive video has become the dominant source of Internet traffic in many countries.
- **Multimedia surveillance.** Video surveillance networks use video applications to monitor an area, usually to attempt to detect unauthorized or unexpected activity. While they often deliver real-time video content to an end user, video surveillance applications may also deliver a high-quality version of the video to a *storage device* for later forensic requirements.

C. The Networking Protocol Stack

We consider the networking protocol stack shown in Fig. 3. In the following sections, we will discuss how decisions at the application, transport, network, data link, and physical layers influence the received quality of video transmitted over a wireless network. Specifically:

- **Application layer.** The application layer is responsible for the *compression* and *formatting* of the video stream. In Section III we will discuss how different compression techniques can influence the quality of video transmitted in lossy channels.
- **Transport layer.** The transport layer controls the end-to-end delivery of the video packets. This influences both the *rate* at which the video packets can be transmitted through the network as well as any end-to-end *delivery guarantees* that may or may not be provided by the protocol. We will discuss techniques for rate control and admission control for wireless video in Section IV.
- **Network layer.** The network layer controls the path selection for the video packets. Section V will discuss how using video-specific metrics in the routing decisions can significantly affect the video quality.
- **Data link layer.** Among the functionalities handled at the data link layer, medium access control (MAC) is responsible for fair sharing of the wireless broadcast medium among different users. We will see in Section VI that by designing the MAC protocol specifically for video transmission, we can greatly increase the received video quality.
- **Physical layer.** The physical (PHY) layer influences the data rate and bit error rate of the video transmission. In Section VII, we will discuss how new advances in cooperative communication techniques and in rateless coding, among others, can be leveraged to improve video quality.

By looking at the above list, it is easy to see that many advances can be achieved by taking video-specific (i.e., application layer) information into account when making lower layer decisions. This cross-layer approach can help enable much of the video-quality optimization that is used in these approaches.

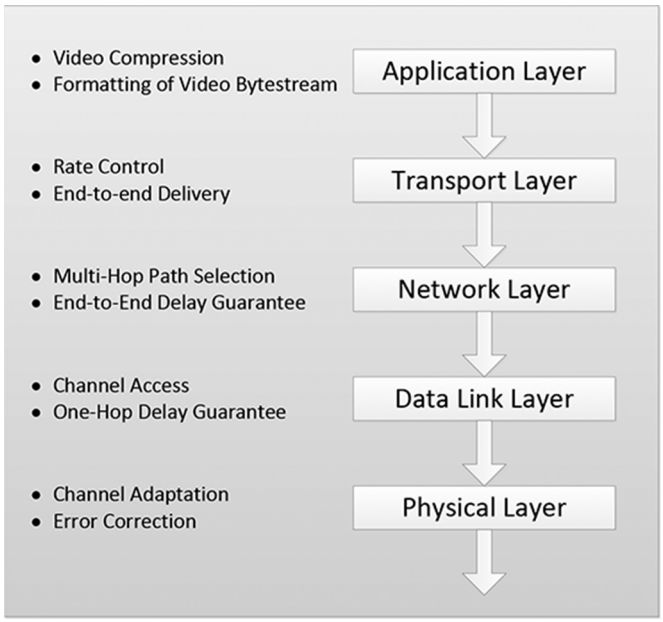


Fig. 3. Networking protocol stack for wireless video streaming.

III. Video Encoders

Wireless video streaming is fundamentally different from video streaming over wired networks. In this section, we will introduce some video encoding techniques that are relevant to wireless video transmission. The reader is referred to [15], [5], [16], [17] and the references therein for more details.

A. Intra-Frame Encoding

Intra-frame video encoders process each frame *independently* using still image encoding. The main difference between this class of encoders and a simple collection of images is that, while the frames are encoded independently, they are transmitted as a single video file (including video headers). While these protocols have relatively poor rate-distortion performance compared to the more complex predictive video encoders, they have low complexity at the encoder. This can be important for those multimedia devices that do not have the processing power (or battery capacity) to execute the more complex algorithms, e.g., wireless sensor nodes. In addition, since each frame is independent, there is far less visible distortion from channel errors.

1) Motion JPEG (MJPEG): MJPEG video encoding is an intra-frame encoding scheme based directly on the JPEG image compression standard [16]. Although there is no official MJPEG standard, the basic concepts of most implementations are the same. Each frame is first divided into 8×8 blocks that are transformed to the frequency domain using the discrete cosine transform (DCT), creating an 8×8 block of DCT coefficients.

From this point, the DCT coefficients of each macroblock are quantized and entropy-encoded, resulting in significant compression. The DCT concentrates most of the signal energy into a few coefficients; this allows the bit-allocation to easily give more weight to the low-frequency components. Moreover, this corresponds well to the human visual system, which allows the encoder to remove many of the high frequency components with little effect on the perceived quality of the resulting image. In MJPEG, this process is done for each frame independently. The resulting video has compression and quality comparable to JPEG image compression and can be extended without significant processing and power requirements at the encoder.

2) Motion JPEG 2000 (MJ2): Similar to MJPEG, MJ2 video is an intra-frame encoding standard based on the JPEG 2000 compression standard [17]. Each MJ2 frame is first split into rectangular regions referred to as *tiles*. Once the image has been decomposed into tiles, a discrete wavelet transform (DWT) is applied. At this point the wavelet transformed coefficients are quantized and entropy encoded. While the compression rate of MJ2 may be better than that of MJPEG [17], there are other drawbacks, specifically flicker artifacts, that decrease the performance of MJ2 [18]. Instead, many of the advantages of MJ2 over the older MJPEG are with the flexibility of the standard. These include a naturally scalable representation and the availability of *regions of interest*, where a spatial portion of the image is encoded at higher resolution than the rest of the image.

B. Predictive Encoding

Discussing the differences between common video encoders (i.e., MPEG2 [19], H.263 [15], H.264/AVC [5], HEVC [6]) is beyond the scope of this work. Instead, we are going to use the H.264/AVC video encoder as defined in [5] as an example of predictive video encoders in general. For details of the other encoders, the reader is referred to the works cited above and the references therein.

Along with the frequency transform - quantization - entropy encoding functions found in MJPEG and MJ2, predictive video encoders take each image block and compare it to other macroblocks both within the same frame (intra-prediction) or in a previous frame (inter-prediction). By identifying the difference between two macroblocks and encoding only that difference, the encoder can significantly reduce the amount of information necessary to represent a video for a desired quality.

For wireless video transmission, the prediction operation has several important consequences. Primarily, it creates an *interdependence* between frames in a video stream. Generally, video streams are divided into three types of frames. There are intra-encoded (I) frames, predicted (P) frames, and bi-directionally predicted (B) frames. As the names suggest, the I frames are encoded using only blocks contained within that frame, P frames are encoded using both blocks within the frame as well as blocks in the previous I or P frame, and B frames are predicted from blocks within the frame and from both the previous *and next* I or P frame. Some encoders may also use more complex frame reference structures in which B frames are predicted from multiple I, P, or even other B frames. As we will see later, while prediction greatly increases the rate-distortion performance of the encoder, there are a number of tradeoffs in both the error resiliency and the complexity. Even though commercial video encoders attempt to be resilient to errors, their primary focus is on improved rate distortion performance. Because of this, while state-of-the-art commercial encoders have very impressive rate-distortion performance, encoders that focus more on error resilience may have other desirable properties in specific video streaming applications.

C. Distributed Video Encoding

It was shown in [20], [21] that the performance of side channel coding could theoretically match that of predictive coding. Since side channel coding naturally pushes much of the complexity from the encoder to the decoder, this could help

address one of our primary challenges if it could be used to develop a video encoder. In [22], the authors accomplish this by developing the Power-efficient, Robust, hIgh-compression Syndrome based Multimedia (PRISM) encoder.

To understand the key concepts of PRISM and similar encoders, first note that the motion prediction is the primary source of computational complexity. Unfortunately, motion prediction is also the primary source of compression in most encoders. Therefore, an encoder that could take advantage of the motion prediction capabilities without the added complexity could be ideal for wireless devices. Side channel coding techniques enable this functionality. Assume as in [22] that X is the block to be coded in the current video frame, Y is the block that represents the best predictor for X, and they are related through a noise term N such that $X = Y + N$. Even though it is computationally infeasible to determine Y at the source, it *is* appropriate to assume that the destination will have access to Y and that the statistics of the "correlation noise" N are known. We can then (after intra-encoding) use the knowledge of N to partition the codeword space of X and send only this *coset* of X to the destination. The destination can then use Y together with the statistics of N to determine which member of the coset should be decoded.

Since there are fewer cosets of X than there are original codewords of X, fewer actual bits need to be transmitted. However, since we know the relationship between X and Y, we can still achieve compression rates similar to that of predictive coding *without determining Y at the source*. In the case of multi-view video, this also provides a potential to exploit inter-view correlation at the receiver side, e.g., jointly decode independently-encoded multi-view videos [23].

D. Compressed Sensing Based Video Encoding

Compressed sensing (CS, aka "compressive sampling") is a new paradigm that allows the faithful recovery of signals from $M \ll N$ measurements where N is the number of samples required for the Nyquist sampling [24]. Hence, CS can offer an alternative to traditional video encoders by enabling imaging systems that sense and compress data simultaneously *at very low computational complexity for the encoder*. CS images and video are also resilient to channel errors [10].

Similar to distributed video encoding described above, compressed sensing based video encoders, specifically CVS [9], take advantage of known properties of the encoded video frame to reduce the number of bits that need to be transmitted to accurately represent the frame. In this case, rather than assuming some prediction signal (as in the Y signal above) CVS leverages the *sparsity* of the DWT or DCT encoded frame, along with the sparsity of the difference between frames, to compress the video. Specifically, the intra-encoded frame $\mathbf{y}_i$ is calculated from the raw frame $\mathbf{x}_i$ as $\mathbf{y}_i = \phi\mathbf{x}_i$, where $\phi \in \mathbb{R}^{M\times N}, M \ll N$ is a "noise-like" under-determined sampling matrix. To recover $\mathbf{x}_i$ from $\mathbf{y}_i$ at the receiver, the decoder solves a sparse signal reconstruction problem, essentially finding the "sparsest" signal that fits the measurements in $\mathbf{y}_i$. To take advantage of temporal correlation, CVS takes advantage of the fact that the difference between two correlated frames is sparse, and uses this information to *again compressively sample* the difference between two encoded frames.

While in general compressed-sensing-based encoders cannot achieve the same compression rates as predictive encoding schemes, there are additional properties including error resilience and extremely low computational complexity [9] that make such encoders appealing for resource-constrained environments. In resource constrained environments with a limited energy budget, CVS was shown to perform better than intra-encoded H.264/AVC [9] in terms of overall energy-rate-distortion performance.

IV. Transport Protocols

Streaming video has traditionally relied on user datagram protocol (UDP) at the transport layer [25]. This was a direct result of the delivery guarantee requirement for TCP. For file transfer applications, it is essential that every packet is delivered. Because of this, TCP will (sometimes severely) increase the latency in order to deliver lost packets. However, as we discussed above for real-time streaming video, latency is the critical parameter in many video streaming applications. This led to video being streamed over the potentially lossy UDP protocol, while the reliability responsibilities were transferred up the protocol stack to protocols such as RTP with RTCP.

However, as video on demand (interactive Internet video services) became more prevalent, the congestion control components of TCP became necessary to prevent congestion of video transmitted from major video distribution services. In addition, higher-bandwidth/lower-latency networks, along with cheap memory for buffering video packets, led to many major video distribution services, including YouTube and Netflix, relying on standard TCP connections [26].

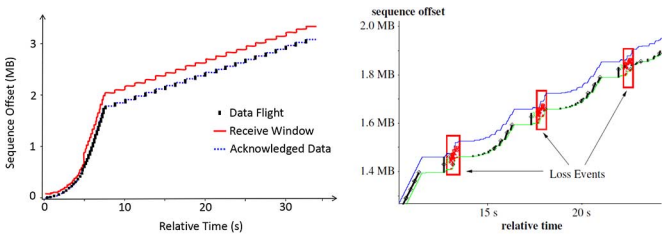
While this is feasible in cellular or Wi-Fi type networks where the data rate is relatively high, MANET and sensor networks pose a different type of challenge. Because of the lossy nature of such networks, TCP performs too poorly to be practical. However, since the data rate of these networks is so low [27], [28], congestion control is essential for multiple video streams to share the network. We will show that integrating congestion control with the video encoder at the application layer can substantially increase the overall received video quality [27].

In this section, we will examine the impact of transport control on wireless video transmission. We will examine how traditional congestion control protocols can be used for different types of wireless video transmission, including an overview of how congestion control is implemented in YouTube. We will also examine some ways in which these traditional techniques can be improved in wireless networks. Finally, we will examine an approach to congestion control in highly-constrained sensor networks.

A. Congestion Control for Wireless Video Using Traditional Protocols

The limited capacity of wireless networks requires that specific functionalities need to be defined to control the video rate when the offered traffic exceeds the network capacity. Therefore, to begin, we will first examine the applicability of standard well-known protocols to the congestion control problem for streaming video.

1) The "Call Manager" Approach: At a high level, a call manager is responsible for admission control into a bandwidth-limited network. The general concept is simply that if the total available capacity of the network is C, and each caller i requires a minimum capacity c_i to achieve an acceptable received call



(a) (b)

Fig. 4. (a) A time sequence graph for YouTube video streaming, and (b) congestion caused by block-based streaming in YouTube [29].

quality, then only k callers such that $\sum_{i=1}^{k} c_i \leq C$ will be allowed into the network.

While this approach was originally designed for wired networks, it can be applied to wireless video streaming, even though some challenges need to be addressed. First, even in a relatively stable network, wireless links are affected by bandwidth fluctuations. Therefore, the value of C will likely change over time, requiring rules for "dropping" video streams that exceed the available capacity. Second, this approach requires a centralized control point in the network that has the ability to prevent new video streams from joining the network. Again, this may be challenging especially in MANET networks where there is by definition no central coordinating node.

Even with these limitations, if the video rate can not be adapted at all, this approach may be the only feasible method for maintaining video quality for at least a subset of users. This approach can enable video in networks where the source of the video is a remote node and/or the capacity of the network is limited. In addition, in video streams where a router can not intelligently drop frames (i.e., due to encryption) the call-manager approach may be the only possible solution.

2) Congestion Control Based on Standard TCP: If the latency constraints of a real-time streaming system can be relaxed, it is possible to implement video streaming services based on standard TCP protocols. Recently, this approach has become more prevalent as the need for congestion control of video streams is becoming more important. For TCP-based video streaming services to be able to deliver acceptable performance, however, a few requirements need to be observed:

- **Sufficient capacity.** Traditional TCP approaches are widely used in commercial video distribution systems. This is largely due to the use of advances video encoders (i.e., H.264 [5]). These encoders allow the video source to compress video to far below the capacity of most cellular and wireless networks (from 285 kbit/s up to 1005 kbit/s [7]).
- **Sufficient buffers.** Capacity in wireless networks is highly variable. Ensuring that the receive buffers are large enough to "smooth over" capacity fluctuations allows for an uninterrupted user experience [29].
- **Sufficient channel quality.** TCP will dramatically underestimate the capacity of a wireless network that has a high packet loss rate [30], [31]. For video streaming to work with TCP, the packet loss rate needs to be very low, or some performance enhancing proxy (PEP) must be utilized to maintain high enough throughput.

Block-based flow control: In [29], in order to understand the operation of the proprietary YouTube video streaming server protocol, the authors investigate application flow control of streaming YouTube video. They were able to determine that the server was limiting the rate that packets were being introduced into the network *based on the playback rate of the video*. The graph showing the TCP sequence number vs. time from [29] shown in Fig. 4(a) demonstrates this "application flow control" scheme based on network traces of YouTube streaming video. For the first 7 seconds, the sequence resembles standard TCP slow-start. However, after this point, the packets are transmitted in "blocks" at regular intervals. The intuitive reason for this is that while the initial slow-start phase is required to build up a buffer at the receiver, there is no reason to transmit at a significantly higher rate than the playback rate of the video. It is conceptually similar to using a third playback window to avoid transmitting more packets than are required. There seems to be two main consequences of this additional window. First, it avoids transmitting video packets from the server that will not be viewed at the receiver, which over many video streams could lead to significant resource savings. It also avoids unnecessary reconstruction and buffering of video content at the receiver. It is also worth noting that while these results are based on YouTube traces, similar functionality has been noted in Netflix streams.

While this scheme effectively limits the transmission rate of the video to the video playback rate, as the authors of [29] point out, the block based streaming scheme as shown in Fig. 4(a) does not perform well in congested or nearly-congested networks. The blocks themselves are relatively large (compared to a single packet). While the rate that new blocks are introduced is limited, all of the packets from an individual block are transmitted at a much higher instantaneous rate. However, since the blocks are large enough to overflow a partially full queue, it is possible that these short bursts of traffic can cause congestion, causing all of the flows to back off (including the offending video stream itself). In a nearly congested home wireless network, this would essentially cause a periodic decrease in the overall throughput of the network. This behavior is verified in [29] and shown in Fig. 4(b), which plots the TCP sequence offset vs time. Each time the source transmits a block of video packets, the TCP protocol experiences a loss event and returns to a slow-start phase.

Non bandwidth-intrusive video streaming. While the block-based implementation may have some drawbacks, the concept of a playback-limited flow control for streaming video can have a lot of benefit. In [32], the authors also noted the block-based rate control. However, they then propose a new end-to-end solution that would limit the rate of video packets into the network based on the video playback rate, but without the burstiness of the existing implementation.

The rate that packets are introduced into the network is based on the playback rate. The server defines an application window that limits the number of in-flight packets based on the video playback buffer at the client. The window is limited *based on how fast the user is watching the video*. If the user pauses the video, the playback buffer will not drain and therefore the server will reduce the introduction of new packets into the system. While this algorithm adapts the TCP sending rate to the video parameters, it does not take the wireless channel into account directly. This is examined explicitly next.

B. TCP Proxy for Wireless Streaming

As wireless networks became more popular, it was clear that TCP alone would not perform well. From very early work in [30] to recent work in [31], it has been shown many times that allowing TCP to treat channel losses as congestion will drastically decrease throughput. Based on this insight, many protocols have been developed to distinguish between channel losses and congestion, and most have shown to improve the wireless performance significantly. Many approaches have been implemented including retransmitting packets over the wireless link [30], using error correction codes [31], and using feedback to estimate the bandwidth explicitly [33], and most of them have been shown to significantly mitigate this problem.

Unfortunately, in most cases, we cannot control the TCP protocol parameters in use at the server. To compensate for this, a performance enhancing proxy (PEP) is often used to split the connection before the TCP data encounters links that have high losses and/or very high latency[2]. PEPs essentially split the TCP stream into two sections. The video is then streamed using TCP over the regions of the network where it performs well, and some other ad hoc protocol in the regions of the network where it does not. In [34], the authors describe such a PEP that is designed to enhance streaming video performance.

In [34], unlike the previous TCP applications we have discussed, the proposed system *adapts the video quality* to the channel quality. The PEP is located in the wireless access point and is responsible for the last-hop wireless delivery of the video stream. This placement allows the PEP to seamlessly integrate with existing protocols without any modifications at either the client or server. The proposed scheme consists of two main components.

Fair resource sharing. One main advantage to placing the proxy at the access point (AP) of a wireless network is that there is a simple one-to-one connection between the AP and the wireless users. This allows the AP to very easily share the wireless resources using any available method (even simple round-robin scheduling could work for this).

Video adaptation. One advantage to using a PEP is that the proxy can inspect and *selectively drop* video packets in a way that would result in a lower-quality *but still acceptable* video stream. Below we describe one simple method for accomplishing this for H.264/AVC [5] encoded video by taking advantage of the network abstraction layer (NAL).

First, the protocol reorders packets based on information in the NAL unit (NALU) headers. The NALU headers indicate, along with the frame type, the *importance* of the frame. For a full explanation of the method that the NAL uses to define the importance of a specific NALU, the reader is referred to [5]. For the purpose of this paper, this value allows the PEP to order the NALUs in order of importance to the received video quality. This ensures that the wireless user will receive the most important portions of the video first.

After the video frames are ordered, the PEP then tries to transmit as many frames as possible. If the wireless channel can support the playback video rate of all videos being transmitted, then there is no noticeable effect. If, however, the channel cannot support the full rate for every user, the PEP will allocate a fair portion of what is available to each node. Since the channel cannot support the full video rate, some packets will be dropped. By ordering and transmitting the packets in the order of their importance of the received video quality, we can ensure that each receiver will receive the highest quality video that the channel can support.

[2]For example, PEPs are used extensively in SATCOM-on-the-move terminals, which experience both very high latency as well as satellite blockage losses.

C. Distortion Minimizing Rate Control (DMRC)

Wireless multimedia sensor networks (WMSNs) are a class of wireless sensor networks that record and collect video data. The classic example of such a network is an infrastructureless video surveillance network for tasks such as wildlife monitoring. Because of constraints in the cost of the devices (they are often damaged and/or lost, and must therefore be low cost), as well as power (they are usually battery powered), WMSNs pose significant challenges for traditional congestion control protocols. However, one advantage to such networks is that, since they are essentially a "single-use" network, the designer has much more freedom that in traditional network design to optimize the network specifically for video transmission.

In this section, we look at the case where the streaming video server is able to intelligently adapt the video encoding rate based on measured channel statistics. In [27], we introduce the distortion minimizing rate controller (DMRC) that uses the signal to noise ratio (SNR) and round trip time (RTT) to determine the cause of distortion at the receiver. Unlike the solutions described above, we assume that the transport protocol is directly able to set the video encoding rate. This allows the transport protocol to make decisions based on the received video quality. Since the video quality may not be directly correlated to the encoding rate for different videos, this ensures that each client receives a "fair" quality video. In addition, DMRC uses forward error correction (FEC) codes to compensate for channel errors. This gives the protocol two "dials" to control to deliver the best quality video possible to the receiver.

Specifically, to determine the overall rate R_i at any decision period i, the video source first uses the difference in the average RTT to determine the network "state". If the RTT is on average increasing, then we can assume that the network is becoming more congested and we should decrease the video rate. Conversely, if the average RTT is decreasing, then we can assume that there is some unused capacity than we can take advantage of to increase the encoding rate. The magnitude that the node increases or decreases the encoding rate is determined by the magnitude of the RTT change as well as the current video encoding rate. Intuitively, this is used to *weight* the flows so that the rates of low-quality flows increase faster than those of high-quality flows, and that high quality flows decrease faster than low quality flows. Overall, this has the effect that nodes are unlikely to stay in a low-quality state for an extended period of time.

While we are able to show in [27] (and in [28] for compressed-sensing based video encoding) that DMRC (and C-DMRC) outperform traditional rate controllers, this comes at the cost of system complexity. This protocol is designed to work directly with the video encoder, forcing the sender to encode the video real time, which may be infeasible for many large-scale systems. However, in systems (such as Netflix) that already pre-encode video at many different encoding rates, there is some potential to adaptively select the encoding rate *on-the-fly* to adapt to changing network conditions.

V. Routing Protocols

Transmitting video streams over networks with multiple wireless hops (MANET and sensor networks mentioned in Section II) is more challenging, where routing, aka path selection, plays a critical role. Different from traditional *application-agnostic* routing policies, the state-of-the-art video-application-centric routing schemes need to consider user-perceived video quality, playback deadlines, and energy consumption rather than hop count and throughput as design metrics. In this section, we will discuss the state-of-the-art video-specific routing protocols that are suitable to the following three types of video applications: video on demand, real-time video services and multimedia surveillance. To be specific, three representative video-centric routing approaches for each aforementioned application will be discussed in detail: i) *minimum distortion routing* to assure the quality requirement of VoD, ii) *packet-delay-deadline-aware routing* to satisfy the stringent delay for real-time video services, and iii) *power efficient routing* to prolong the life time of multimedia surveillance sensors deployed in WMSNs.

A. Minimum-Distortion Routing for Video Streaming

Video on Demand as discussed in Section II is usually delay-tolerant. This is because video frames are pre-coded at the server end and may be buffered before being played out at the receiver. However, because of the ever-increasing high entertainment demands of users, routing protocols with assured minimum distortion are needed. Some existing routing schemes [35]–[37] have been proposed to maximize the received video quality for video transmission over multi-hop wireless networks. We start the discussion by looking at the Minimum Distortion Routing (MDR) [37] scheme, where an existing rate-distortion model is used to evaluate the impact of packet loss rate on end-to-end video quality.

Distortion model based on frame loss probability. In wireless multi-hop networks, video quality is mainly impacted, in addition to the compression strategy at the source, by frame losses (impacted by Packet Loss Rate (PLR)). Therefore, how to choose an accurate distortion model with respect to frame loss is an important design consideration. In [37], the authors consider video frames with Group of Pictures (GOP) structure that consists of an I-frame followed by $(F-1)$ P-frames. A temporal-distance-based distortion model is considered, where distortion for the entire GOP is estimated, for the case in which $i \leq F-1$ frames are lost, as

$$D^{(i)} = (F-i) \cdot \frac{i \cdot F \cdot D^{min} + (F-i-1) \cdot D^{max}}{(F-1) \cdot F}, \quad (1)$$

where $D^{max} = D^{(0)}$ is the resulting distortion when the first I frame is lost, and $D^{min} = D^{(F-1)}$ is the distortion if the last P frame is lost, respectively. A loss probability distribution function $\boldsymbol{\pi}_t^k$ of kth frame is measured at each hop t based on a transition matrix, which represents a mapping from packet loss rate to frame loss rate. Since the value of distortion D at hop t depends on the position of the first unrecoverable frame in the GOP, therefore, the abovementioned loss probability distribution function $\boldsymbol{\pi}_t^k$ can be adopted to calculate the probability of the first lost frame in the GOP and hence, further specify the distortion D. Finally, a modified distortion model based on $\boldsymbol{\pi}_t^k$ is proposed. Readers are referred to [37] for further details.

MDR routing design. The algorithm then selects the optimal path $\boldsymbol{x}^*$ from a set of candidate paths $\mathcal{X}$ by minimizing the expected cost C, i.e., $\boldsymbol{x}^* = \arg\min_{\boldsymbol{x} \in \mathcal{X}} C(\boldsymbol{x}, \boldsymbol{D}(\boldsymbol{x}))$, where $\boldsymbol{D}(\boldsymbol{x})$ represents the distortion estimated for path $\boldsymbol{x}$ according to the model discussed above. A dynamic programming approach is used to solve the optimal routing problem discussed above. The MDR algorithm examined here therefore adopts an end-to-end video distortion model as the routing metric.

B. Packet-Delay-Deadline-Aware Routing for Real-Time Video Transmission

Video conferences and online gaming are typical applications belonging to the *real-time video* traffic paradigm category. In such scenarios, constraining the end-to-end delay is the most important design consideration, followed by the perceived video quality, which should be as high as possible. The work in [38] proposes a packet-delay-deadline-aware routing protocol to meet strict video playback deadline constraints while assuring the best user-perceived video quality. The technical approach is based on jointly selecting the best packet routing path and video encoding parameters (e.g., quantization step (QP) size).

Packet-delay-deadline-aware expected distortion. The design objective here is to choose the best transmission path and optimal video encoding parameter (i.e., QP) for each slice[3] $i \leq I$, where I is the total number of slices that compose the current video frame to be transmitted, so as to minimize the total expected distortion, under the constraint that the transmission delay T_i of each slice i is lower than a pre-defined frame decoding deadline T^{budget}, i.e.,

$$\begin{aligned} &\underset{S_i,\xi_i}{\text{minimize}} \sum_{i=1}^{I} E[D_i] \\ &\text{subject to } \max\{T_1, \ldots, T_I\} \leq T^{budget}, \end{aligned} \quad (2)$$

where S_i and ξ_i are source coding parameters and the selected transmission path for packet i, and $E[D_i]$ is the resulting expected distortion. To evaluate $E[D_i]$, the authors elaborate a distortion model based on packet loss probabilities considering i) packet drop probability due to violating the predefined decoding deadline and ii) packet error probability caused by lossy wireless links. An M/G/1 queue model is adopted to estimate the packet delay.

By explicitly introducing packet arrival deadline constraints in the optimization problem in (2), the proposed routing algorithm guarantees good perceived video quality while attempting to satisfy stringent delay requirements for real-time video services. Other routing approaches that consider packet delay constraints in routing optimization for video streaming services over multi-hop wireless networks include the Multi-path Multi-SPEED (MMSPEED) protocol [39] and the delay-bounded energy-constrained adaptive routing (DEAR) protocol [40].

C. Power Efficient Multimedia Routing

Energy conservation is another important consideration in wireless networks, especially networks (i.e., MANET, Tactical MANET and sensor networks) deployed with inconveniently replaced and battery-powered nodes, where the aforementioned multimedia surveillance is a typical application greatly desiring

[3]In this work, a slice consists of a number of encoded macroblocks in a certain raster scan order within one video frame.

high energy efficiency [14] due to significant energy consumption caused by large volume video acquisition and processing. For this reason, several energy-efficient route selection schemes have been proposed for streaming video traffic, including the Energy-Efficient QoS Assurance Routing (EEQAR) in [41], Hierarchical Energy Aware Protocol for routing (HEAP) [42] and Power Efficient Multimedia Routing (PEMuR) protocol [43]. In particular, PEMuR in [43] is designed with the dual objective of reducing the energy consumption and of improving the perceived video quality in wireless video communications.

Energy efficient hierarchical routing protocol. Here, we discuss the energy-minimized routing scheme - PEMuR proposed in [43], suitable to multimedia surveillance application. In PEMuR, a hierarchical routing selection scheme adopting the current residual energy of each sensor node as a metric is proposed, attempting to prolong the life of the entire network. Specifically, the considered hierarchical model includes a base station and a set of homogeneous sensor nodes, randomly distributed within the area of interest. The topology is assumed to be static.

All nodes are grouped into clusters. One node in each cluster is elected as cluster head by the base station based on residual energy. Cluster heads are classified into two categories: a) Upper level cluster heads and b) Lower level cluster heads, depending on whether the nodes can directly transmit information to the base station or not. The routing path between the lower level cluster heads and the base station is selected based on

$$p_k = \max_{l \in A} RI(p_l), \tag{3}$$

where A is the set of all possible paths from one cluster head to the base station, $RI(p_l)$ is the Routing Index (RI) defined as

$$RI(p_l) = \sum_{i=2}^{n-1} E_{r_i} - \sum_{i=1}^{n-1} E(p_l, c_i, c_{i+1}), \tag{4}$$

where E_{r_i} denotes the residual energy of the cluster head node c_i and $E(p_l, c_i, c_{i+1})$ is the energy required to route message between the node c_i and c_{i+1}.

By solving the optimization problem (3), the appropriate path from the cluster head to the base station is determined. In case the transmission rate exceeds the available bandwidth of the selected path, the combination of video packets resulting in minimum distortion will be dropped based on a distortion prediction model [44]. Therefore, this approach is particularly appealing for multimedia surveillance application distributed in hierarchical structure.

D. Future Work

Adaptive video-centric routing. As the amount of video content increases in wireless networks, there still does not exist an ideal routing protocol that can help effectively improve the perceived quality by jointly considering the application-specific requirements, such as timeliness, end-to-end quality, and energy consumption. To cater for the explosion of video-based content in wireless networks, recently, a clean architecture approach named as Information-Centric Networking (ICN) has drawn significant attention [45], which aims at providing a general infrastructure that provides in-network caching so that content is distributed in a scalable, cost-efficient manner. However, how to select the path to request video content that may have been cached in networks or not to satisfy diverse video-specific requirements is greatly unexplored so far. Therefore, developing a more efficient routing scheme for video transmission in information-centric networks may be another possible future direction of research.

Routing for CS-based video. So far, all routing protocols discussed above are designed for wireless video transmission adopting traditional video encoding paradigms discussed in Sections III-A and III-B. However, recently, compressive sampling based video applications (e.g., CS-based multi-view video streaming [23]) have been emerging [24], which are different from conventional video encoding methods. We refer the readers to Section III-D for more details. As discussed in Section III-D, CS-based video encoding paradigms are more resilient to channel error but with lower compression rate compared with predictive coding approach. However, in multi-hop wireless networks, the advantage of error resilience will be weakened. Hence, how to select the best path for CS-based video streaming to make up the undesirable consequence of the above mentioned issue introduced by multi-hop links but without decreasing compression ratio is unexplored and challenging.

VI. MAC Layer Solutions

The Medium Access Control (MAC) layer aims at regulating channel access so that multiple wireless devices can share the same wireless link effectively and fairly. An efficient MAC protocol is essential for all video networking paradigms aforementioned in Section II. With the growing demand for multimedia applications, the MAC is expected to support various video applications with diverse Quality of Service (QoS) requirements. For example, real-time video is delay-sensitive, while large volume video data access poses challenges to VoD and multimedia surveillance applications. Moreover, in MANET and sensor networks, battery-powered devices require energy efficient access schemes. To address these challenges, a plethora of video-application-aware MAC protocols have been proposed [46]–[50]. MAC protocols are usually classified into two categories: contention-based and schedule-based. Below, we examine two MAC schemes: schedule-based priority-guaranteed real-time video transmission access [49] and contention-based massive video traffic access [50] to address delay sensitive and energy efficiency high-volume transmission requirements.

A. Collision-Free MAC With Priority Guaranteed for Real-Time Multimedia Transmission

Conventional schedule-based (especially TDMA-based) MAC protocols are often used in wireless video transmission because they can guarantee collision-free and QoS-compliant transmissions [49] but with hard-to-control end-to-end delay. First we will discuss a representative TDMA-like MAC scheme for wireless mesh backbone network proposed in [49] which attempts to provide priority guarantees for real-time video applications besides assuring the required quality-level. Next, we examine two key techniques of the protocol: a) *guaranteed priority access for real-time video traffic* and b) *improved per-flow fairness*.

Guaranteed priority access for real-time video traffic. The core idea of priority-guaranteed access policy for real-time video steaming is to use a novel distributed time slot structure

that includes a control segment and transmission segment. The control segment consists of mini-slots, further divided into real-time mini-slots with descending urgency level U_i, and non-real-time mini-slots, allocated to each node to contend the corresponding slot if there are packets to send. For real-time priority access, the urgency level of the real-time packet j is measured by the value $\frac{t_j}{n_j}$ (with t_j and n_j being the remaining time and remaining hops). If $U_{i-1} \leq \frac{t_j}{n_j} \leq U_i$ and no other jamming signal is detected, the node owning packet j will win the corresponding slot and later transmit that packet in the collision-free transmission slot. Therefore, the proposed scheme guarantees that nodes with real-time video packets have priority access over those with delay tolerant packets. Moreover, a mini-slot reuse policy is proposed by allowing two devices with more than two-hop distance to use the same mini-slot. Based on this, the duration period of the contention phase is only dependent on the number of devices in a two-hop neighborhood but not on the total number of nodes in the entire network, which overcomes the limited scalability of some schedule-based MAC protocols.

Improved per-flow fairness. Fairness among end users retrieving videos (per-flow) or intermediate nodes routing the videos (per-node) is another important factor for scheduling design. In [49], per-node fairness is achieved by using the mini-slot rotation scheme, based on which, per-flow fairness is implemented as follows:

First, nodes need to exchange traffic information (e.g., the number of flows) with their one-hop and two-hop neighbors. Then, nodes redetermine the fraction of time to access the channel in proportion to the number of flows being transmitted. Although resources are allocated more fairly to flows, this introduces extra control overhead required to exchange traffic information with neighbors. Therefore, there is a tradeoff between per-flow fairness and efficient resource allocation.

To summarize, the idea of using real-time rotating mini-slots into a schedule-based MAC proposed in [49] achieves collision-free transmissions as well as some application-specific requirements in terms of priority-guarantees and fairness for real-time video streaming.

B. Massive Video Transmission Assured MAC Protocol

In the MAC layer of WMSNs, energy efficiency and large-scale transmission are required for delivering multimedia data which can lead to high traffic. Recently, some contention-based MAC policies for high-volume video data transmissions have been proposed [50], [51]. Next, a representative scheme - Massive Transmission Scheme (MTS) [50], is examined in detail.

MTS: contention-based massive MAC. MTS is a novel contention-based massive transmission scheme designed to support massive multimedia data delivery in WMSN systems. Specifically, the proposed scheme operates in two modes: a) Massive Transmission (MT) mode, b) Normal Operation (NO) mode. In the MT mode, fast, continuous and massive video data transmission in a multi-hop environment with a reduced number of ACK frames and contention time is proposed. Moreover, a Network Allocation Vector (NAV) is employed to reduce unnecessary listening energy consumption, avoid collisions and solve the hidden terminal problems caused by half-duplex.

Switch method between MT and NO. To concurrently support video streaming and common data transmission, a switch criterion is proposed for switching between *MT* and *NO* modes. For this purpose, a dynamic SDTL (Short Data Threshold Length) based switching criterion is proposed given a value λ_t at time t, expressed as

$$\lambda_t = \alpha \cdot (C_t - C_{t-1}) + \lambda_{t-1}, \tag{5}$$

with C_t and C_{t-1} being the number of packets at time t and $t-1$, respectively, and α is an empirically predefined coefficient. Then, the value of SDTL is updated as $SDTL_t = SDTL_{t-1} + \lambda_t \cdot l_{MTU}$, where l_{MTU} denotes the length of maximum allowed transmission unit (MTU). If the length of the remaining data in the queue is higher than $SDTL_t$, MT mode will be started, otherwise the system stays in NO mode.

The idea to massively and successively transmit video data by reserving the wireless link for some dedicated time can reduce the end-to-end delay. However, a switching criterion based only on the amount of data in the buffer is insufficient to assure a guaranteed level of perceived video quality because the time-varying channel condition may cause drastic throughput degradation. Therefore, the MTS access policy still may be enhanced by considering a combination of application-aware requirements (e.g., playback deadline) and instantaneous channel conditions into the switching threshold.

C. Future Work

Adaptive and application-centric MAC protocols. As discussed in Section II, various video applications may traverse heterogeneous wireless architectures with diverse QoS requirements. These heterogeneities must be dynamically handled by mobile terminals. Thus, the integration of the existing medium access schemes aforementioned spotlights an adaptive and seamless medium access control [51] layer that can meet diverse QoS requirements and improve network utilization.

Due to many shortcomings of current Medium Access Control protocols, including hidden-terminal-like problems caused by the half-duplex nature of current wireless devices, full-duplex radios [52], [53] are emerging, which can be considered as another promising approach to develop new MAC schemes to satisfy the diverse QoS requirements of various kinds of video applications. However, self-interference cancellation is the major challenge to design full-duplex MAC schemes and how to integrate it with video transmission is substantially unexplored.

VII. Physical Layer Solutions

Wireless links provide only limited and time-varying data transmission rates, and may hence become the bottleneck in mobile video applications. This is conventionally alleviated by adopting techniques like joint source and channel resource allocation [54]–[56] or streaming video content in a scalable fashion [57], [58]. However, given the ever-increasing video traffic and the low-energy consumption requirements, wireless video streaming is far from being a mature technology. This has stimulated additional research efforts to redesign the video encoder/decoder to adapt more smoothly to the time-varying channels with even higher received video quality, e.g., by leveraging the channel state information (CSI) with a finer granularity or integrating video streaming tightly with newly-emerging physical layer technologies. While a comprehensive discussion on physical layer techniques for video streaming is clearly beyond the scope of this paper, in what follows we rely on three representative technologies to discuss

newly emerging solutions and challenges: i) *soft coding and decoding* to exploit error resilience in wireless video streaming [59], ii) *cooperative video streaming* to enhance transmission reliability or rate [60], and iii) *compressive and cooperative video streaming* to achieve both error resilience and low-power transmissions [61]. While the first approach is mostly suitable for cellular networks and WLANs, the latter two can find more broad applicability, including in wireless ad hoc and sensor networks.

A. Soft Video Coding and Decoding

In traditional video encoders, the quantized DCT or Wavelet coefficients are in general encoded using *variable length coding (VLC)* [62]. In the case of bit flips caused by errors in wireless transmission, this may cause synchronization problems in decoding VLC-coded packets (aka "all-or-none" problem). Moreover, the wireless channel is usually quite different for each video terminal, and the decoding capabilities for each terminal can also be different. As a result, without perfect adaptation of transmission schemes} (which is not easy to achieve because of the unpredictability of the time-varying wireless channel quality), wireless video streaming may suffer from severe cliff effects and be rather choppy. In [59], [63]–[65], the authors address this challenge by proposing soft video coding and decoding, which, roughly speaking, leverage soft physical-layer information like bit-level probability belief (instead of only the hard binary channel-decoding-results), or use linear analog coding (instead of variable-length digital coding) for video compression and channel error protection. Next, we discuss FlexCast [59] as an example. The core design objective of FlexCast is to gracefully adapt the reconstructed video quality to the time-varying channel quality, and hence avoid cliff effects.

FlexCast: constant-length representation. The core idea of FlexCast is to use rateless coding (whose benefits are discussed later) to protect constant-length-encoded coefficients (rather than VLC), and then decode the received, possibly noisy, coefficients by integrating soft information provided by physical-layer channel decoding.

The motivation for using constant-length coding (e.g., represent each quantized coefficient using, say 8 bits) is that this can completely avoid the problem of possibly losing synchronization in decoding VLC-encoded bit sequences. To achieve high bandwidth efficiency, FlexCast assigns the available bit rate budget among the resulting raw bits in an unequal fashion based on rateless channel coding. For this purpose, the raw bits are grouped into multiple clusters with different importance levels according to their contribution to the overall reconstruction error if they are not successfully decoded. In [59], the authors use the bit positions in each 8-bit coefficient to indicate their importance, e.g., the MSBs are more important than the LSBs since flips occurred at MSBs cause larger deviation to the original coefficient value, and hence possibly larger video reconstruction distortion.

Rateless coding and soft decoding. Given the allocated bit rate budget, error protection bits are generated for each group based on rateless coding and then appended to the raw bits. Advantages of rateless coding include i) it may produce coded bits with arbitrary, and hence very flexible, channel coding rate; and ii) by decoding the rateless coded bits, the receiver is able to compute a *soft information* for each raw bit, i.e., the probability that a bit is "1" or "0" (rather than a hard decision). Denoting $p_1(i)$ as the resulting probability that the ith bit in the representation of a DCT component is "1", FlexCast then estimates the DCT component as

$$y = \sum_{i=1}^{L-1} 2^i \cdot p_1(i), \tag{6}$$

where L is the quantization bit-depth. Since $p_1(i)$ contains soft information about the ith bit, the reconstruction error with DCT component y is allowed to gracefully adapt to the variations of channel coding rate and channel quality, with stronger error protection and better channel quality leading to lower distortion.

The idea of exploiting "*soft information*" to enable noise (error) resilience has also been leveraged in SoftCast [63], its MIMO-OFDM extension ParCast [64], and also in D-Cast designed for distributed video coding [65]. There, the available transmission power budget is allocated among DCT coefficients proportionally to their magnitude levels, e.g., through magnitude scaling and DCT-chunk transformation in [63]. The authors in [59], [63]–[65] show that *soft video streaming* enables the reception video quality to scale gracefully with wireless channel quality, and in practice be higher than that of non-soft ones; this graceful reception characteristic can also be achieved using the CVS encoder introduced in Section III-D. Hence, soft video streaming may provide a promising solution for those mobility-rich video applications where the channel quality may change very fast, like VoD on smart phones and in-vehicle VoD, and video broadcasting and multicasting applications where video terminals may experience different wireless channels hence having different decoding capabilities.

B. Cooperative Video Streaming

Instead of redesigning video encoders, a second possible approach to address the choppiness of wireless video streaming is to enhance the transmission reliability, so that good or at least acceptable video quality can be received for most levels of channel quality. This approach is particularly appealing for applications like video surveillance with wireless sensor networks, where it is not easy to run intelligent but potentially very complex video encoders in battery-powered sensors due to energy and computational capability limitations.

At the physical layer, emerging technologies with potential for improved reliability include cooperative relaying [66] and interference alignment [67]. While a detailed treatment of these approaches is out of the scope of this paper, we take cooperative relaying as an example to show how these technologies can be used to enhance wireless video quality [60], [68].

Cooperative relaying. Cooperative relaying techniques attempt to leverage the spatial diversity of the wireless channel in a distributed fashion. While this is traditionally done by relying on multiple transceiver antennas, it may not be practical to implement this on sensor nodes usually with only limited size. Instead, cooperative relaying relies on antennas of neighboring devices to form a virtual multiple-input-single-output (VMISO) link and hence to achieve spatial diversity [66].

A cooperative transmission is typically completed in two consecutive time slots. In the first time slot, the source node broadcasts information to both destination and potential relay nodes,

and in the second, the relay node forwards the received information to the destination. The resulting cooperative link capacity C_{cop} can be expressed as

$$C_{\text{cop}} = \frac{1}{2} C_{\text{two\_slot}}, \tag{7}$$

with $C_{\text{two\_slot}}$ being the capacity achievable through combining at the destination signals received in the two time slots.

Capacity-matched video streaming. In [60], we studied a cooperative video streaming network, which consists of a set of video sensor nodes transmitting the captured video sequences to their own intended destinations, either through a direct link or through cooperative relaying. In the latter case, the source node optimally selects a relay from a set of potential relay nodes for cooperative transmission, resulting in an overall capacity expressed in (7). Due to the coefficient $\frac{1}{2}$ there, the capacity C_{cop} can be higher, or lower than that using only direct transmission. Hence, at physical layer, it is important to decide for each session, i) whether to transmit using a cooperative relay or using the direct link only, and ii) which relay should be selected in the former case.

At the same time, at the application layer, the video encoding rate is jointly controlled to match the resulting physical-layer link capacity. While a low video rate causes high encoding distortion, too high a video rate may potentially make the network more congested causing high packet drop rate caused by exceeding the video playout deadline. In [60], we formulated the problem of optimal joint relay selection and rate control in a multi-user wireless network as a mixed nonlinear, nonconvex combinational problem (MINLP), and solved it through newly designed distributed and centralized algorithms. We found that a noticeable gain in sum PSNR can be achieved through cooperative relaying with in practice even lower average transmission power compared to using direct transmissions only. The effectiveness of cooperative relaying is also demonstrated in [68] by considering uplink wireless video streaming in cellular networks.

C. Compressive and Cooperative Video Streaming (CCVS)

Finally, we discuss CCVS, an approach exploiting both error resilience and enhanced transmission for wireless video streaming [61]. Different from using complex redesigned video encoder as discussed in Section VII-A, in [61] we showed that error resilience can also be achieved while keeping video encoders very simple, hence facilitating video applications on resource-constrained devices with limited computational capabilities, e.g., video surveillance using wireless sensor networks discussed in Section II. The core idea of CCVS is to leverage the inherent error resilience properties of compressive sampling discussed in Section III-D at the application layer, and at the physical layer, to develop cooperative wireless networking based on the unique properties of video representation with compressed sensing.

Error resilient compressive sensing (ERCS). ERCS was first formalized in [69], saying that a bit-vector can be exactly recovered from a linearly encoded and then sparsely-error-corrupted version of the vector, by solving an ℓ_1-minimization problem. While ERCS may work well in principle, the additional communication overhead introduced by the linear encoding can however be quite high for wireless video streaming. This is because i) quantizing real-valued CS samples may add additional quantization noise to the signal, and ii) removing all of the bit errors in CS samples requires getting every reconstructed sample exactly correct, this may require excessive overhead if only non-important bits (e.g., LSBs) are flipped.

In [61], we showed that both of these problems can be avoided by i) using real valued (unquantized) CS samples to create the error correction samples, and ii) using mean square error (MSE) of the reconstructed CS samples as performance metric. Compared to bit error rate (BER), MSE can naturally weight the significance of the small scale (e.g., LSB) errors less than more important (e.g., MSB) errors, and hence needs less parity bits.

We showed that, with ERCS and cooperative networking, the reconstructed video quality in terms of SSIM can be considerably improved. This has the important consequence of potentially enabling systems that can transmit video at SNR values that are a fraction of traditional cooperative relaying systems without sacrificing video quality, hence enabling extra low-power video sensors.

D. Future Work

Softer video streaming. While different schemes have been studied to exploit error resilience and hence to achieve soft video streaming, there are still several challenges to address. i) Since the discussed FlexCast and ParCast are limited to unicast, and D-Cast targets only Gaussian channels, soft video streaming schemes still need to be designed for multicast, broadcast and P2P video streaming, and by considering fading channels; ii) SoftCast, ParCast and CCVS exploit only partial temporal correlations in video encoding hence possibly causing waste of bandwidth. There is still significant room for creating encoders with higher bandwidth efficiency; iii) While all the discussed works focus on 2D videos, designing soft streaming systems to support the emerging 3D/Multi-view/Stereo video applications [70] can be another potential research direction.

VIII. Cross Layer Solutions

We conclude our discussion by focusing on cross-layer design work that has concentrated on joint optimization of application and multiple lower layers. We discuss several representative examples of cross-layer video streaming with different global design objectives: i) maximize the quality of experience (QoE) of users $(\text{application} + \text{Transport} + \text{MAC})$, considering cellular networks and WLANs [71], ii) minimize system energy consumption (application + MAC + physical) [9], considering wireless ad hoc or sensor networks, and iii) meet the stringent delay constraint $(\text{application} + \text{network} + \text{physical})$, considering hybrid cellular/ad hoc networks (or cellular networks with device-to-device (D2D) communications enabled) [72]. In addition to discussions about different design objectives, we also highlight the potential advantages of incorporating CLD in those newly emerged video applications such as cloud-assisted wireless video gaming discussed in the last example. Readers are referred to [73], [74] and references therein for comprehensive surveys of this area.

A. DASH With Channel-Content-Aware Rate Control

As discussed in Section IV, TCP connections have been used in several commercial streaming systems such as YouTube and Netflix. In wireless video streaming with remotely located video servers, server-to-user TCP connections may behave quite

differently in their wired and wireless components in terms of average throughput, jitter, and delay. This may potentially cause severe network congestion (when source rates are too high) or bandwidth waste (if the selected video rate is too low). To address this challenge, the authors of [71] proposed WiDASH, a DASH (Dynamic Adaptive Streaming over HTTP [75]) framework to jointly coordinate the wired and wireless TCP connections for multiple concurrent users. Different from the TCP performance enhancement proxy (PEP) discussed in Section IV-B, which focuses primarily on the adaptability of a single TCP proxy to wireless channel quality, here we discuss WiDASH and its application in optimization of multiuser video streaming systems with concurrent TCP links.

WiDASH via TCP splitting. WiDASH splits a long TCP connection into cascaded multiple short TCP connections, and hence several problems related to the wireless networks can be effectively isolated from the wired, e.g., the higher transmission error and unpredictable time-varying channel quality. Taking 3G UMTS as an example, WiDASH can be located at the Gateway GPRS Support Node (GGSN), which splits the server-to-user TCP link into two shorter ones, one between the server and the WiDASH proxy, and the other between the WiDASH and mobile users. Consequently, it is easier for the proxy to collect in real time the link quality information for both shorter links, and use the collected information for optimally coordinating the wired and wireless transmissions.

WiDASH-based optimization. In 3G UMTS cellular networks, concurrent users are scheduled by differentiating two priority levels, say $\{p_{\text{low}},\ p_{\text{high}}\}$. To assure QoE fairness between high-rate and low-rate video users, and between video traffic and background traffic (e.g., file downloading), it is desirable to dynamically assign different priority to each video user and to each packet for a single user. Intuitively, this can help prevent a video user from requesting a high streaming rate and at the same time that its packets are assigned high priority.

For this purpose, WiDASH adopts a linear-mapping-based priority-assignment framework, with mapping function

$$w(r) = \begin{cases} 0, & r \le r_{\text{low}} \\ \frac{x - r_{\text{high}}}{r_{\text{high}} - r_{\text{low}}}, & r_{\text{low}} < r \le r_{\text{high}} \\ 1, & r > r_{\text{high}} \end{cases} \tag{8}$$

where r is the video rate associated with the packet, r_{low} and r_{high} are two thresholds representing low and high rates, respectively. WiDASH assigns high priority p_{high} to the packet with probability $1-w(r)$, and low priority p_{low} with probability $w(r)$; here, the random priority mapping is used to avoid TCP synchronization among different DASH flows.

In this framework, WiDASH then maximizes the average QoE of the users, by i) coordinating the concurrent wired TCP links through jointly deciding the video rate for each user, and ii) scheduling transmissions over the wireless TCP links based on real-time observation of the wireless channel quality. It is shown WiDASH is able to support higher video rate and smoother rate changes for each individual user compared with the competing scheme Akamai [71].

B. Rate-Energy-Distortion-Predictable Video Streaming

Different from the discussions above, where the objective of WiDASH is to maximize the system QoS in cellular video streaming systems, here we concentrate our discussion on energy-efficient video streaming based on CLD, considering differential compressive video sensing (D-CVS) [9]. Here, D-CVS means integrating inter-frame prediction (the prediction will be explained later in this section) into the compressive video sensing (CVS) introduced in Section III-D; D-CVS has been shown to have a potential to reduce the required energy and computational complexity for video capturing, processing and streaming [9], and hence to boost applications like video surveillance in wireless multimedia sensor networks discussed in Section II and ocean exploration in the envisioned Internet underwater [76].

We discuss the rate-energy-distortion behavior of D-CVS, considering the interactions among application-layer video encoding, MAC-layer error handling, and physical-layer SNR adaptation. An illustrative scenario is given in Fig. 5, where the energy budget available at the source node is split between the energy needed for video encoding and transmission.

Differential compressive video sensing (D-CVS). The individual-frame-based CS encoding takes advantage of the spatial correlation within each video frame, in [9] we proposed D-CVS to exploit the inter-frame correlation and in the meanwhile avoid complex motion estimation operations. Consider a group of pictures (GOP) patterned as $IPP\cdots P$. Then, D-CVS encodes the I-frame in traditional intra fashion, while encoding the ith P-frame $\mathbf{y}_i$ through encoding the difference vector $\mathbf{dv}_i$ between the frame and a selected reference frame $\mathbf{y}_{ref}$ computed as $\mathbf{dv}_i = \boldsymbol{y}_i - \boldsymbol{y}_{ref}$. The advantage of doing so is that (as discussed in Section III-D), in the case of low- or moderate-level motion, consecutive two or several frames do not differ too much from each other, and the resulting difference vector might be much sparser than the frame themselves. Then, the data rate needed to transmit over the lossy channel can be considerably reduced, to achieve certain reconstructed video quality.

Error-level-adaptive packet dropping. As discussed in Section I, keeping bad samples (i.e., with bit errors) may only slightly degrade the reconstructed video quality in the case of low bit error rate (e.g., lower than 10^{-3} in Fig. 1). This implies that, retransmitting a sample packet is necessary only if the BER is high. Then, together with considering the effects of D-CVS discussed above, the video quality can be empirically modelled using a low-pass-filter function of encoding rate r_v and transmission rate r_{ch},

$$U(r_{ch}, r_v) = \frac{D_0 - \frac{\theta}{r_v - R_0}}{\sqrt{1 + \tau^2 (BER(r_{ch}, r_v))^2}}, \tag{9}$$

where D_0, θ and R_0 are video dependent constants that can be determined through linear least squares estimation techniques, τ is the encoder dependent constant used to indicate the quality degradation level with respect to $BER(r_{ch}, r_v)$. Then, given an energy budget, the optimal rate allocation r_v and r_{ch}, and hence the optimal energy allocation can be obtained by solving a nonlinear optimization problem; here, the nonlinearity is due to the complex expression of (9). As shown in Fig. 5, D-CVS outperforms the competing schemes H.264 and MJPEG in a wide range of energy budget in terms of structural similarity (SSIM).

C. Cloud-Assisted Mobile Gaming

In addition to low energy consumption and high reconstruction quality as discussed above, some applications like mobile

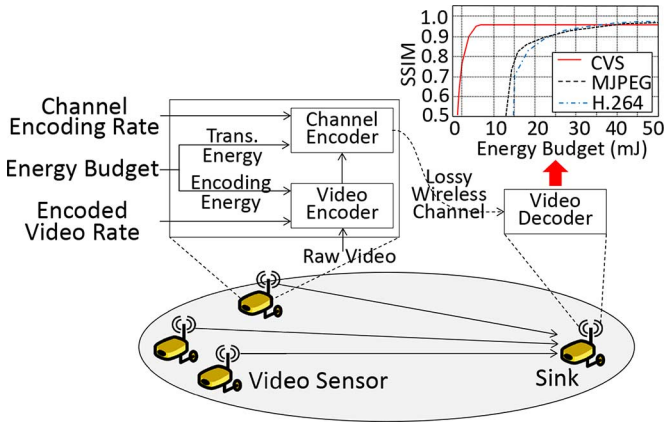


Fig. 5. Energy-aware video encoding and transmission [9].

video gaming also needs to meet a very stringent delay requirement. However, both the processing capability of mobile devices (e.g., smart phones, glasses) and the bandwidth of wireless links are only limited, which may result in unacceptable processing and transmission delay.

Cloud computing technology has emerged with the potential to enable high-quality and energy-efficient wireless video streaming, while still meeting the stringent delay requirement [77]; and hence to enable a wide set of multimedia-rich and mobility-rich applications, like mobile video gaming, 3D/Stereo/Multiview video streaming. Next, we discuss cloud-assisted cross-layer optimization, considering mobile gaming applications as an example [72].

Rate minimized video encoding. In [72], the authors considered a gaming network that consists of a game cloud and a set $\mathcal{N}$ of mobile players. Different from traditional game servers, which respond to player commands by rendering and streaming their video content separately, the game cloud uses an additional video encoder server to exploit inter-player correlation in favor of higher encoding compression ratio.

For this purpose, in [72] the authors first group game players into different groups, e.g., players closely located *in the game* can be grouped into one group, since they are seeing similar game scenes.[4] Then, the video sequences in each group can be encoded in the same way of encoding multi-view 3D videos [70] which captures the same scene of interest but from different eyes. Specifically, frames are encoded in three modes, $\{I,\ P_{intra},\ P_{inter}\}$, to exploit the intra-frame spatial correlation, inter-frame temporal correlation, and the inter-view correlation, respectively. Experiments of encoding natural videos showed that, exploiting the inter-view correlation may reduce up to 50% the encoder output rate. By searching for the optimal player grouping scheme and frame encoding modes using the powerful video cloud, the authors of [72] showed that this rate reduction can be up to 70% for the considered video gaming with 10 game players.

Delay reduction through multi-hop avoidance. Since the video sequences are encoded in a correlated fashion, players still need, e.g., through WiFi- or bluetooth-based ad hoc links, to share their reconstructed video frames with other players in the same group if their frames were selected as references. This implies that a player cannot decode his/her P_{inter}-encoded frames before receiving the reference frames from other players, and as a result this causes one-hop processing and transmission delay. If cross-reference encoding is used in the cloud, i.e., an I_{inter} encoded frame of a player is further used as reference for other players, the delay will accumulate along the multi-hop video content sharing path. In [72], the authors avoid this by simply disabling the cross-reference encoding. This however takes a cost of around twice video encoding output rate compared with that of multi-hop. Therefore, it is still necessary to explore the tradeoff between low video encoding rate in the cloud server and the possible high multi-hop sharing delay among players, by jointly considering game scene characteristics, location-aware multi-hop sharing, and also wireless channel quality in a cross-layer manner.

[4]In [72], the authors concentrated on the third-person game, where players watch the whole game scene in a bird-view. Examples of such game are Diablo, Command&Conquer.

D. Future Work

Cross-layer design for energy-efficient 3D video streaming. *3D/Multi-view/Stereo* video applications have recently emerged as services with a potential to offer a higher Quality of Experience (QoE) compared with conventional 2D video [70]. However, the computational complexity of encoding 3D multi-view video and transmitting the encoded data may result in a high energy burden for mobile devices, which ultimately leads to short operational lifetime. It is therefore essential to design novel transmission schemes, e.g., in cross-layer manner, and clean-slate network architectures with higher energy efficiency by integrating compressive sampling technology [24].

Cloud-assisted mobile video streaming. Another potential direction is to integrate mobile cloud computing (MCC) technologies with wireless video streaming [78]. Then, mobile devices can continuously offload their computationally-intensive tasks to a remote cloud server, hence potentially extending the battery lifetime. On the other hand, by optimizing the streaming strategies using the powerful cloud server, this may enable real-time, network-friendly and scalable video streaming. Through cloud-enabled mobile networks, we envision that high quality mobile video streaming will be supported without considerably increasing the energy consumption.

IX. Conclusion

In this work, we discussed state-of-the-art video encoders and networking protocols for wireless video streaming. We first examined emerging video encoders, and discussed how compressed sensing and distributed systems could help enhance video systems where the video was created by resource constrained wireless devices. We then examined transport protocols, with special emphasis on techniques used by large-scale streaming services such as YouTube. Then, we discussed application-centric routing protocols for video applications including on-demand video streaming, real-time interactive video services and video surveillance in WSN. In MAC layer, channel access policies for high-volume video data transmission are examined. In physical layer, we discussed error-resilient video streaming, by highlighting soft video encoding/decoding and cooperative streaming. Finally, we examined cross layer solutions, by paying attention to newly emerging service architectures like DASH-based and cloud-assisted wireless video streaming.

REFERENCES

[1] I. U. Sandvine, Global Internet Phenomena Rep.: 2H 2013, 2013.
[2] [Online]. Available: http://www.netflix.com
[3] [Online]. Available: http://www.youtube.com
[4] C. Huang, J. Li, and K. W. Ross, "Can internet video-on-demand be profitable?," in *Proc. ACM SIGCOMM*, Kyoto, Japan, Aug. 2007.
[5] *Advanced Video Coding for Generic Audiovisual Services*, ITU-T Rec. H.264, 2005.
[6] B. Bross, W.-J. Han, J.-R. Ohm, G. J. Sullivan, and T. Wiegand, "High efficiency video coding (HEVC) text specification draft 6," in *Document JCTVC-H1003, 8th JCT-VC Meeting*, San Jose, CA, USA, 2012.
[7] X. Cheng, J. Liu, and C. Dale, "Understanding the characteristics of internet short video sharing: A YouTube-based measurement study," *IEEE Trans. Multimedia*, vol. 15, no. 5, pp. 1184–1194, Aug. 2013.
[8] V. Adhikari, Y. Guo, F. Hao, M. Varvello, V. Hilt, M. Steiner, and Z.-L. Zhang, "Unreeling Netflix: Understanding and improving multi-CDN movie delivery," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Orlando, FL, USA, Mar. 2012, pp. 1620–1628.
[9] S. Pudlewski and T. Melodia, "Compressive video streaming: Design and rate-energy-distortion analysis," *IEEE Trans. Multimedia*, vol. 15, no. 8, pp. 2072–2086, Dec. 2013.
[10] S. Pudlewski and T. Melodia, "On the performance of compressive video streaming for wireless multimedia sensor networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Cape Town, South Africa, May 2010, pp. 1–5.
[11] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
[12] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Comput. Netw. (Elsevier)*, vol. 38, no. 4, pp. 393–422, Mar. 2002.
[13] *IEEE Standard for Information Technology* , IEEE Std. 802.11b-1999/Cor 1-2001, 2001.
[14] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, "A survey on wireless multimedia sensor networks," *Comput. Netw.*, vol. 51, no. 4, pp. 921–960, Mar. 2007.
[15] *Video Coding for Low Bit Rate Communication*, ITU-T Rec. H.263.
[16] *Digital Compression and Coding of Continuous-Tone Still Images—Requirements and Guidelines*, ITU-T Rec. T.81, 1992.
[17] *JPEG2000 Requirements and Profiles*, ISO/IEC JTC1/SC29/WG1 N1271, Mar. 1999.
[18] A. Leontaris, Y. Tonomura, and T. Nakachi, "Rate control for flicker artifact suppression in motion JPEG2000," in *Proc. IEEE Conf. Acoust., Speech, Signal Process. (ICASSP)*, Toulouse, France, May 2006, pp. 41–44.
[19] *Generic Coding of Moving Pictures and Associated Audio Information: Video*, ISO/IEC 13818-2, ITU-T Rec. H.262, 1995.
[20] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inf. Theory*, vol. 19, no. 4, pp. 471–480, Jul. 1973.
[21] A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Trans. Inf. Theory*, vol. IT-22, no. 1, pp. 1–10, Jan. 1976.
[22] R. Puri and K. Ramchandran, "PRISM: A new robust video coding architecture based on distributed compression principles," in *Proc. Allerton Conf. Commun., Control, Comput.*, Allerton, IL, USA, Oct. 2002.
[23] N. Cen, Z. Guan, and T. Melodia, "Joint decoding of independently encoded compressive multi-view video streams," in *Proc. Picture Coding Symp. (PCS)*, San José, CA, USA, Dec. 2013.
[24] S. Pudlewski and T. Melodia, "A tutorial on encoding and wireless transmission of compressively sampled videos," *IEEE Commun. Surveys Tutorials*, vol. 15, no. 2, pp. 754–767, Second quarter 2013.
[25] D. Wu, Y. T. Hou, W. Zhu, Y.-Q. Zhang, and J. M. Peha, "Streaming video over the internet: Approaches and directions," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 3, pp. 282–300, Mar. 2001.
[26] A. Rao, A. Legout, Y.-S. Lim, D. Towsley, C. Barakat, and W. Dabbous, "Network characteristics of video streaming traffic," in *Proc. Conf. Emerging Netw. EXper. Technol. (CONEXT)*, Tokyo, Japan, Dec. 2011, pp. 25:1–25:12.
[27] S. Pudlewski and T. Melodia, "A distortion-minimizing rate controller for wireless multimedia sensor networks," *Comput. Commun. (Elsevier)*, vol. 33, no. 12, pp. 1380–1390, Jul. 2010.
[28] S. Pudlewski, T. Melodia, and A. Prasanna, "Compressed-sensing-enabled video streaming for wireless multimedia sensor networks," *IEEE Trans. Mobile Comput.*, vol. 11, no. 6, pp. 1060–1072, Jun. 2012.
[29] S. Alcock and R. Nelson, "Application flow control in YouTube video streams," *SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 2, pp. 24–30, Apr. 2011.
[30] H. Balakrishnan, S. Seshan, and R. Katz, "Improving reliable transport and handoff performance in cellular wireless networks," *ACM Wireless Netw. J. (WINET)*, vol. 1, no. 4, pp. 469–481, Dec. 1995.
[31] J. K. Sundararajan, D. Shah, M. Médard, M. Mitzenmacher, and J. Barros, "Network coding meets TCP," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Rio De Janerio, Brazil, Apr. 2009, pp. 280–288.
[32] H. Hisamatsu, G. Hasegawa, and M. Murata, "Non bandwidth-intrusive video streaming over TCP," in *Proc. Int. Conf. Info. Tech.: New Generat. (ITNG)*, Las Vegas, NV, USA, Apr. 2011, pp. 78–83.
[33] S. Mascolo, C. Casetti, M. Gerla, M. Y. Sanadidi, and R. Wang, "TCP westwood: Bandwidth estimation for enhanced transport over wireless links," in *Proc. ACM Int. Conf. Mobile Comput. Netw. (MobiCom)*, Rome, Italy, Jul. 2001.
[34] W. Pu, Z. Zou, and C. W. Chen, "New TCP video streaming proxy design for last-hop wireless networks," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Brussels, Belgium, Sep. 2011, pp. 2225–2228.
[35] X. Tong, Y. Andreopoulos, and M. van der Schaar, "Distortion-driven video streaming over multihop wireless networks with path diversity," *IEEE Trans. Mobile Comput.*, vol. 6, no. 12, pp. 1343–1356, Oct. 2007.
[36] B. Rong, Y. Qian, K. Lu, R. Hu, and M. Kadoch, "Multipath routing over wireless mesh networks for multiple description video transmission," *IEEE J. Sel. Areas Commun.*, vol. 28, no. 3, pp. 321–331, Mar. 2010.
[37] G. Papageorgiou, S. Singh, S. Krishnamurthy, R. Govindan, and T. Porta, "Distortion-resilient routing for video flows in wireless multi-hop networks," in *Proc. IEEE Int. Conf. Netw. Protocols (ICNP)*, Austin, TX, USA, Nov. 2012, pp. 1–10.
[38] D. Wu, S. Ci, H. Wang, and A. Katsaggelos, "Application-centric routing for video streaming over multihop wireless networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 12, pp. 1721–1734, Dec. 2010.
[39] E. Felemban, C.-G. Lee, and E. Ekici, "MMSPEED: Multipath multi-SPEED protocol for QoS guarantee of reliability and timeliness in wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 5, no. 6, pp. 738–754, Jun. 2006.
[40] S. Bai, W. Zhang, G. Xue, J. Tang, and C. Wang, "DEAR: Delay-bounded energy-constrained adaptive routing in wireless sensor networks," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Orlando, FL, USA, Mar. 2012, pp. 1593–1601.
[41] K. Lin, J. J. P. C. Rodrigues, H. Ge, N. Xiong, and X. Liang, "Energy efficiency QoS assurance routing in wireless multimedia sensor networks," *IEEE Syst. J.*, vol. 5, no. 4, pp. 495–505, Dec. 2011.
[42] M. Moazeni and A. Vahdatpour, "HEAP: A hierarchical energy aware protocol for routing and aggregation in sensor networks," in *Proc. Int. Conf. Wireless Internet (WICON)*, Austin, TX, USA, Oct. 2007.
[43] D. Kandris, M. Tsagkaropoulos, I. Politis, A. Tzes, and S. Kotsopoulos, "Energy efficient and perceived QoS aware video routing over wireless multimedia sensor networks," *Ad Hoc Netw.*, vol. 9, no. 4, pp. 591–607, Jun. 2011.
[44] I. Politis, M. Tsagkaropoulos, T. Pliakas, and T. Dagiuklas, "Distortion optimized packet scheduling and prioritization of multiple video streams over 802.11e networks," *Adv. Multimedia*, vol. 2007, pp. 1–11, Aug. 2007.
[45] L. Saino, I. Psaras, and G. Pavlou, "Hash-routing schemes for information centric networking," in *Proc. ACM SIGCOMM Workshop Information-centric Netw. (ICN)*, Hong Kong, China, Aug. 2013.
[46] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, New York, NY, USA, June 2002.
[47] Rajendran, Venkatesh, Obraczka, Katia, and J. J. Garcia-Luna-Aceves, "Energy-efficient, collision-free medium access control for wireless sensor networks," *Wireless Netw.*, vol. 12, no. 1, pp. 63–78, Feb. 2006.
[48] P. Wang, R. Dai, and I. Akyildiz, "A differential coding-based scheduling framework for wireless multimedia sensor networks," *IEEE Trans. Multimedia*, vol. 15, no. 3, pp. 684–697, Apr. 2013.
[49] P. Wang and W. Zhuang, "A collision-free MAC scheme for multimedia wireless mesh backbone," *IEEE Trans. Wireless Commun.*, vol. 8, no. 7, pp. 3577–3589, Jul. 2009.
[50] J.-H. Lee, "A massive transmission scheme in contention-based MAC for wireless multimedia sensor networks," *Wireless Personal Commun.*, vol. 71, no. 3, pp. 2079–2095, Aug. 2013.
[51] Y. Z. Zhao, C. Miao, M. Ma, J. B. Zhang, and C. Leung, "A survey and projection on medium access control protocols for wireless sensor networks," *ACM Comput. Surveys*, vol. 45, no. 1, pp. 7:1–7:37, Dec. 2012.
[52] J. I. Choi, M. Jain, K. Srinivasan, P. Levis, and S. Katti, "Achieving single channel, full duplex wireless communication," in *Proc. Int. Conf. Mobile Comput. Netw. (MobiCom)*, Chicago, IL, USA, Sep. 2010.
[53] S. Gollakota and D. Katabi, "Zigzag decoding: Combating hidden terminals in wireless networks," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 4, pp. 159–170, Oct. 2008.

[54] Y. Zhang, C. Zhu, and K.-H. Yap, "A joint source-channel video coding scheme based on distributed source coding," *IEEE Trans. Multimedia*, vol. 10, no. 8, pp. 1648–1656, Dec. 2008.
[55] D. Wang, V. S. Somayazulu, and J. R. Foerster, "Efficient cross-layer resource allocation for H.264/SVC video transmission over downlink of an LTE system," in *Proc. IEEE Int. Symp. World of Wireless, Mobile, Multimedia Netw. (WoWMoM)*, San Francisco, CA, USA, Jun. 2012.
[56] Z. Chen and D. Wu, "Rate-distortion optimized cross-layer rate control in wireless video communication," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 3, pp. 352–365, Mar. 2012.
[57] H. Zhang, Y. Zheng, M. A. A. Khojastepour, and S. Rangarajan, "Cross-layer optimization for streaming scalable video over fading wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 28, no. 3, pp. 344–353, Apr. 2010.
[58] E. Ekmekcioglu, H. K. Arachchi, C. G. Gurler, and S. S. Savas, "Content aware delivery of visual attention based scalable multi-view video over P2P," in *Proc. IEEE Int. Packet Video Workshop*, Munich, Germany, May 2012, pp. 71–76.
[59] S. T. Aditya and S. Katti, "FlexCast: Graceful wireless video streaming," in *Proc. ACM Int. Conf. Mobile Comput. Netw. (MobiCom)*, Las Vegas, NV, USA, Sep. 2011.
[60] Z. Guan, T. Melodia, and D. Yuan, "Jointly optimal rate control and relay selection for cooperative wireless video streaming," *IEEE/ACM Trans. Netw.* , vol. 21, no. 4, pp. 1173–1186, Aug. 2013.
[61] S. Pudlewski and T. Melodia, "Cooperating to stream compressively sampled videos," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Budapest, Hungary, Jun. 2013.
[62] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.
[63] S. Jakubczak and D. Katabi, "A cross-layer design for scalable mobile video," in *Proc. ACM Int. Conf. Mobile Comput. Netw. (MobiCom)*, Las Vegas, NV, USA, Sep. 2011.
[64] X. L. Liu, W. Hu, Q. Pu, F. Wu, and Y. Zhang, "ParCast: Soft video delivery in MIMO-OFDM WLANs," in *Proc. ACM Int. Conf. Mobile Comput. Netw. (MobiCom)*, Istanbul, Turkey, Aug. 2012.
[65] X. Fan, F. Wu, D. Zhao, and O. C. Au, "Distributed wireless visual communication with power distortion optimization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 6, pp. 1040–1053, Jun. 2013.
[66] J. N. Laneman, D. N. C. Tse, and G. W. Wornell, "Cooperative diversity in wireless networks: Efficient protocols and outage behavior," *IEEE Trans. Inf. Theory*, vol. 50, no. 12, pp. 3062–3080, Dec. 2004.
[67] S. A. Jafar and S. Shamai, "Degrees of freedom region of the MIMO X channel," *IEEE Trans. Inf. Theory*, vol. 54, no. 1, pp. 151–170, Jan. 2008.
[68] N. Mastronarde, F. Verde, D. Darsena, A. Scaglione, and M. van der Schaar, "Transmitting important bits and sailing high radio waves: A decentralized cross-layer approach to cooperative video transmission," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 9, pp. 1597–1604, Oct. 2012.
[69] E. Candes, M. Rudelson, T. Tao, and R. Vershynin, "Error correction via linear programming," in *Proc. IEEE Symp. Foundat. Comput. Sci. (FOCS)*, Pittsburgh, PA, USA, Oct. 2005.
[70] P. Merkle, J. B. Singla, K. Müller, and T. Wiegand, "Stereo video encoder optimization for mobile applications," in *Proc. 3DTV Conf.: The True Vis.—Capture, Transmiss., Display of 3D Video (3DTV-CON)*, Antalya, Turkey, May 2011.
[71] W. Pu, Z. Zou, and C. Chen, "Video adaptation proxy for wireless dynamic adaptive streaming over HTTP," in *Proc. Int. Packet Video Workshop (PV)*, Munich, Germany, May 2012.
[72] W. Cai and V. C. M. Leung, "Multiplayer cloud gaming system with cooperative video sharing," in *Proc. Int. Conf. Cloud Comput. Technol. Sci. (CloudCom)*, Taipei, Taiwan, Dec. 2012.
[73] A. Seema and M. Reisslein, "Towards efficient wireless video sensor networks: A HW/SW cross layer approach to enabling sensor node platforms," *IEEE COMSOC MMTC E-Lett.*, vol. 7, no. 4, pp. 6–9, Apr. 2012.
[74] C. Zhu and Y. Li, *Advanced Video Communications Over Wireless Networks*. Boca Raton, FL, USA: CRC, 2013.
[75] I. Sodagar, "The MPEG-DASH standard for multimedia streaming over the internet," *IEEE Multimedia*, vol. 18, no. 4, pp. 62–67, Apr. 2011.
[76] Y. Sun and T. Melodia, "The internet underwater: An IP-compatible protocol stack for commercial undersea modems," in *Proc. ACM Int. Conf. UnderWater Netw. Syst. (WUWNet)*, Kaohsiung, Taiwan, Nov. 2013.
[77] Z. Guan and T. Melodia, "Cloud-assisted smart-camera networks for energy-efficient 3D video streaming," *IEEE Comput.*, vol. 47, no. 5, pp. 60–66, May 2014.
[78] W. Zhu, C. Luo, J. Wang, and S. Li, "Multimedia cloud computing," *IEEE Signal Process. Mag.*, vol. 28, no. 3, pp. 59–69, May 2011.

Scott Pudlewski (M'07) received his B.S. in electrical engineering from the Rochester Institute of Technology, Rochester, NY, in 2008, and his M.S. and Ph.D. degrees in electrical engineering from the University at Buffalo, The State University of New York (SUNY), Buffalo, NY in 2010 and 2012, respectively. He is currently a Technical Staff Member at the Massachusetts Institute of Technology (MIT) Lincoln Laboratory in Lexington, MA. His main research interests include video transmission and communications, networking in contested tactical networks, convex optimization, and wireless networks in general.

Nan Cen received her B.S. and M.S. degrees from Shandong University, Jinan, China, in 2008 and 2011, respectively. From 2011 to 2012, she was a software developer with Alcatel-Lucent Co. in Qingdao, China. Currently, she is working towards the Ph.D. degree as a Research Assistant under the supervision of Dr. Tommaso Melodia in the Wireless Networks and Embedded Systems Laboratory, Department of Electrical Engineering, State University of New York at Buffalo. Her main research interests are in wireless multimedia sensor networks, compressed sensing based imaging, multiview video and optimizing.

Zhangyu Guan (M'11) is a postdoctoral research fellow in the Department of Electrical Engineering at the State University of New York at Buffalo. His current research interests are wireless network modeling and optimization, multimedia wireless networks, and mobile cloud computing. He received a Ph.D. in communication engineering from Shandong University, Jinan, China, in 2010. He is a member of IEEE, the IEEE Computer Society, and ACM.

Tommaso Melodia (M'07) received the "Laurea" and Doctorate degrees in telecommunications engineering from the University of Rome "La Sapienza," Rome, Italy, in 2001 and 2005, respectively, and the Ph.D. degree in electrical and computer engineering from the Georgia Institute of Technology, Atlanta, GA, USA, in 2007. He is an Associate Professor with the Department of Electrical Engineering, University at Buffalo, The State University of New York (SUNY), Buffalo, NY, USA, where he directs the Wireless Networks and Embedded Systems Laboratory. His current research interests are in modeling, optimization, and experimental evaluation of wireless networks, with applications to cognitive and cooperative networking, ultrasonic intrabody networks, multimedia sensor networks, and underwater networks. Prof. Melodia serves in the editorial boards of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, IEEE TRANSACTIONS ON MOBILE COMPUTING, IEEE TRANSACTIONS ON MULTIMEDIA, and *Computer Networks*. He received a National Science Foundation CAREER Award. He coauthored a paper that was recognized as the Fast Breaking Paper in the field of Computer Science by Thomson ISI Essential Science Indicators and a paper that received an Elsevier Top Cited Paper Award.