

# Enhancing Industrial Video Surveillance over Wireless Mesh Networks

Chaofan Yang, Chenshu Wu, Zheng Yang, Tongtong Liu, Zuwei Yin, Yunhao Liu, Xufei Mao  
School of Software and TNList, Tsinghua University  
{yangcf10, wucs32, hmilyyz, liutongtong7, yinzw81, yunhaoliu, xufei.mao}@gmail.com

**Abstract**—Industry 4.0 brings forward higher requirements on the monitoring of industrial production. Video surveillance based on Wireless Mesh Networks (WMNs) has demonstrated its effectiveness in a number of applications. Different from some typical applications of WMN that solve the “last mile” Internet access problem, WMN-based video surveillance for industrial monitoring is very likely to work in extreme circumstances or requiring high performance. Thus, the guarantee of video quality is the key for the success of industrial video surveillance. Through extensive experimental research, we find that the state of the art mapping and queuing algorithms are approaching complication and the room for improvement is decreasing. The possible solution for performance breakthrough lies in exploiting the potentials of data granularity for mapping. In this work, we propose IMesh, a video transmission solution based on WMN, which takes frame type, frame location, data packets and other factors into consideration and quantifies their impacts on video quality. The proposed approach is particularly suitable for video surveillance in industrial production under aggressive conditions. To the best of our knowledge, IMesh is the first one that differentiates, prioritizes, and schedules video data in the packet level, which is the finest granularity one can achieve while keep the MAC layer protocol unchanged. Experiment results show that the proposed solution outperforms previous works both in terms of video quality and packet delay.

## I. INTRODUCTION

The rising Industry 4.0 brings forward higher requirements on the monitoring of industrial production. Video surveillance (e.g., Closed Circuit Television, CCTV) becomes a prevalent solution due to its rich information and human friendly style. In some adverse conditions that are often for industrial production, however, it is very difficult or even impossible to deploy cabled systems for video transmission. Under such circumstances, wireless mesh networks (WMNs) [1], as an alternative, have demonstrated their effectiveness for video surveillance in a number of real-world applications [2], [3].

It is well known that the unreliable nature and shared media of multihop communications make the deployment of multimedia applications in wireless mesh networks a difficult task. Take our recent project deployment at an oilfield as an example. The oilfield located in Middle East area is about 14.7 square kilometers and consists of a base camp, an operating office, and over 200 well pads and rigs, which are scattered in an area of 400 square kilometers. The infrastructure of telecommunication and even road system are destroyed due to long term wars and conflicts. These difficulties prohibit personnel patrol and inspection of oil production sites due to cost and safety reasons. The video surveillance project of oilfield aims to deploy about 500 cameras and 121 of them have been installed on fixed locations, mobile vehicles, or handheld. High temperature and strong wind are two major factors for unpredictable network performance.

Different from some typical applications of WMN that solve the “last mile” Internet access problem [4], WMN-based

video surveillance for industrial monitoring is very likely to work in extreme circumstances yet requires high performance, such as broad monitoring area, large network scale, high reliability, low data transmission delay, unpredictable link quality, and even atrocious weather. Thus, the guarantee of video quality is the key for the success of industrial video surveillance [5], [6].

In the past two decades, video streaming over WLAN has drawn many efforts from researchers, resulting in a rich bundle of algorithms, protocols, and standards related to QoS management. For example, IEEE 802.11e defines EDCA and implements a set of priority queues to transmit data packets with various importance. Specifically, voice and video usually possess higher priority than other types of data. Compared with original 802.11, 802.11e partializes video data and accordingly gains better video quality. Meanwhile, the progresses in video coding have brought the so-called hierarchical coding technology that transforms video data into a stream of frames with different importance. As a result, the convergence of 802.11e and hierarchical coding has spawn a cross-layer design for video transmission: assigning more important frames into queues with higher priority, resulting in improved video quality in case of exhausted bandwidth [7], [8], [9], [10].

Some researchers have extended this idea to WMN, but these attempts do not achieve satisfying results. Especially in cases of increasing network scale and video streams, the bandwidth of WMN will be over stretched easily [11]. Other researchers propose to reduce transmitted data volume via in-network video analysis [12], [13]. However, computer vision technology for video content analysis often requires powerful computation capability, contradicting limited resources of WMN nodes. In addition, most of these efforts are validated through simulation, lacking applicability confirmation for real hardware platforms and systems [14], [15]. A reasonable trade-off between efficient network transmission and limited computation resource of WMN nodes remains an unsolved problem for real systems.

Through extensive experimental research, we find that the state of the art mapping and queuing algorithms are approaching complication and the room for improvement is decreasing. The possible solution for performance breakthrough lies in exploiting the potentials of data granularity for mapping. That is, instead of using frames as the basic data unit, mapping algorithms should drive towards finer granularity of video data.

The above statement comes from 3 basic observations in our experiments. (1) Frames of the same type are of different importance. Typical video coding approaches such as H.264 and MPEG-4 encode video data into 3 types of frames, namely I-frame, P-frame, and B-frame. We find that, by coding rules, all P-frames in a group (we will discuss group of pictures later) have decreasing impacts according to their appearing location in the group. This conclusion also holds for all B-

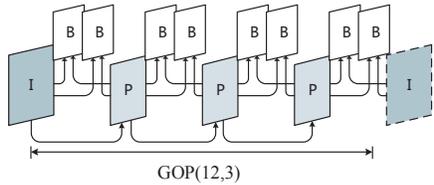


Fig. 1. Frame relations between  $GOP(12,3)$

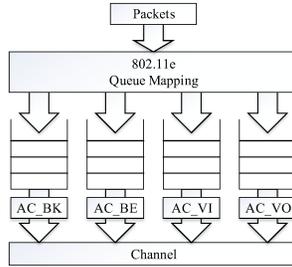


Fig. 2. Four ACs of EDCA

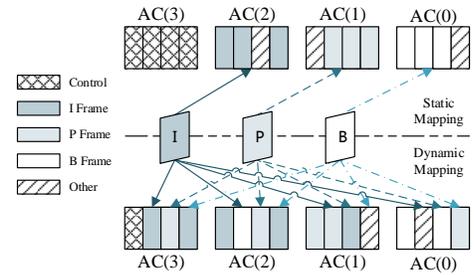


Fig. 3. Illustration of static and dynamic mapping

frames in both theoretical and experimental aspects. (2) Data packets conveying a single frame have different importance. A single video frame has to be divided into a number of data packets (e.g., datagram in network layer or other data unit in lower layers) for network transmission. Data packets of a single frame have various impacts on video quality; thus, it is beneficial to consider in packet level during mapping, instead of frame level. (3) Mapping should be flexible such that all data packets have chances to use high priority queues in order to take full advantages of bandwidth.

In this work, we propose IMesh, a video transmission solution based on WMN, which takes frame type, frame location, data packets and other factors into consideration and quantify their impacts on video quality. Our solution tries to find an equilibrium between efficient network transmission and limited computation resource on real WMN systems. The proposed approach is particularly suitable for video surveillance under aggressive conditions. Experiment results show that the proposed solution outperforms the default EDCA by  $>50\%$ , the static mapping schemes by 16%, and even the best dynamic mapping mechanism by 5% on average in terms of decoded video quality.

Our contributions are as follows. The first two observations point out the deficiency of using frame types or even a single frame as a metric for mapping. To the best of our knowledge, IMesh is the first one working in the packet level, which is the finest granularity one can achieve while keep the MAC layer protocol unchanged (i.e., under IEEE 802.11 framework). Besides, our solution is orthogonal to other QoS schemes including routing, scheduling, etc. [16], [17], [18], [19], [20], and can integrate with them readily. At last, we design and implement the hardware and communication protocol of WMN nodes, and evaluate IMesh's performance extensively through real experiments.

The rest of the paper is organized as follows. We review the preliminaries of this work about video hierarchical coding and 802.11e's QoS support. Section 3 presents the design and implementation details of IMesh. The prototype construction and experiments are discussed in Section 4. We investigate the state-of-the-art of network video streaming technologies in Section 5. At last, Section 6 concludes this work and discusses possible directions of future work.

## II. PRELIMINARIES

### A. Video Hierarchical Coding

In the era of big data, video data become one major type of data and video streaming has grown as the dominant traffic in wireless networks. Various video coding technologies have thus been developed to realize efficient and robust video streaming over wireless networks for plentiful multimedia applications and services. Among many alternatives, the in-

creasingly popular hierarchical coding technologies such as H.264 and MPEG-4 are widely adopted to provide wonderful video quality. Such techniques are advantaged in their excellent performance even over poor network conditions with acceptable coding complexity.

As a compressing technology, hierarchical coding generally involves three different types of video frames, i.e., I-frame, P-frame, and B-frame. Each frame type has different characteristics, as explained in the following.

1) I-frame (intra coded frame): the least compressible frame that contains all required information to decode itself and thus does not relies on any other video frames. An I-frame is typically compressed as a static picture and the size is relatively large. 2) P-frame (predictive coded frame): P-frames convey the difference compared to previous I-frames or P-frames, and thus require the information of preceding I- or P-frames to recover. 3) B-frame (bi-predictive coded frame): the most compressed frame that utilizes previous and forward frames (I- and P-frames) as reference data.

According to H.264 and MPEG-4 coding, a video clip is decomposed into a number of frames that are organized as groups, which is known as Group of Pictures (GOP). Each GOP always begins with an I-frame, which is followed by a certain number of consecutive P-frames and B-frames. The structure of GOP can be expressed by two parameters  $G(N, M)$ , where  $N$  indicates the total amount of frames included in the GOP (i.e., the distance between two consecutive I-frames in the whole video sequence) and  $M$  denotes the distance between the I-frame and the first P-frame (or two consecutive P-frames) within the same GOP. Taking Fig. 1 as a concrete example,  $G(12,3)$  denotes the GOP structure of 'IBBPBBPBBPBB'.

As shown in Fig. 1, different frames in a GOP are closely interlinked, although all GOPs of a video stream are independent from each other. Concretely, an I-frame is encoded independently and can be decompressed by itself. To decode a P-frame, however, information of preceding I-frame or P-frames is required. Furthermore, B-frames are encoded and decoded based on information of previous and forward I-frames or P-frames. In other words, if an I-frame fails to recover due to packet loss, all other frames in the same GOP cannot be correctly decoded and thus become useless even if they are all perfectly received. Similarly, if a P-frame is lost, all subsequent P-frames and B-frames become useless. No other frames will be implicated in case of the loss of a B-frame. Consequently, different frame types hold different importance factors, which can be drawn from the number of frames that are affected by a specific frame and the number of those affect it. Apparently, we can conclude that the importance of the three types of frames ranks as  $I > P > B$ .

As a result, for QoS consideration, more important frames

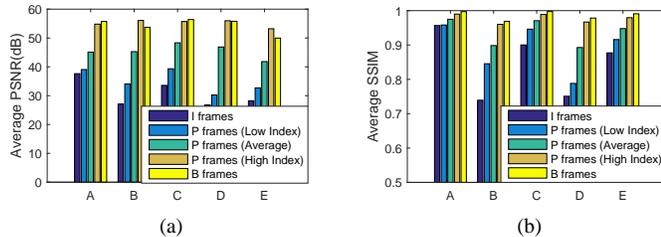


Fig. 4. Video quality when losing the same amount of I-frames, P-frames, B-frames, respectively

TABLE I. EDCA PARAMETERS OF EACH AC IN CASE OF 802.11A/N

AC	Priority	CWmin	CWmax	AIFS	TXOP
AC_BK	Low	15	1023	7	0
AC_BE	15	1023	3	0	
AC_VI	7	15	2	3.008ms	
AC_VO	High	3	7	2	1.504ms

should be first transmitted for better video quality, especially when the bandwidth is limited. Previous mapping mechanisms also mostly take advantages of this property of hierarchical coding [21], [10]. However, they merely distinguish the frame difference for delivery, yet do not dive into finer granularity, e.g., frame sequence and packet diversity, for QoS guarantee, thus leaving room for further improvement.

### B. 802.11e Enhanced Distributed Channel Access (EDCA)

H.264 video coding standard achieves efficient compression on application layer for video streaming. In practice, however, such techniques are not well elaborated because the resource management and scheduling strategies on lower layers are not optimized with awareness to the unique features of multimedia applications.

Original IEEE 802.11 standard employs distributed coordination function (DCF) as a basic access mechanism, which is built based on CSMA/CA and does not provide QoS guarantees. To support service differentiation, a hybrid coordination function (HCF) is introduced in 802.11e with two concurrent mechanisms, i.e., the HCF controlled channel access (HCCA) and the enhanced distributed channel access (EDCA) [22]. In this paper, we focus on EDCA, which works in a distributed manner and is promising to become the dominant channel access scheme in WLANs.

EDCA enables QoS support by introducing four access categories (AC). Different from original 802.11 that has only one queue on MAC layer, each AC in EDCA has its own transmission queue on MAC layer, which is assigned with different parameters to differentiate their priorities for channel access. As shown in Fig. 2, the four ACs, from low to high priority, are AC\_BK (background), AC\_BE (best effort), AC\_VI (video), AC\_VO (voice), or AC(0), AC(1), AC(2), AC(3), respectively. Service differentiation is realized by setting different channel access parameters of each AC, which include contention window bound (CWmin and CWmax), arbitration inter-packet space (AIFS), and transmit opportunity (TXOP). CW can be dynamically adjusted by itself while AIFS is a constant. TXOP indicates the period of time during which packets can be persistently sent. Table I illustrates the default values of these parameters, which are decided by physical layer. Generally, larger CW and AIFS both reduce channel collision probability yet increase transmission delay. Thus a smaller CW (CWmax and CWmin) or AIFS and a higher TXOP lead to higher priorities for channel access.

By default, QoS support is realized in EDCA by reserving

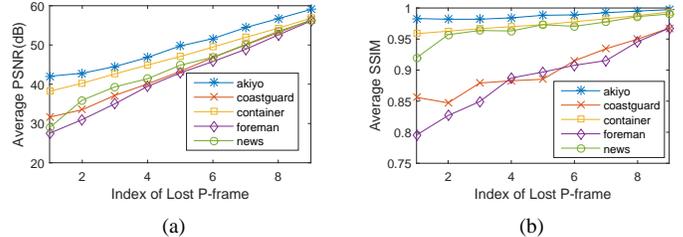


Fig. 5. Video quality when losing the same amount of P-frames at different positions

the prior AC(3) and AC(2) for real-time traffic and the others for best-effort and background traffic. The original EDCA mechanism neglects the video frame diversity and equivalently allocates all video frames into AC(2). Since video data occupy most of the traffics in a wireless video surveillance network, such mechanism could easily cause queue congestion in some ACs while the resources are not fully utilized, leading to unnecessarily packet losses and video quality degradation. Thus for video streaming over WMNs, the distinctive characteristics of real-time video streaming traffics on application layer should be taken into consideration to augment the QoS mechanism on MAC layer for better video delivery quality [9], [23]. Several previous mechanisms have driven towards this direction to improve the performance of EDCA by differentiating different frame types and considering dynamic traffic load [10], [7], [9], [8]. As shown in Fig. 3, previous static mapping mechanisms [9], [21] merely differentiate different frame types and allocate diverse frames into different priority queues by a set of fixed rules. Dynamic mapping schemes [10], [7] move a step further by considering dynamic network traffic load to mitigate queue congestion while improve network utilization.

In this paper, we also exploit all ACs for video streaming, yet from an outcome-based perspective to optimize the ultimate video quality and network resource utilization. We investigate to understand the relationships between video quality and various relevant factors including video frame types, diverse packets, traffic load queues, etc. Specifically, we dive into a finer granularity of video data and accordingly propose the first cross-layer design that works in packet-level, as detailed in the following section.

## III. THE PROPOSED MECHANISM

### A. Design Space

Generally, video traffic dominates the primary data flow of a wireless video surveillance system; while control and background traffics only occupy a negligible part. Thus, any efficient real-time streaming protocol for a wireless video surveillance network usually has the following characteristics and requirements:

- 1) High bandwidth utilization. We should make the utmost of the network resources to deliver as many effective packets as possible.
- 2) Low end-to-end delay. The transmission delay should be controlled to be as low as possible to guarantee the real-time capability.
- 3) Good video quality. Under certain network conditions (e.g., limited bandwidth), a video streaming protocol should first optimize the outcome video quality, rather than maximize the absolute size of data delivered.

Among the above three requirements, while the former two are common goals of general data communication networks, the last one acts as a unique feature of video streaming networks. Besides traditional performance criteria like packet

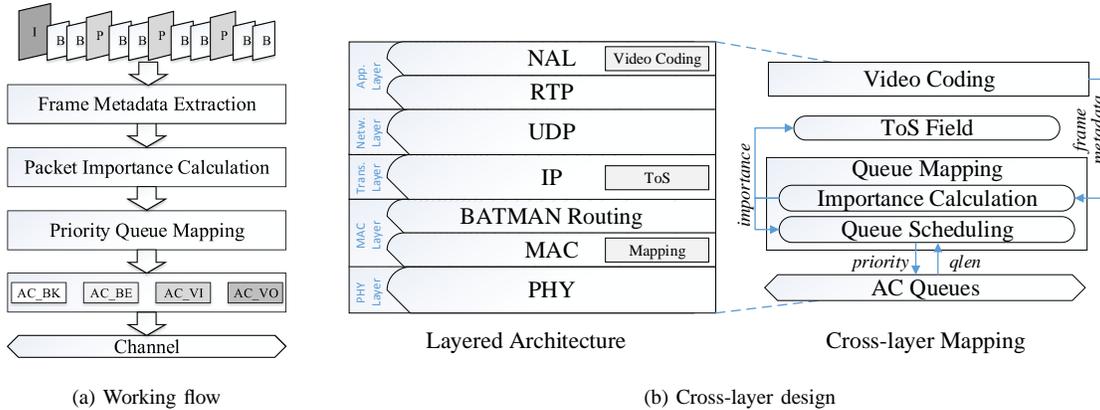


Fig. 6. Overview of the proposed mechanism

loss ratio, video quality should be better measured by specific outcome-based metrics, such as PSNR (Peak Signal-to-Noise Rate) and SSIM (Structural Similarity), etc. Video quality is the ultimate demand of a video surveillance system and thus should be the primary pursuit, especially for industrial video surveillance applications. In this sense, a real-time video streaming mechanism should not only improve the bandwidth utilization and reduce transmission delay, but more importantly, optimize the video quality under certain network conditions.

Inspired by video hierarchical coding used in popular video formats, we postulate video frames of different types contribute considerably differently to the outcome video quality, even with identical volumes. This is caused by the internal dependences between diverse frames in a GOP and the resulting diverse importance. Recall hierarchical coding mentioned in Section II, an I-frame is encoded and decoded independently. In contrast, a P-frame is only decodable by using the information from preceding I-frame and P-frames in the same GOP, while a B-frame can only be successfully encoded and decoded based on preceding and succeeding I-frames and P-frames. Informally, given an identical amount, I-frame packets make for the best video quality while B-frames derive the worst. Moreover, in the same GOP, P-frames and B-frames that come early turn out to be more important in terms of video quality.

As illustrated in Fig. 4 and Fig. 5, we conducted preliminary measurements on the impacts of different frame types and different frame indexes on the video quality. As seen in Fig. 4, losing an identical amount of I-frames results in significantly worse quality of the recovered video than P-frames and B-frames, measured by either PSNR (Fig. 4a) or SSIM (Fig. 4b). So does P-frames to B-frames. It is further noticed that P-frames appearing at different positions of a GOP have different importance for recovering the video, which is better validated in Fig. 5. Apparently, the earlier a P-frame appears, the more important it is. The result potentially validates our observation that not only frame types but also frame positions sway the different importance of a frame for decoding a video.

Previous mapping mechanisms, however, merely distinguish different frame types but do not differentiate frames that belong to the same type yet appear at different positions [23], [21]. In this paper, the proposed IMesh takes advantages of diverse impacts of video frames with different *types* and different *positions* on the ultimate video quality.

Furthermore, we dive into the packet structure of a video frame to understand the relationship between received video

packets and recovered video quality. A video frame is usually segmented into a number of packets for transmission. Apart from the video frame differences, it is observed that data packets of a specific frame act differently for recovering a video frame. In detail, the header packets contain the structure information (e.g., length) of a frame and appear to be more critical for decoding the whole frame. More rigorously speaking, all packets of a frame lose significance if the corresponding header packet is lost. Inspired by this observation, we propose a *packet-level* mapping mechanism that accounts for packet diversity. Such packet-level data granularity, to the best of our knowledge, is beyond the consideration of existing protocols.

### B. Adaptive Mapping Mechanism

Recalled Section II, traffics on one AC with smaller AIFS or CWmax or CWmin have a greater chance to access the wireless channel. To achieve low transmission delay, we are inclined to feed video traffics preferentially to higher priority ACs. Yet to fully utilize network resources and avoid potential queue congestion, we exploit all four ACs for transmission. Particularly, when the prior queues are congested, video traffics are also scheduled to be transmitted on lower priority ACs. Most importantly, all video packets are scheduled based on their importance with respect to the outcome video quality, which are derived from their frame types and positions (in one GOP) and the specific packet properties.

1) *Embracing Frame-Level and Packet-Level Diversity:* Given a video unit  $\text{GOP}(N, M)$ , the importance of each packet, denoted as  $w$ , is generated as follows.

Basically, all packets in the I-frame hold the highest importance value  $w = 1$ . For a packet in a P-frame or a B-frame, its importance value depends on the preceding and succeeding frames that will affect the current frame and that it will affect. Concretely, denote the amount of frames that are effected by the current frame as  $f_0$ , and the amount of frames that affect the current frame's decoding as  $f_1$ , we define the importance value of an ordinary packet as

$$w = g(f_0 \alpha^{f_1}),$$

where  $\alpha \in (0, 1)$  is a weighting constant that balances the impacts of  $f_0$  and  $f_1$ ,  $f_0 \geq 1$  (all P-frames and B-frames are at least affected by the I-frame),  $f_1 \geq 1$  (a frame will at least affect itself),  $g(x)$  is a monotonously increasing function. That is, the more frames depend on the current frame and less frames the current frame relies on, more important the current frame turns out. More intuitively, a frame appearing earlier in a GOP turns out to be more important than a later one.

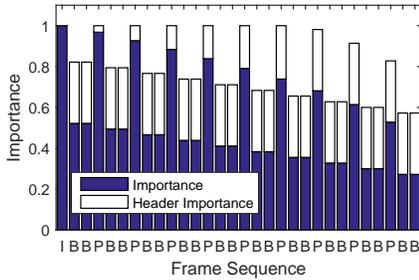


Fig. 7. An example of importances of all frames within a GOP(30, 3) ( $\alpha = 0.8, b_0 = 0.3, h = 0.3$ )

To reduce the computation overhead on a mesh node,  $g(x)$  is defined as  $g(x) = a(\log(x) + b) + b_0$ . Consequently, the importance value of a packet appearing in a P-frame or a B-frame is derived as

$$w = a(\log f_0 + f_1 \log \alpha + b) + b_0.$$

Here  $b_0$  is a constant baseline factor that increases the importance values to endow packets from B-frames a chance to get into a high priority queue.  $a$  and  $b$  and  $b_0$  are introduced to accommodate the first term  $w$  to the range of  $[b_0, 1]$ . Here  $a$  and  $b$  are given the values as follows.

$$a = \frac{1 - b_0}{\log(N) - \log(\alpha^{N/M})}, b = -\log(\alpha^{N/M}),$$

where  $\log(N)$  and  $\log(\alpha^{N/M})$  are the maximum and minimum value of  $\log(f_0 \alpha^{f_1})$ , respectively.

For the  $p$ th ( $p > 0$ ) P-frame in a GOP,  $f_0 = N + M - 1 - M * p, f_1 = p$ . For any subsequent B-frame of  $p$ th P-frame, we have  $f_0 = 1, f_1 = \min\{p + 2, N/M\}$ . Note that for a B-frame following the I-frame,  $f_1 = 2$ , i.e., it only relies on the preceding I-frame and the followed P-frame. Equivalently,  $p$  can be treated as 0 for any B-frames that are sandwiched between an I-frame and a P-frame.

So far, we have defined the calculation of an ordinary packet's importance value, which well adapts to the *frame-level* diversity of video traffics. To emphasize the importance of a frame-header packet, which is known to be crucial to decode a frame, we add an additional term  $h$  to upraise  $w$  of a header packet to beyond other ordinary ones. If  $w + h > 1$ , we set  $w = 1$ . By doing this, we achieve a *packet-level* adaptation scheme that differentiates individual packets to appropriate ACs, even when they originate from an identical frame type. Fig. 7 illustrates an example of importance diversity of frames within a GOP using IMesh.

2) *Priority Queue Scheduling*: Given the importance indicator of each packet, we can accordingly schedule it into an appropriate AC by considering the congestion window and the dynamic queue lengths. The key principle is to make the most of high priority ACs to reduce transmission delay and packet loss ratio. Particularly, when the traffic is lighter, even packets with least importance gain large opportunities to be assigned to a high priority queue to achieve low delay. In contrast, less important packets acquire a higher probability to fall into a lower priority queue in case of crowded traffics, in purpose of vacating prior queues for more important packets to guarantee the video quality.

To feed a packet into a specific AC, we sequentially check each AC from high priority to low priority. Once an AC is found to have sufficient buffer, we assign the packet into that AC and stop checking. Particularly, given a packet's

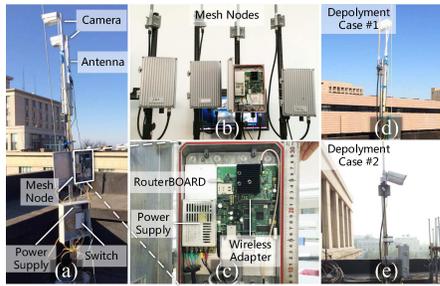


Fig. 8. Experimental platforms and deployment scenarios

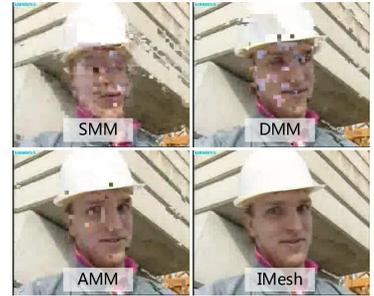


Fig. 9. Visual comparison of received video

importance value  $w$ , we put it into  $AC(i)$  if  $w * threshold(i) > qlen(i)$ ; otherwise, we continue to check the next AC with a lower priority.  $threshold(i)$  and  $qlen(i)$  are the preset maximum and the current queue length of  $AC(i)$ , respectively. If a packet is not eligible for any one of  $AC(3)$ ,  $AC(2)$  and  $AC(1)$ , then it is injected into  $AC(0)$  directly with no need of validating the above conditions. The mapping algorithm is designed as simple as possible to reduce computation complexity and thus moderate energy consumption on mesh nodes, because in industrial video systems many nodes are likely to be deployed in the wild and supplied by battery or solar energy.

The above queue mapping strategy gives important packets larger chances to be sent via a high priority AC, and meanwhile prevent the less important packets from congesting the queue (thus always reserve the probability for more important packets to compete for superior ACs even they are crowded).

3) *Practical Realization*: The processing flow of IMesh is presented in Fig. 6a and the cross-layer design is demonstrated in Fig. 6b, where we mainly modify the MAC layer while consider the real-time requirements from application layer. In practice, the importance value of each packet is calculated at the source node that initiates the video streaming, according to the inherent coding structure. When a mesh node receives a request to transmit a video, it first breakdowns a GOP into frames and then segments each frame into data packets appropriately. The importance value of each packet is calculated at this stage, according to its frame type, frame position, and packet property. The importance value is then carried by the data packet itself as an internal field and keeps unchanged when propagating to other nodes in the networks (since the importance value is actually independent from concrete network conditions). In this paper, we leverage the rarely used Type-of-Service (ToS) field on IP layer for recording the importance value.

The queue mapping needs to be conducted on each intermediate forwarding node, accounting for its specific channel conditions. Specifically, the node first parses the packet and derives the importance value from the ToS field. Then the packet is sent into an appropriate AC accounting for the concrete queue lengths on that node. Applying such mapping scheme, some packets will potentially arrive out-of-order. To mitigate such effects, the play-out buffer of the destination node should be slightly increased in practice. Fig. 6a shows the overall working flow of mapping a frame to a specific AC.

## IV. EXPERIMENTS AND EVALUATION

### A. Experiment Methodology

We implemented IMesh on real video surveillance system over WMN. As shown in Fig. 8, the core part of our WMN

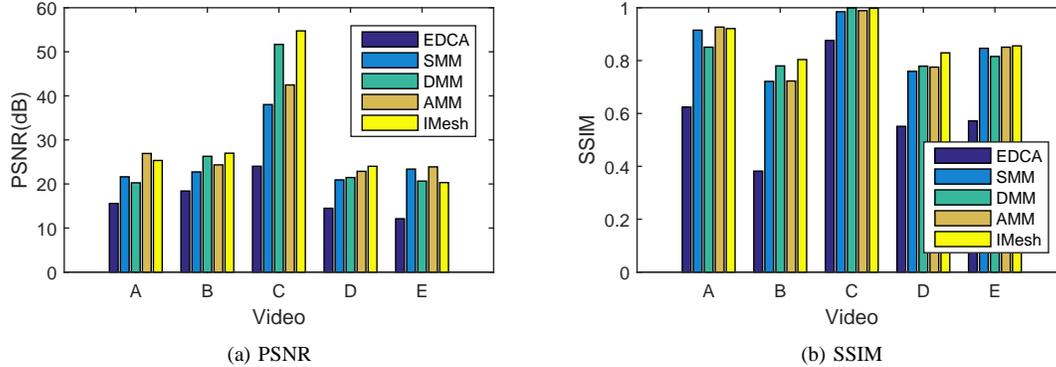


Fig. 10. Overall performance of IMesh

nodes consists of MikroTik RouterBOARD 411U with Atheros AR7130 300MHz network processor and 32M memory. We use a Broadcom antenna at 5GHz frequency band with a gain of 12dBi. The nominal communication range of a mesh node is about 3~5 km, which is suitable for industrial applications even in the wild. In addition, we use an Atheros AR9220 wireless adapter, which supports EDCA queuing mechanism on hardware level. Each EDCA class is mapped to a hardware queue. The deployable mesh node is encapsulated in a waterproof tin box for wild adoption.

The operating system is OpenWRT, which is built upon Linux 2.4.30 kernel and designed for embedded devices such as wireless AP. We do not use the built-in mesh protocol provided by OpenWRT. Instead, B.A.T.M.A.N. advanced (referenced as batman-adv below) is adopted for mesh routing due to its performance superiority.

The *mac80211* module in OpenWRT controls how packets are sent and received. Particularly, *queue selection* is a sub-procedure of transmitting packets procedure. We modified the queue selection method to introduce IMesh. Our method first reads headers of the packet to be sent. If it is a packet of an H.264 stream, necessary information such as frame type, GOP, and frame id are stored. The importance of the packet is accordingly calculated. The packet is finally given a queue id which depends on packet importance and queue loads. We store the importance value into the ToS field of IP header, since the field is seldom used and won't be modified when transmitting via the mesh network.

As depicted in Fig. 8, we deployed a mesh network at our university campus for daily security monitoring, which has run for more than half a year. Every mesh node within the network is attached with one or two cameras. Some node pairs share a direct Line-Of-Sight link, while the others are obstructed by buildings and walls. We argue that, although deployed in campus, our mesh nodes are all produced and packaged on an industrial standard, which are water-proof and dust-proof and applicable in aggressive conditions.

We implemented IMesh upon the mesh network and collect real traces for evaluation. Two PCs (namely A and B) are connected to two nodes (a source node attached with cameras and a sink node that finally receives and decodes the video) respectively with Ethernet cables. The two nodes are three hops away from each other in the wireless network. During the experiment, we transmit test video sequences from PC A to PC B and record transmission logs on both transmitter and receiver. The records on transmitter is used as ground truth. All result analyses are based on the pair of records. Since

the channel conditions are vulnerable to the environmental dynamics (especially in campus environments where the 5GHz band is extremely crowded), the link quality of mesh network will be unstable. To reduce the uncertainties caused by such variants, all the experiments are repeated at least three times and the results are integrated for performance evaluation.

Five publicly available and commonly used video sequences [23] are chosen to test the proposed method. They are all in YUV CIF (352\*288) format, and their frame counts are 300 (“coastguard”, “container”, “foreman”) and 2000 (“bridge-close” and “highway”). The videos are converted to H.264 format with a rate of 25 frames per second and GOF(12,3) before streaming. Considering mesh transmission delay and human tolerance, the play-out buffer is set to 1s to mitigate the potential out-of-order packets. Any packet whose delay is longer than 1s will be dropped and unrecorded.

Three video quality metrics, PSNR, SSIM, and DFR (Decoded Frame Rate) are employed to evaluate the performance of a specific mapping mechanism. PSNR describes pixel-level divergence of two pictures. As for video, it is the average PSNR of each corresponding frames in two videos. SSIM describes the structural divergence, which, compared with PSNR, reflects human feeling of video quality. DFR is the ratio of successfully decoded frames, which describes the video quality on frame-level.

We also implemented four previous mechanisms as baselines. Specifically, we compare with (1) SMM [9] accounting for frame types, (2) DMM [10], [11] accounting for frame types and traffic load, (3) AMM [23] considering frame types, traffic load and GOP structure, and (4) the default EDCA mechanisms without advanced mapping strategy that does not differentiate video traffics.

## B. Performance

1) *Overall Performance*: We first evaluate the overall performance of IMesh using a set of parameters as  $\alpha = 0.6$ ,  $b_0 = 0.2$ ,  $h = 0.6$ , and  $threshold = [\infty, 80, 50, 50]$  for AC(0) to AC(3) respectively. Fig. 9 illustrates a visual comparison of the mapping mechanisms. The overall quantitative results on different video clips are shown in Fig. 10, where video A, B, C, D, E respectively represent the video “bridge”, “coastguard”, “container”, “foreman”, “highway”. As seen, IMesh achieves the best quality for video B, C, and D in terms of PSNR and for video B, C, D, and E in terms of SSIM. For video A, IMesh performs comparable with other mechanisms, i.e., there's only 1.5dB difference in PSNR and 0.005 in SSIM difference. As comparison, previous methods including SMM

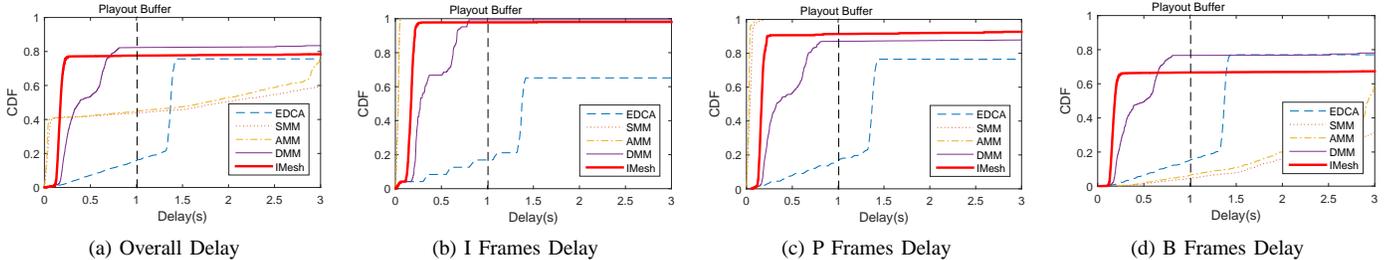


Fig. 11. Delay of Packets

and AMM merely realize good performance on parts of the videos (e.g., A and E) yet degrade on others (e.g., B and C). In average, IMesh improves delivered video quality by 5% than AMM, 9% than DMM, and 16% than SMM, respectively. In addition, IMesh outperforms the default EDCA mechanism in all cases, achieving an average improvement of 74% in terms of PSNR and 54% in terms of SSIM respectively. Thus in a nutshell, the superiority of IMesh lies in its adaptivity to different types of videos and thus is particularly applicable to various industrial scenarios.

We further inspect the delay distribution of video packets, which is shown in Fig. 11 as CDF graphs. Fig. 11a shows the overall delay of all packets while the rest depict packet delays of disaggregated I-frames, P-frames and B frames, respectively. Since EDCA doesn't distinguish frame types, its packet delays are the same for all frames. AMM and SMM yields very low packet delay for I-frames and P-frames, but very high delay on B-frames (or even leading to packet losses). The lowest overall delay and lowest packet loss rate are both achieved by DMM, but due to the high P-frame loss rate, the resulted video quality turns out to be worse than others. IMesh balances loss rate of B frame and P frame well and thus achieves better performance.

*2) Impacts of Individual Factors:* We examine the impact of individual factors, mainly  $\alpha$ ,  $b_0$ , and  $h$ , on the ultimate video quality by tracing the delivery performance under different values of them. The evaluation is conducted using three different typical videos, i.e., coastguard, container and foreman.

**Impact of  $\alpha$ .** In principle,  $\alpha$  balances the impacts of affecting frame amount  $f_0$  and affected frame amount  $f_1$ , i.e., smaller  $\alpha$  emphasizes impacts of  $f_1$  and vice versa. Fig. 12a illustrates the importance values with  $\alpha$  changing from 0.1 to 1.0. As seen, with the increasing of  $\alpha$ , the importance values of P-frames increase while those of B-frames decrease. In addition, if  $\alpha$  is too small, preceding B-frames turn out to be more important than some of later P-frames. Otherwise if  $\alpha$  is too large, the importance of all B-frames will be too light to get any of them into a high priority queue. This is confirmed by Fig. 12c that the DFR remarkably decreases with  $\alpha$  larger than 0.7, which may cause potential congestion and thus packet losses of B-frames in lower priority queues. Consequently, as shown in Fig. 12b, an appropriate  $\alpha$  with neutral values, e.g., from 0.4 to 0.6, yields the best quality of decoded video (we only present the results using SSIM while omit the similar results using PSNR due to space limitation).

**Impact of  $b_0$ .**  $b_0$  is designated to allow some preceding B-frames to acquire opportunities to get in a high priority queue. Fig. 13a illustrates the varying trend of the importance of a sequence of frames in a GOP with respect to  $b_0$  of different values. Note that a packet of larger importance acquires greater chance to be sent via a high priority queue, yet a packet assigned with a higher priority queue does not necessarily

has an absolutely larger importance value than that in a lower priority queue because the mapping further depends on the queue lengths. As demonstrated in Fig. 13c, with the increasing of  $b_0$ , more B-frames will get into prior queues and thus the overall DFR increases. The overall video quality, however, does not necessarily improve since these B-frames contribute less to PSNR yet might take over the channel and squeeze out some posterior P-frames, as shown in Fig. 13b. Synthesizing all results in Fig. 13, an appropriate range of  $b_0$  in practice falls in the interval of 0.2 to 0.4, which improves SSIM and simultaneously reserves a considerable DFR.

**Impact of  $h$ .** As shown in Fig. 14c, DFR obviously increases with a larger  $h$ , which is postulated to guarantee the delivery of more frame header packets. The trends potentially validate the essentiality of frame header for decoding a frame. Although more frames can be (partially) decoded as the header is successfully received, the integrated PSNR or SSIM does not accordantly rise. The reason lies in that frame header only ensures that one frame is recoverable (partially or completely) but does not indicates the successful delivery of other packets within the frame. The PSNR or SSIM, however, is related to the whole frame and thus do not necessarily grow with more headers. Nevertheless, more decoded frames (even partially) render smoother video experience. During the practical implementation of IMesh,  $h$  is assigned with a value in the range of [0.6, 0.8], which sufficiently results in a considerably high header delivery rate.

**Impact of different video content.** The testing videos differ in size, duration, and most importantly, content richness. For example, the "container" video is a clip of a security monitoring camera that captures a slowly changing view of a bridge. As comparison, "coastguard" tracks a motor boat sailing on the sea and the scene change fast. Consequently, the real-time generated video traffic of "coastguard" is much heavier than "container". As seen from Fig. 12, Fig. 13, and Fig. 14, video "container" can be delivered almost perfectly in any case, regardless of various values of the above parameters. In other words, IMesh only improves video quality in case of heavy network traffic and bad link conditions, since the design goal of IMesh (as well as previous mapping mechanisms) is to mitigate the effects of packet losses on video quality, i.e., improving and guaranteeing the video quality in case of equivalent packet losses. Hence for industrial video surveillance that usually works in extreme environments, IMesh is expected to achieve even remarkable improvement of video quality.

## V. RELATED WORKS

Numerous research efforts have been devoted in routing, scheduling, mapping etc to achieve efficient video streaming for real-time applications over wireless networks (either WLANs or wireless ad-hoc networks) [6], [4], [11], [18]. Specifically, this paper is closely related to QoS guarantee for video streaming applications in WMNs.



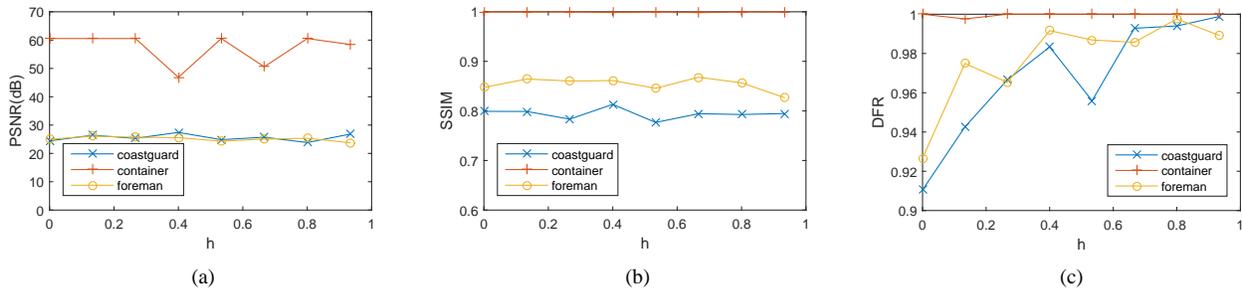


Fig. 14. Impact of  $h$  on video delivery quality measured by (a) PSNR, (b) SSIM, (c) DFR

queues to explore any potentials for improvement. In addition, more problems will arise along with the progress of our video surveillance project at oilfield. We will face these practical problems and report our deployment experience in the future.

#### ACKNOWLEDGMENTS

This work is supported in part by NSFC under grant 61522110, 61332004, 61472098, 61303209, 61572366, and Beijing Nova Program under grant Z151100000315090.

#### REFERENCES

- [1] M. L. Sichitiu, "Wireless mesh networks: opportunities and challenges," in *Proceedings of World Wireless Congress*, vol. 2, 2005.
- [2] A. Networks, "Using Wireless Mesh Networks for Video Surveillance," <http://www.arubanetworks.com/>, 2015.
- [3] Strixsystems, "Strix High Performance Wireless Mesh For Video Surveillance," <http://www.strixsystems.com/cswifimeshforipvideosurveillance.aspx>, 2015.
- [4] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris, "Link-level measurements from an 802.11 b mesh network," *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 4, pp. 121–132, 2004.
- [5] N. Cranley and M. Davis, "Performance evaluation of video streaming with background traffic over ieee 802.11 wlan networks," in *ACM workshop on Wireless multimedia networking and performance modeling*, 2005.
- [6] T. Zhang, A. Chowdhery, P. Bahl, K. Jamieson, and S. Banerjee, "The design and implementation of a wireless video surveillance system," in *ACM MobiCom*, 2015.
- [7] Z. Wan, N. Xiong, N. Ghani, M. Peng, A. V. Vasilakos, and L. Zhou, "Adaptive scheduling for wireless video transmission in high-speed networks," in *IEEE Conference on Computer Communications Workshops*, 2011.
- [8] I. Ali, M. Fleury, S. Moiron, and M. Ghanbari, "Enhanced prioritization for video streaming over wireless home networks with ieee 802.11 e," in *IEEE International Symposium on Broadband Multimedia Systems and Broadcasting*, 2011, pp. 1–6.
- [9] A. Ksentini, M. Naimi, and A. Guéroui, "Toward an improvement of h. 264 video transmission over ieee 802.11 e through a cross-layer architecture," *Communications Magazine, IEEE*, vol. 44, no. 1, pp. 107–114, 2006.
- [10] C.-H. Lin, C.-K. Shieh, C.-H. Ke, N. K. Chilamkurti, and S. Zeadally, "An adaptive cross-layer mapping algorithm for mpeg-4 video transmission over ieee 802.11 e wlan," *Telecommunication Systems*, vol. 42, no. 3-4, pp. 223–234, 2009.
- [11] N. Chilamkurti, S. Zeadally, R. Soni, and G. Giambene, "Wireless multimedia delivery over 802.11 e with cross-layer optimization techniques," *Multimedia Tools and Applications*, vol. 47, no. 1, pp. 189–205, 2010.
- [12] I. Haritaoglu, D. Harwood, and L. S. Davis, "W 4: Real-time surveillance of people and their activities," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 8, pp. 809–830, 2000.
- [13] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, "Pfinder: Real-time tracking of the human body," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 19, no. 7, pp. 780–785, 1997.
- [14] S. Khan, Y. Peng, E. Steinbach, M. Sgroi, and W. Kellerer, "Application-driven cross-layer optimization for video streaming over wireless networks," *Communications Magazine, IEEE*, vol. 44, no. 1, pp. 122–130, 2006.
- [15] Y. Andreopoulos, N. Mastrorarde, and M. Van der Schaar, "Cross-layer optimized video streaming over wireless multihop mesh networks," *Selected Areas in Communications, IEEE Journal on*, vol. 24, no. 11, pp. 2104–2115, 2006.
- [16] D. S. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," *Wireless Networks*, vol. 11, no. 4, pp. 419–434, 2005.
- [17] M. Afanasyev and A. C. Snoeren, "The importance of being overheard: throughput gains in wireless mesh networks," in *ACM SIGCOMM IMC*, 2009.
- [18] V. Gabale, B. Raman, P. Dutta, and S. Kalyanraman, "A classification framework for scheduling algorithms in wireless mesh networks," *Communications Surveys & Tutorials, IEEE*, vol. 15, no. 1, pp. 199–222, 2013.
- [19] W. Wei and A. Zakhor, "Interference aware multipath selection for video streaming in wireless ad hoc networks," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 19, no. 2, pp. 165–178, 2009.
- [20] N. Mastrorarde, Y. Andreopoulos, M. van der Schaar, D. Krishnaswamy, and J. Vicente, "Cross-layer video streaming over 802.11 e-enabled wireless mesh networks," in *IEEE ICASSP*, 2006.
- [21] R. MacKenzie, D. Hands, and T. O. Farrell, "Qos of video delivered over 802.11 e w lans," in *IEEE ICC*, 2009, pp. 1–5.
- [22] S. Mangold, S. Choi, P. May, O. Klein, G. Hiertz, and L. Stübör, "Ieee 802.11 e wireless lan for quality of service," in *European Wireless*, 2002.
- [23] X.-W. Yao, W.-L. Wang, S.-H. Yang, Y.-F. Cen, X.-M. Yao, and T.-Q. Pan, "Ipb-frame adaptive mapping mechanism for video transmission over ieee 802.11 e w lans," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 2, pp. 5–12, 2014.
- [24] Y. Ding, Y. Yang, and L. Xiao, "Multisource video on-demand streaming in wireless mesh networks," *IEEE/ACM Transactions on Networking*, vol. 20, no. 6, pp. 1800–1813, Dec. 2012.
- [25] S. Kompella, S. Mao, Y. T. Hou, and H. D. Sherali, "Cross-layer optimized multipath routing for video communications in wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 4, pp. 831–840, May 2007.
- [26] K.-W. Lim, Y. Seo, W.-S. Jung, Y.-B. Ko, and S. Park, "Design and implementation of adaptive wlan mesh networks for video surveillance," *Wireless Networks*, vol. 19, no. 7, pp. 1511–1524, 2013.
- [27] Y. Ding and L. Xiao, "Video on-demand streaming in cognitive wireless mesh networks," *IEEE Transactions on Mobile Computing*, vol. 12, no. 3, pp. 412–423, 2013.
- [28] Y. Ding, Y. Yang, and L. Xiao, "Multi-path routing and rate allocation for multi-source video on-demand streaming in wireless mesh networks," in *IEEE INFOCOM*, 2011, pp. 2051–2059.
- [29] L. Zhou, X. Wang, W. Tu, G. m. Muntean, and B. Geller, "Distributed scheduling scheme for video streaming over multi-channel multi-radio multi-hop wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 3, pp. 409–419, April 2010.
- [30] "Providing mac qos for multimedia traffic in 802.11e based multi-hop ad hoc wireless networks," *Computer Networks*, vol. 51, no. 1, pp. 153–176, 2007.
- [31] W.-P. Lai and B. Li, "A piecewise packet mapping algorithm for video transmission over 802.11 e wireless networks," in *IEEE International Conference on Intelligent Networking and Collaborative Systems*, 2011.