Experimental Analysis of Bandwidth Allocation in Automated Video Surveillance Systems

Sina G. Davani and Nabil J. Sarhan
Wayne State University
5050 Anthony Wayne Dr
Detroit, Michigan 48202
(sina.gholamnejad.davani,nabil.sarhan)@wayne.edu

ABSTRACT

We consider the bandwidth allocation problem in automated video surveillance systems, in which a monitoring station analyzes the video streams captured and delivered wirelessly by multiple cameras. In contrast with prior studies, we provide a detailed experimental analysis of cross-layer optimization by developing a real system and conducting extensive experiments. In addition, we present an enhanced cross-layer optimization solution that allocates bandwidth to different cameras in a manner that optimizes the overall detection accuracy. The solution works with the popular HTTP streaming approach and includes a new online scheme for estimating the effective airtime of the network. The results show that the proposed solution significantly improves the detection accuracy.

KEYWORDS

Bandwidth allocation, cross-layer optimization, effective airtime estimation, video streaming, wireless networks, WLAN.

1 INTRODUCTION

Automated Video Surveillance (AVS) enables the realtime detection of threats by running computer vision algorithms as opposed to human observation. The research on AVS has focused primarily on developing robust computer vision algorithms for the detection, tracking, and classification of objects and the detection and classification of unusual events [3, 4] (and references within). By contrast, much less work has considered the design of AVS systems.

In the considered AVS system, illustrated in Figure 1, multiple video cameras capture and deliver live video streams to a central monitoring station over a single-hop IEEE 802.11e wireless LAN. The monitoring station is connected to the access point with a high-bandwidth wired link that is not deemed as a bottleneck. The monitoring station runs computer vision algorithms to generate automated alerts whenever suspicious events, subjects, or threats are detected in the monitored site. Multiple such systems or cells can be used to construct a larger system.

This paper addresses the main challenge in the considered system, that is the wireless network has limited available bandwidth,

MM'17, October 23–27, 2017, Mountain View, CA, USA.
 ISBN 978-1-4503-4906-2/17/10…$15.00
DOI: https://doi.org/10.1145/3123266.3123376

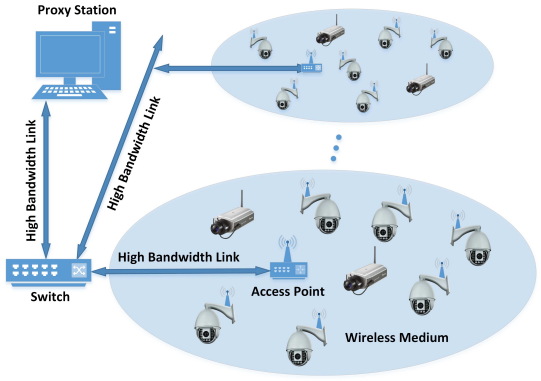


Figure 1: Considered Automated Video Surveillance System

which should be estimated accurately and then distributed efficiently among various cameras. Bandwidth allocation in general *many-to-one* video streaming systems has been addressed by only few studies [2, 6, 7] by using cross-layer optimization. In addition to being simulation-based, they sought to optimize video distortion. Distortion can be a good metric for certain systems, but in AVS, the detection accuracy is the most important metric. Study [1] considered accuracy-based optimization but was simulation-based as well. None of the prior studies considered HTTP streaming, which is the most popular method used today, as opposed to Real-time Transport Protocol (RTP) streaming. In addition, none of these studies used H.264 encoding. In particular, MJPEG was used in [1, 2], whereas abstract video data was used in [6].

In contrast with prior work, we build a real video surveillance system and run actual experiments. In addition, we employ HTTP streaming and use homogeneous cameras as well as heterogeneous ones with varying capabilities (frame rates, resolutions, etc.) and limitations to mimic different realistic deployment scenarios. We primarily use H.264, but we consider the co-existence of different encoders. We use *FFmpeg*, *SDL*, *Snappy*, and *Curl* libraries in the development of various system units. The system includes a customized video player to provide full control over the video decoding process and provide the required statistics by the optimization solution. One of the main challenges in building the system was working with poorly documented commercial products with different standards and interfaces.

Moreover, we propose an enhanced cross-layer optimization solution for managing the network bandwidth in a manner that optimizes the overall detection accuracy. The solution considers the

limitations of various cameras and manages the application rates and transmission opportunities of various video sources based on the dynamic network conditions. Furthermore, the solution includes a new scheme for estimating the effective airtime of the network based on a new novel method for estimating data dropping in the medium, employing smoothing calculations of the stream bitrates received by the monitoring station. The results show that the proposed solution significantly improves the detection accuracy.

The rest of this paper is organized as follows. Section 2 provides background information and discusses the related work. Subsequently, Section 3 presents the proposed cross-layer optimization solution and Section 4 presents the developed system. section 5 discusses the performance evaluation methodology. Finally, Section 6 presents and analyzes the main results.

2 BACKGROUND INFORMATION AND RELATED WORK

The 802.11e standard [8] enhances the support of multimedia applications by enabling the provision of different quality-of-service (QoS) levels among different access categories (AC) within the same station. In the preferred *Enhanced Distributed Channel Access* (EDCA) mode priorities are implemented using four parameters, including the *Transmission Opportunity Time* (TXOP) [6, 10].

Cross-layer optimization in video streaming over wireless networks has recently received much attention. A system in which only one station streams a video at a time was considered in [5, 9] (and references within), whereas a system in which a central video server streams to multiple stations was considered in [11, 12] (and references within), and a system in which multiple stations deliver video streams to a central station was considered in [1, 2, 6, 7].

Studies [2, 6] formulated and solved an optimization problem that minimizes the sum of distortion in all video streams, instead of accuracy error which is the main concern in video surveillance systems. Study [1] used the accuracy error as the main factor in the optimization problem, but the formulation does not include the limitation of the cameras in sending the encoded video streams. All the mentioned studies were simulation-based, used RTP streaming, and assumed MJPEG or abstract data streams.

The effective medium airtime can be defined as the fraction of the network time that is used for delivering useful data. As will be discussed later, solving the optimization problem requires an accurate estimation of the effective airtime. Study [2] developed an online effective airtime estimation algorithm, which addressed the shortcomings of the analytical model in [6]. In this study, we enhance that algorithm by allowing a flexible adjustment of effective airtime values based on system statistics.

3 PROPOSED CROSS-LAYER OPTIMIZATION SOLUTION

We develop an enhanced cross-layer optimization solution that dynamically distributes and allocates the available network bandwidth among various video sources in such a way that optimizes the overall detection accuracy. As illustrated in Figure 1, the system has S cameras and each one streams a different video at rate r_s to the access point, which in turn delivers it to the monitoring station thorough a high bandwidth link. Different video sources may have dissimilar physical rates. The AVS system can be expanded by considering multiple cells, each controlled by the proxy station thorough separate, uncorrelated optimization processes.

3.1 Cross-Layer Optimization Problem Formulation

The optimized bandwidth distribution problem can be solved by minimizing the sum of detection accuracy error of all video streams received by the central monitoring station. We follow the formulation in [1] and enhance it to consider the total available physical rate of the access point, as will be discussed later.

Since the wireless medium is shared by all cameras, the optimization solution should allocate a portion of the airtime to each camera. Undoubtedly, the effective medium airtime should be the upper bound of the total airtime. Precisely, the problem is formulated as minimizing the total accuracy error ($\sum_{s=1}^{S} AccuracyError(r_s)$) by finding the optimal fraction of the airtime allocation $F^* = \{f_s^* | s = 1, 2, 3, ..., S\}$ for various cameras, where r_s is the application-layer transfer rate of camera s. The optimization solution should observe the following constraints. (1) The total airtime of all cameras is equal to the effective airtime of the medium (A_{eff}). (2) The application-layer transfer rate of camera s is the product of its airtime fraction (f_s) and the medium bandwidth (Y) (i.e., total medium capacity). (3) The airtime of each camera is between 0 and 1 (inclusive). (4) The allocated bandwidth for each camera should be less than or equal to its physical-layer rate.

Mathematically, the problem can be stated as follows:

$$F^* = \arg\min_{F} \sum_{s=1}^{S} AccuracyError(r_s), \tag{1a}$$

$$Subject \quad to \tag{1b}$$

$$\sum_{s=1}^{S} f_s = A_{eff}, \tag{1c}$$

$$0 \le f_s \le 1, \tag{1d}$$

$$r_s = f_s \times Y, \tag{1e}$$

$$f_s \le \frac{y_s}{Y}, \tag{1f}$$

$$and \qquad s = 1, 2, 3, ..., S. \tag{1g}$$

Condition (1e) is improved from [1] in order to consider the total available bandwidth, and Condition (1f) is newly introduced to consider the physical limitations of the cameras in sending the encoded video streams. In [1], r_s is set as $f_s \times y_s$, where y_s is the perceived physical rate by camera s, and thus only a portion of the physical capacity of the medium is utilized by considering this constraint.

3.2 Effective Airtime Estimation

Solving the formulated optimization problem depends on estimating an accurate value for the effective airtime of the medium. Consequently, we propose an enhanced effective airtime estimation algorithm based on a novel method for estimating data dropping. This method includes value smoothing calculations of the bitrates of the received streams by the monitoring station. As shown in Figure 2, the algorithm proceeds as follows. First, the algorithm

obtains the effective throughput t_s for the video streams of camera s as received by the application layer of the monitoring station, when all cameras stream videos, each at a rate that is equal to the medium bandwidth divided by the number of cameras. The algorithm then uses this throughput to determine the initial value of the effective airtime (A_{eff}) as follows: $A_{eff} = \sum_{s=1}^{S} t_s/Y$. Subsequently, during a period of time, called estimation period, the monitoring station assesses the overall data dropping and corruption (i.e., error) rate d_s while receiving the video stream from video camera s, and then uses this information to efficiently adjust the effective airtime estimation. We determine d_s as the measured corrupted data rate for the stream plus a value capturing the difference between the nominal announced frame rate from security cameras and the actual received frame rate. In particular, we develop the following equation to assess d_s:

$$d_S = [CorruptData_s + (SSDR_s \times SumFrameDelayVar_s) \times DW]/EP, \quad (2)$$

where $SSDR_s$ is the smoothed stream bitrate for source s, $SumFrameDelayVar_s$ is the sum of variations between the frames of source s as received by the monitoring station and those produced by that source in terms of the delays between consecutive frames, DW is the delay weight constant determining the weight for the second term, and EP is the estimation period.

By using a *smoothing operation*, we can accurately estimate SSDR over the evaluation time rather than using the momentary bitrate value which yields an inaccurate date dropping rate. SSDR can be determined as follows:

$$SSDR = SC \times PSSDR + (1 - SC) \times MSDR, \quad (3)$$

where SC, $PSSDR$, and $MSDR$ are the smoothing constant, previously estimated smoothed stream data rate, and momentary stream data rate, respectively.

The monitoring station determines the overall average dropping ratio as follows: $A_\Delta = \sum_{s=1}^{S} d_s/Y$. A_Δ is used to adjust the current value of A_{eff} at the end of current estimation period. If A_Δ is equal to 0, the monitoring station increases A_{eff} by $C \times A_{thresh}$. In contrast, if A_Δ is greater than some threshold A_{thresh}, the monitoring station reduces A_{eff} by $\acute{C} \times (A_\Delta - A_{thresh})$, where A_{thresh} controls the allowable data dropping in the network and $\acute{C}$ and C are constants depending on the network characteristics. Otherwise (i.e. if A_Δ, is less than A_{thresh}), the AP increases A_{eff} by $\tilde{C} \times (A_{thresh} - A_\Delta)$ where $\tilde{C}$ is also a constant value. Guided by extensive experiments, we set C, $\acute{C}$ and $\tilde{C}$ respectively to 20, 0.8, and 16 to ensure better convergence and stability.

3.3 Cross-Layer Optimization Solution

According to [1], the accuracy error for face detection can be expressed as $AccuracyError(r_s) = a \times r_s^b + c$, where a, b, and c are constants. Thus, it can be shown that the optimization problem in Equation (1) is a budget-constrained convex problem and can be solved using the Lagrangian Relaxation method. Assuming that all cameras have the same b_s, which is practically valid, solving these two equations yields the following solution:

$$f_s^* = \left(\frac{-\lambda}{a_s Y^{b_s} b_s}\right)^{(1/(b_s-1))}, \quad (4)$$

where

$$\lambda = \left(\frac{A_{eff}}{\sum_{s=1}^{S}\left(\frac{-1}{a_s Y^{b_s} b_s y_s}\right)^{(1/(b_s-1))}}\right)^{(b_s-1)}, \quad (5)$$

We devise the following method to ensure that Condition (1f) is met: if f_s^* is larger than y_s/Y, we restart this solving process after setting f_s to y_s/Y, subtracting y_s/Y from A_{eff}, and eliminating that source from the problem domain.

Input: $\{t_1, .., t_S, PSSDR_1, .., PSSDR_S, MSDR_1, .., MSDR_S, CorruptData_1, .., CorruptData_S, sumFrameDelayVar_1, .., sumFrameDelayVar_S\}$
Output: $\{A_{eff}\}$
if this is the first time to run the algorithm
  $A_{eff} = \sum_{s=1}^{S} t_s/Y$;
At the end of each estimation period{
  For s = 1 to S\{
    $SSDR_s = SC \times PSSDR_s + (1 - SC) \times MSDR_s$;
    $d_S = [CorruptData_s + (SSDR_s \times SumFrameDelayVar_s) \times DW]/EP$;\}
  $A_\Delta = \sum_{s=1}^{S} d_s/Y$;
  if (A_Δ == 0){
    $A_{eff} = A_{eff} + C \times (A_{thresh})$;\}
  else if ($A_\Delta > A_{thresh}$){
    $A_{eff} = A_{eff} - \acute{C} \times (A_\Delta - A_{thresh})$;\}
  else{ //keep increasing A_{eff} to cause the first decrement
    $A_{eff} = A_{eff} + \tilde{C} \times (A_{thresh} - A_\Delta)$;\}
}

Figure 2: Simplified Algorithm for Dynamically Estimating the Effective Airtime

3.4 Addressing Limitation of Optimization Solution

The main limitation in the cross-layer optimization approach is that the constants of the detection accuracy model should be known a priori, but they depend on the system and surveillance site. We propose the following method to address this limitation. During initial calibration or re-calibration, the constants in the *AccuracyError* model, namely a and b, are determined offline by recording a short video of the target environment using one of the surveillance cameras at the highest supported bitrate and resolution. Subsequently the video is transcoded to different bitrates, and the proportion of detected faces relative to the original video is computed to find the *AccuracyError* value. Using the *AccuracyError* model, the values of a and b are estimated. Although the computational complexity of the transcoding is high, recomputing the constants is only needed when significant changes happen in the system or environment. In addition, readjustment of these constant values is a relaxed requirement, meaning that the values can be determined in a background process, while the system continues to operate normally. F^* is determined dynamically by applying Equation 4, during the operation of AVS system.

4 DEVELOPED AVS SYSTEM

We build a real video surveillance system and analyze the effectiveness of cross-layer optimization by providing the results of actual experiments. The system consists of a monitoring station, a variety of cameras, including PTZ cameras, all connected by a Wi-Fi network. The system employs HTTP streaming for delivering videos from the cameras to the monitoring station. To capture real deployment scenarios, we use homogeneous cameras as well as heterogeneous ones with varying capabilities and limitations. We primarily use H.264, but we consider the co-existence of different encoders in the heterogeneous case. Figure 3 illustrates a simplified view of the developed system. Different libraries, including *FFmpeg*, *SDL*, *Snappy*, and *Curl* were used to create and program different aspects of the system, involving a great deal of programming and debugging.

This system is comprised of two main sections: the monitoring station and the set of IP cameras. The IP camera set contains four PTZ surveillance cameras (IPCam 7210W), one surveillance camera (VivoTek IP7139), and two webcams (HP Truevision HD and Labtec PRO Webcam). The VLC media streaming tool is used to turn the two webcams into functional IP cameras, referred to here as *Virtualized IP cameras*. As the virtualized cameras are required to adjust their encoding and transmission bitrates based on the received control messages from the *Camera Adaptation Control* unit, we developed a program in Python to serve as an interface, called *Virtual Interface*, for these special IP cameras. The simulated Python interface allows the treatment of these cameras as regular IP cameras.

The monitoring station has four main units: *Optimization Problem Solver*, *Parallel Video Decoder and Stream Analyzer*, *Camera Adaptation Control*, and *Parallel Online Compressor and Recorder*. Video streams from IP cameras are received and decoded by the *Parallel Video Decoder and Stream Analyzer* unit. This unit analyzes the streams to accurately determine all the system parameters involved in the effective airtime estimation, bitrate smoothing, and optimization problem solving processes. We developed a multi-threaded customized video player in C++ using FFmpeg and SDL libraries to provide full control over the video decoding process in this unit and supply all the required statistics by the optimization solution. SDL library provides special multi-threading tools for media-rich applications.

The calculated and estimated system parameter values are used by the *Optimization Problem Solver* unit to determine the optimal distribution of the effective airtime of the medium among various cameras. This unit sends the medium bandwidth portion for each camera to the *Camera Adaptation Control* unit, which produces an appropriate HTTP control message for each camera and transfers it using Curl's HTTP message transfer. Finally, the cameras receive the control messages, containing the required video streaming bitrate value, and act accordingly to adjust their capturing and encoding parameters.

In the developed AVS system, the captured video streams should be evaluated by applying computer vision algorithms to evaluate different bandwidth allocation solutions. We were not able to analyze all streams in real-time, even when using a high-end workstation with 4-core Intel Core i7 running at 3.6 GHz with 16 GB of DDR3 RAM due to the combined computational complexity of computer vision algorithms. To bypass this challenge, we devised an off-line approach for the evaluation of the received video streams. Specifically, the video streams from various cameras are recorded for further analysis, without any transcoding. This requirement poses a severe challenge, as no storage device can provide the required performance and capacity at reasonable cost. In order to alleviate this challenge, we developed a recording process that can simultaneously handle writing the video streams on multiple hard disks by applying a fast lossless compression on the received data. We developed the *Parallel Online Compression and Recording* unit for this purpose using SDL and Snappy libraries. The latter provides the toolkit for fast online compression. The compressed recorded video streams are uncompressed and analyzed offline, using the *Video Analysis and Evaluation* unit, developed utilizing the OpenCV library. We stress that the recording and offline examination requirement is only for evaluation purposes and not required by the optimization solution itself. In actual AVS systems, the real-time detection should be performed by a distributed processing system.

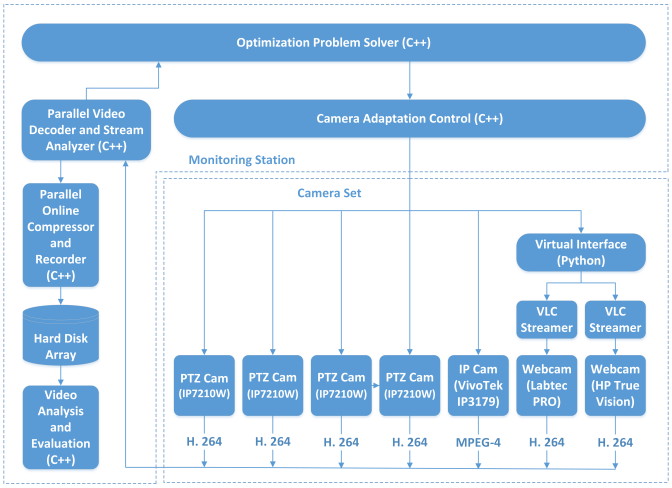


Figure 3: An Illustration of the Developed System

5 PERFORMANCE EVALUATION METHODOLOGY

Table 1 summarizes the main parameters. Extensive analysis of various design parameters indicates that A_{thresh} and *Estimation Period* values are best set to 0.0075 and 5 seconds, respectively, to enhance the performance in terms of stability and convergence.

We conduct two types of experiments. The first utilizes real surveillance videos, whereas the second deals with a real lab environment. All experiments are performed using a TP-LINK TL-WR841N wireless router as the access point and a Dell XPS 8700 workstation with 4-core Intel Core i7 running at 3.6 GHz with 16 GB of DDR3 RAM as the monitoring station. H.264 is used in all cameras except for the VivoTek IP7139, which does not support it, and thus MPEG-4 is used in that case.

Table 1: Summary of Experiment Parameters

Parameter	Model/Value(s)
Number of video cameras	4, 7
Recording Period	10 min
Application Rate	Optimized, Default = Max Access Point Rate / Number of Cameras
Video Frame Rate	Camera Dependent
Physical characteristics	Extended Rate (802.11n)
Physical Data Rate	30, 25, 20, 15, 10, 5 Mbps
State Report Interval	5 seconds
Model Constants: a, b	3103, -1.309
Used Cameras	IP7210W (4 units), HP Truevision HD, Labtec PRO Web-cam, VivoTek IP7139
Recording Resolutions	1280×720, 800×600, 640×480

We compare the proposed solution, referred to as **NS**, with the existing solution (**ES**) in [1]. We also analyze the case when the optimization is disabled, referred to as **DO**. The main analyzed metric is *face detection accuracy*, measured in terms of the overall number of detected faces. We use *OpenCV* to run the Viola-Jones algorithm on the decoded video streams.

5.1 Experimental Setup I: Using Real Video Surveillance Dataset

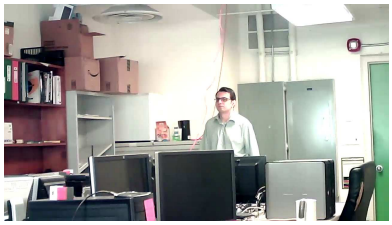
In Experimental Setup I, we use different types of cameras (discussed in Section 4) to capture real surveillance videos rendered on separate monitors to provide repeatable, diverse, and realistic scenery. The videos have different characteristics and come from different environments, including *office*, *campus*, *stores*, and *busy streets*. Table 2 summarizes the main characteristics of the video dataset, which will be publicly available. The original videos, collected from YouTube and other sources, are truncated to have nearly equal total video duration for each category.

Table 2: Summary of Video Files Characteristics Used in Experimental Setup I

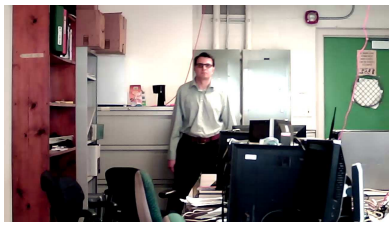
Environment Type	Resolution	Duration (sec)	Frame Rate (fps)	Bitrate (Kbps)
Campus	1280 × 720	27	29	2704
Campus	1920 × 1080	44	23	3145
Campus	1280 × 720	77	30	2149
Office	1920 × 1080	10	30	2317
Office	480 × 360	8	30	343
Office	1280 × 720	57	30	2098
Office	1280 × 720	32	25	2088
Office	640 × 360	42	29	575
Store	480 × 360	57	6	355
Store	370 × 252	23	23	363
Store	1280 × 720	69	23	2550
Street	1280 × 720	14	30	768
Street	1920 × 1080	27	23	4215
Street	1920 × 1080	40	23	4035
Street	1280 × 720	68	29	2199

5.2 Experimental Setup II: Real Laboratory Environment

In Experimental Setup II, four PTZ IP-Cam 7210W cameras are used to capture videos from a real lab. These cameras receive target bitrates from the monitoring station (which runs the optimization solution) and produce and transmit the H.264 video streams. The monitoring station also controls the movements of these PTZ cameras, providing a controlled patrol movement for each camera. The lab includes a person acting based on a predefined script in each evaluation session to allow fair comparisons among different bandwidth allocation solutions. The script includes descriptions of the paths that must be traversed by the acting person, the standing and walking directions, and the required time that should be spent on each specific path. Figure 4 displays sample concurrent views from two PTZ cameras.



(a) Camera 1



(b) Camera 4

Figure 4: Sample Concurrent Views of Cameras in Experimental Setup II

6 RESULTS PRESENTATION AND ANALYSIS

6.1 Tuning of System Constants

Let us first discuss how the values of the delay weight (*DW*) and smoothing constant (*SC*) can be selected. Figure 5 illustrates the effect of these two constants on the number of detected faces. This number increases initially with *DW* because of the tendency towards producing higher frame rates. This tendency is satisfied by reducing the stream bitrates, which reduces the contention for the available medium bandwidth, reducing the dropped and damaged data transmitting by the cameras. This reduction in turn increases the perceived frame rate by the monitoring station. After a certain point, however, the produced high frame rates greatly reduce the stream bitrates and thus quality. In the considered system configuration, a value of 0.65 for *DW* achieves the best detection accuracy. Similarly, the number of detected faces increases with *SC* up to a certain point and then decreases. The increase happens because

the system can estimate the stream bitrates more accurately via bitrate smoothing, resulting in a better effective airtime estimation. By contrast, the subsequent decrease is due to aggressive smoothing, greatly marginalizing the impacts of the momentary values of the stream bitrates. In the considered system configuration, it is best to set *SC* to 0.99.

6.2 Comparing Effective Airtime Estimation under Different Solutions

As shown in Figure 6, the proposed solution (NS) produces the largest area under the effective airtime curve. In particular, its area is 125% larger than the best existing solution (ES) and 8% larger than the area resulted when the optimization is disabled (DO). As discussed in Subsection 3.1, ES causes the system to use only a portion of available bandwidth, and thus the effective airtime plummets to even lower values than DO!

6.3 Analysis of Cross-Layer Optimization

Let us now demonstrate the effectiveness of cross-layer optimization in the terms of detection accuracy under Experimental Setup I. Figure 7 shows the number of detected faces versus bandwidth capacity for the entire video dataset and for each video category. The proposed solution (NS) achieves 19%, 10%, 10% and 10% higher accuracy than ES in campus, office, store and street environments, respectively and 54%, 45%, 72% and 69% higher accuracy than DO. As expected, the number of detected faces generally increases with the medium capacity. The occasional dips in the case of ES are due to the aforementioned problem in utilizing the medium bandwidth. After a certain point, NS and DO converge to similar values, when the medium capacity is large enough to accommodate the maximum supported bitrates of the cameras, thereby eliminating the contention for the available bandwidth. *In actual systems, the number of cameras and the maximum supported bitrates will be larger, thereby increasing the medium bandwidth at which convergence occurs.* Interestingly, ES performs worse than DO when the contention among cameras goes below a certain level. In Figure 8, Cam1, Cam2, Cam3 and Cam4 refer to IPCam 7210W wireless PTZ security cameras. Cam5 and Cam6 refer to HP Truevision HD and Labtec PRO web-cams, which are turned into functioning IP cameras using VLC media player and Cam7 is related to the VivoTek IP7139 camera. This figure, compares various solutions in detection accuracy for each camera. NS consistently performs the best, whereas in certain cameras, ES performs worse than DO due to its aforementioned problem.

Figure 9 compares various solutions in terms of the number of detected faces under Experimental Setup II (real lab environment). This comparison is done by determining the number of detected faces in the streams received from various cameras while operating under different solutions. The results for each individual camera as well as the entire surveillance system is demonstrated. Although there is only one person in the scene, the person appears in multiple frames of the video streams and having a higher quality video stream translates to a higher number of detected faces. The proposed solution achieves 123% higher accuracy than ES and 148% higher than DO.

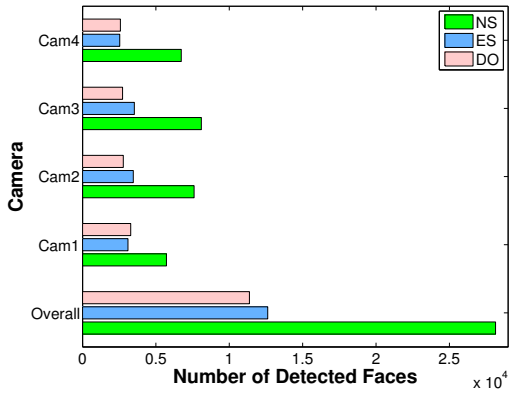


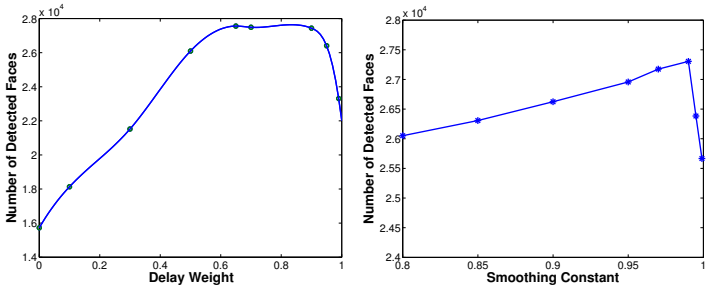
Figure 9: Comparing Various Solutions in Detection Accuracy in the Real Lab Environment [Experimental Setup II]

Finally, let us shed light onto the dynamics of the system as a result of the interplay of various factors. Figure 10 demonstrates the relationships among different aspects and attributes of an AVS system. The different solutions are compared in terms of the average received rate by the monitoring station, effective airtime, frame rate, and number of detected faces. To effectively display the different values of attributes together in the same chart, attribute values were normalized. The number of detected faces is the most prominent metric, and indeed *NS* continues to hold the lead in that metric. Achieving a high value in this metric depends on two main factors: the effective airtime and the average received frame rate. *NS* demonstrates perfect balance in improving these two main factors, resulting in producing the highest face detection accuracy. In contrast, *ES* produces the second best results, with low medium bandwidth usage. When the power consumption is of utmost significance and the medium bandwidth is low, ES becomes a viable choice. The results also demonstrate that cross-layer optimization is greatly important. The number of detected faces in *DO* is considerably lower than the two optimization solutions. Fortunately, the reason behind this behavior is easy to rationalize. When the available medium bandwidth is limited, having each camera sending at the highest rate without any governing policy would cause sever congestion in the network, thereby increasing the chance of data packet losses and frame droppings, and subsequently resulting in a severe decrease in the received frame rate at the monitoring station. This regulation of bandwidth allocation (via cross-layer optimization) is greatly important, especially in cases where a subset of cameras have considerably lower physical rates than the rest, causing significant differences in the number of detected faces among the cameras when no cross-layer optimization is used.

7 CONCLUSIONS

We have built a real video surveillance system and have provided the results of actual experiments.

The main results can be summarized as follows. (1) Cross-layer optimization in the AVS systems provides great benefits. (2) High quality video stream at high frame rates can be received by the monitoring station by optimally distributing the available medium



(a) Impact of Delay Weight (b) Impact of Smoothing Constant

Figure 5: Impact of Smoothing Constant and Delay Weight on Detection Accuracy [Experimental Setup I, 15 Mbps Medium Bandwidth]

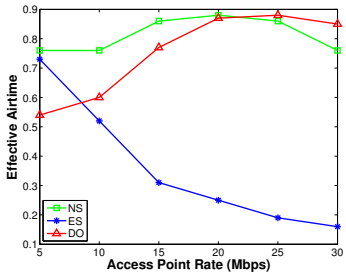
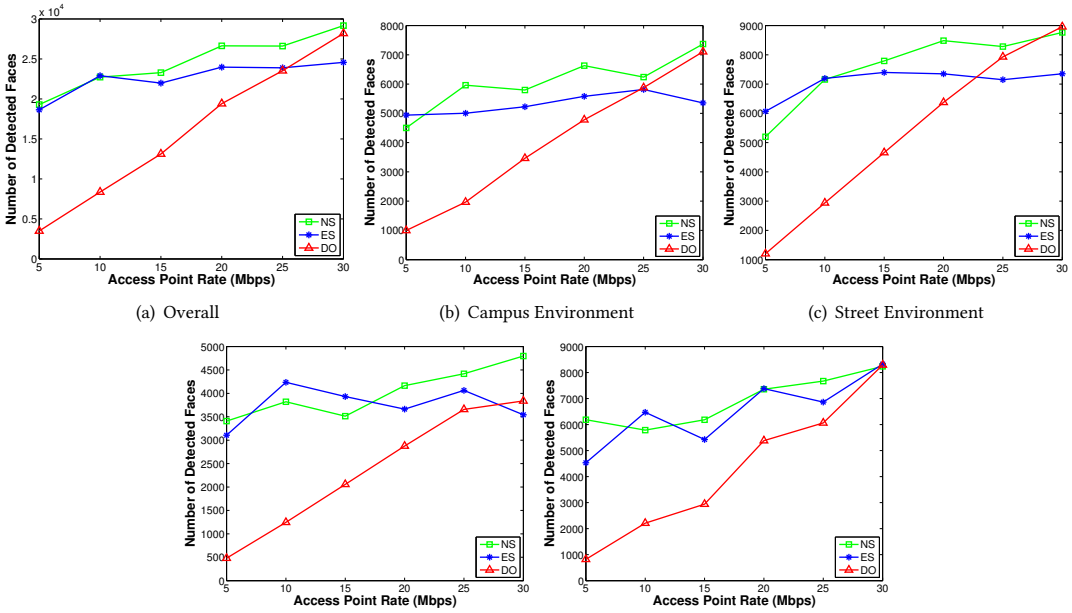


Figure 6: Comparing Effective Airtime with Different Solutions [Experimental Setup I]



(a) Overall (b) Campus Environment (c) Street Environment

(d) Store Environment (e) Office Environment

Figure 7: Comparing Various Bandwidth Allocation Solutions for Each Video Category [Experimental Setup I]

bandwidth and increasing the effective medium airtime. (3) The proposed solution significantly enhances the detection accuracy. (4) The proposed effective airtime estimation algorithm provides high accuracy by properly calculating the overall data corruption and dropping rate and using bitrate smoothing calculations. (5) The highest detection accuracy is achieved when the packet dropping and error rate is very small. (6) A distributed processing system is required for the real-time detection of threats from a large number of streams simultaneously.

REFERENCES

[1] Mohammad Alsmirat and Nabil J. Sarhan. 2016. Cross-Layer Optimization for Automated Video Surveillance. In *2016 IEEE International Symposium on Multimedia (ISM)*. 243–246.

[2] Mohammad A. Alsmirat and Nabil J. Sarhan. 2012. Cross-Layer Optimization and Effective Airtime Estimation for Wireless Video Streaming. In *2012 21st International Conference on Computer Communications and Networks (ICCCN)*. 1–7.

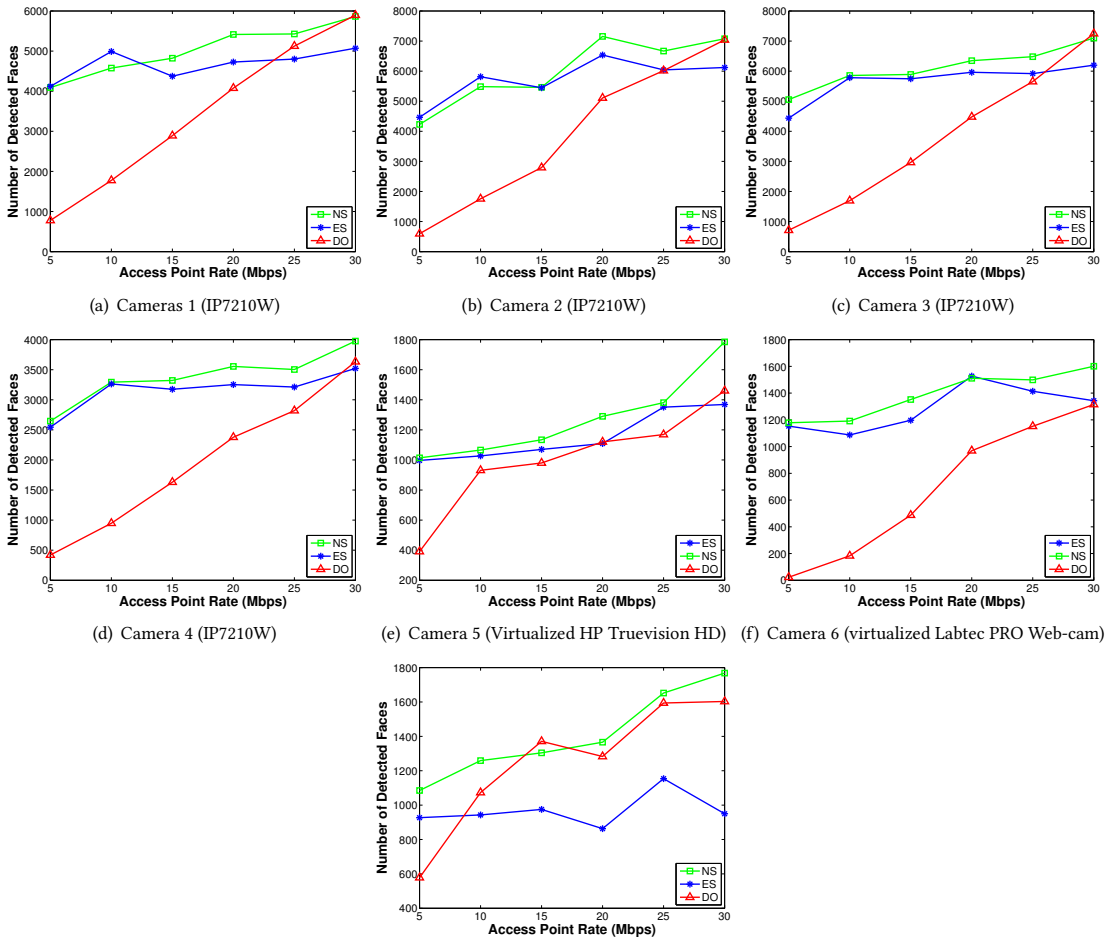
[3] Qiang Chen, Zheng Song, Jian Dong, Zhongyang Huang, Yang Hua, and Shuicheng Yan. 2015. Contextualizing Object Detection and Classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37, 1 (Jan 2015), 13–27.

[4] Kai-Wen Cheng, Yie-Tarng Chen, and Wen-Hsien Fang. 2015. Gaussian Process Regression-Based Video Anomaly Detection and Localization With Hierarchical Feature Representation. *IEEE Transactions on Image Processing* 24, 12 (Dec 2015), 5288–5301.

[5] Zhihai He and Dapeng Wu. 2006. Resource allocation and performance analysis of wireless video sensors. *IEEE Transactions on Circuits and Systems for Video Technology* 16, 5 (May 2006), 590–599.

[6] Cheng-Hsin Hsu and Mohamed Hefeeda. 2011. A framework for cross-layer optimization of video streaming in wireless networks. *ACM Trans. Multimedia Comput. Commun. Appl.* 7, Article 5 (February 2011), 28 pages. Issue 1.

[7] Jian Huang, Zhu Li, Mung Chiang, and Aggelos K. Katsaggelos. 2006. Pricing-based Rate Control and Joint Packet Scheduling for Multi-user Wireless Uplink Video Streaming. In *Proc. 15th International Packet Video Workshop (PV2006)*.



(a) Cameras 1 (IP7210W) (b) Camera 2 (IP7210W) (c) Camera 3 (IP7210W)

(d) Camera 4 (IP7210W) (e) Camera 5 (Virtualized HP Truevision HD) (f) Camera 6 (virtualized Labtec PRO Web-cam)

(g) Camera 7 (VivoTek IP7139)

Figure 8: Comparing Various Solutions in Detection Accuracy Using the Entire Video Dataset [Experimental Setup I]

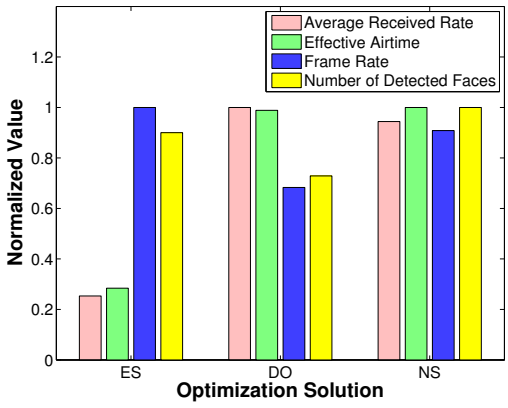


Figure 10: Relationships Among Different Attributes of an AVS System [Experimental Setup I]

[8] IEEE. 2007. IEEE 802.11 Standard Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. (12 2007), C1–1184.

[9] Xiaoan Lu, Elza Erkip, Yao Wang, and David Goodman. 2003. Power efficient multimedia communication over wireless channels. *IEEE Journal on Selected Areas in Communications* 21, 10 (Dec 2003), 1738–1751.

[10] Sai Shankar N and Mihaela van der Schaar. 2007. Performance Analysis of Video Transmission Over IEEE 802.11a/e WLANs. *IEEE Transactions on Vehicular Technology* 56, 4 (July 2007), 2346–2362.

[11] Mihaela van der Schaar, Yiannis Andreopoulos, and Zhiping Hu. 2006. Optimized Scalable Video Streaming over IEEE 802.11a/e HCCA Wireless Networks under Delay Constraints. *IEEE Transactions on Mobile Computing* 5 (June 2006), 755–768. Issue 6.

[12] Honghai Zhang, Yanyan Zheng, Mohammad Ali Khojastepour, and Sampath Rangarajan. 2010. Cross-layer optimization for streaming scalable video over fading wireless networks. *IEEE Journal on Selected Areas in Communications* 28, 3 (2010), 344–353.