Adaptive Home Surveillance System using HTTP Streaming

Duc V. Nguyen, Hung T. Le, Anh T. Pham, Truong Cong Thang
University of Aizu
Aizuwakamatsu City, Japan
{s1180245, m5171120, pham, thang}@u-aizu.ac.jp

Jin Young Lee, Kugjin Yun
Electronics and Telecommunications Research Institute
Daejeon, South Korea
{jinlee, kjyun}@etri.re.kr

Abstract—In this paper, we investigate the use of HTTP streaming in home surveillance system. First, we propose a dynamic resource allocation framework among the cameras. Under a set of constraints, e.g., link bandwidth and delay, our resource allocation solution decides which video versions will be requested from which cameras in order to maximize the weighted sum of utility for all cameras. The optimization, which is performed at the client, is based on a utility model for each video stream that takes into account not only perceptual quality but also the delay of a video stream. Video content is obtained through HTTP request/response transaction between the client and the cameras. The experiment results show that the proposed method is able to effectively improve the overall utility while still supporting low end-to-end delay and good perceptual quality for important cameras.

Keywords—HTTP streaming, surveillance system, resource allocation, end-to-end delay, GOP.

I. INTRODUCTION

Video-based surveillance system [1][2] has been widely applied in many places such as airports, train stations, banks, etc. The general purpose of a surveillance system is to help detecting and preventing illegal and unwanted activities. Recently, due to the widespread increase of security threats, e.g., terrorism, the use of surveillance technologies has become an essential requirement for organizations and individuals. However, most existing surveillance systems still depend on human operations. That raises the issue of having a surveillance system which effectively supports human operators to make proper decisions.

Thanks to the development in both hardware and video processing technologies, surveillance cameras have become smarter and more popular. Today, cameras usually are equipped with a processing unit that is able to detect, recognize and analyze the events in captured video stream in real time. In addition, they have large memory for data storage and the ability to connect to the Internet as other network devices.

Recently, HTTP streaming has become a popular approach for video delivery over the Internet [3][4]. The use of standard Web servers as the streaming servers and HTTP as the delivery protocol brings several advantages compared with the traditional streaming services [3]. Different from conventional streaming technologies, the client in HTTP streaming fully controls the streaming process, i.e., make decision on which/when the video content will be requested; which/when the received data will be played out. To support adaptivity to heterogeneous network and terminal capabilities, the provider/server should generate multiple versions (or alternatives) of a video and each of the versions is further divided into a sequence of segments. The metadata (called Media Presentation Description – MPD) describing the characteristics of the versions is also made available to the client. Then, based on the metadata and network/terminal status, the client sends request for the most appropriate video version.

In this paper, we exploit the advantages of smart camera and HTTP streaming technology in home surveillance system. Obviously, among a home’s cameras, some important cameras should have low delay and high video quality in order to efficiently support surveillance tasks, while others may have long delay and low bitrate/quality. Our aim is to enable the client to dynamically adapt to the changes in surveillance context without deploying complex streaming mechanisms at the servers.

In our solution, the home surveillance problem is represented as a dynamic resource allocation framework among cameras. Under a set of constraints, e.g., link bandwidth and end-to-end delay, our resource allocation solution decides which video versions will be requested from which cameras in order to maximize the weighted sum of utility for all cameras. The optimization is performed at the client and the decision is based on a utility model for each video stream that takes into account not only perceptual quality but also the delay of a video stream [5].

The rest of the paper is organized as follows. In Section II, a review of related work is provided. The resource allocation problem in a home surveillance system is presented in Section III. Our solution to optimization problem is given in Section IV. The experiment results are given in Section V, and finally the paper is concluded in Section VI.
II. RELATED WORK

Rate allocation problem for surveillance system has been addressed in previous studies. Most existing works proposed server-side approach to resolve the problem.

Two server-side methods are proposed in [7][15]. In [7], the problem is considered in the context of streaming multiple video objects. The authors proposed adaptation framework by taking into account both the selection and the transcoding of multiple video stream at different bitrates. A perceptual priority based packet scheduling for video transmission in an ad hoc live surveillance system is proposed in [15]. The original video content is encoded into several versions of different perceptual priorities. The source node then schedules sending packets to minimize the network delay under a given visual quality constraint. In the studies, perceptual quality is used as the metric to evaluate the utility of a video stream. [6] is the first study that considers the tradeoff between delay and perceptual quality in video surveillance system. They proposed delay-aware rate control scheme that adapt Quantization Parameter (QP) for the best tradeoff between delay and distortion. Specifically, the QP for the current frame is chosen to minimize the Lagrangian cost of encoded distortion and the product of expected delay and frame size. This method deals with transport delay caused by network and the task of rate control is performed completely at the server.

There are few works that deal with the problem by client-side solution. Xiaooung et al. [1] proposed a client-side rate allocation framework for multi-camera surveillance over an ad hoc wireless network. Allocated rate for each camera is determined based on Rate-Distortion (R-D) characteristic of video stream from the cameras and conformed by adapting QP. Ahmad [2] investigated video surveillance system on public transport using WiMAX technology. In their method, the client estimates the available bandwidth and stops live feeding from some cameras which have low utility in order to maintain the minimum bandwidth demand for the remaining ones. A utility model that is based on the ratio of the receiving rate to the bandwidth requirement and cameras’ locations has been proposed. The two methods have the same characteristic in which only rate selection is performed at the client while still rely on streaming mechanism at the server for video transmission.

Regarding latency in surveillance system, a measurement is performed for both analogue systems and digital systems [9]. The results demonstrate that while analogue cameras do have a lower latency, most IP cameras are in within acceptable tolerances. Birisan et al. [8] investigated whether Dynamic Preference Specification (DPS), which gives users the ability to dynamically specify preferences related to frame rate and resolution, positively affects the ability to achieve surveillance tasks in wireless surveillance network.

In most surveillance systems, the servers fully control the streaming process. This approach therefore results in expensive cost for deployment of surveillance system as well as limits the system’s scalability. As our system is based on the new trend of HTTP streaming, control task is completely moved to the client. Every decision is performed by the client while each camera acts as a content provider. Therefore, system scalability is significantly improved. In this work, we analyze system’s delay at media segment level. We also propose a new utility model combining perceptual quality and waiting time for each video stream to eliminate the limitation of previous methods.

III. PROBLEM DESCRIPTION

A. GOP size and end-to-end delay

Live streaming services usually require a very tight constraint on the end-to-end delay, i.e. the time elapsed from when a picture is captured by the video camera until it is displayed on the user’s monitor [10]. For instance, an end-to-end delay exceeding 500 ms is unacceptable for video conferencing applications. It should be noted that, in today’s modern IP networks, the transport delay has been significantly improved (about tens of milliseconds) [11].

In HTTP streaming, segment duration forms a dominant part in the end-to-end delay as shown in [10]. Therefore, in order to enable low end-to-end delay for important cameras, the segment duration should be chosen as the Group of Picture (GOP) duration. A Group of Picture (GOP) starts with an I-frame that contains the entire image following by a sequence of B-frames and/or possibly P-frames. Because a video encoder can only start generating the video segment once a GOP is completely captured, the selection of the GOP size causes a major impact on the processing delay at the server. As a result, the end-to-end delay will depend mainly on the value of GOP size.

The end-to-end delay $D_{e2e}$ consists of two main components, namely processing delay $D_{\text{Gap}}$ and transport delay $D_{\text{net}}$, and is given as follows.

$$D_{\text{e2e}} = D_{\text{Gap}} + D_{\text{net}} .$$

Currently, our focus is to control the delay caused by GoP size. As the home network and Internet connection are good nowadays, we suppose that $D_{\text{net}}$ is small and known in advance.

B. Problem formulation

We consider resource allocation problem among cameras in a home surveillance system. In order to provide such a multiple source streaming service, we assume that video sequence captured at each camera is encoded into multiple versions of different bitrates and different GOP sizes. Next, the decision engine located at the client will decide which versions to be requested from which cameras in order to maximize the overall utility received by the user.

Let us denote $M$ as the number of cameras in the system and $N_i$ $(1 \leq i \leq M)$ as the total number of video versions provided at the $i^{th}$ camera. With each video stream $O_{ij}$ which is the $j^{th}$ version provided at the $i^{th}$ camera, denote $U(O_{ij})$ as the corresponding utility for the video stream. At a given time, under an available throughput constraint $R$ and delay constraint $D$, we have the problem statement as follows:

Find $(O_{ij})$ for each video camera to maximize the overall utility $OU$ that is defined by:
subject to

\[ \sum_{i=1}^{M} R_{ij} \leq R^e \]

and \[ D_{ij} \leq D^e \] for all \( i, j \).

where \( w_i \) is the weight of the \( i \)-th camera; \( R_{ij} \) and \( D_{ij} \) are the bitrate and the delay \( D_{ij} \) of video stream \( O_{ij} \).

The value of \( w_i \) reflects the importance of the scene the \( i \)-th camera is monitoring. Initially, weight value for each camera can be assigned based on user preference. Since video events may change over time, these values could be adjusted in order to cope with various circumstances (e.g., event detector at a camera detects a suspect). We assume that each camera can detect the activities of the scene and inform the client via a signaling channel.

IV. OPTIMIZATION SOLUTION

In a video surveillance system, perceptual quality for a video stream must not be lower than a certain limit where the security experts cannot interpret the video content. Hence, the utility of a streaming video decreases significantly if video quality is not maintained sufficiently. Along with that, another key requirement is that the video delay should be as small as possible for quick reaction when strange events occur. The client may achieve this objective by selecting the video version of low GOP size; however, this results in higher required bandwidth at the same quality level. Considering the trade-off between the perceptual quality and the end-to-end delay, we propose the utility model for a video stream as below.

\[ U(O_{ij}) = a_1 U_q(O_{ij}) + a_2 U_d(O_{ij}) \]

where \( U_q \) is a function of video bitrate that represents the perceptual quality of the video stream; \( U_d \) is a function of the end-to-end delay caused by the video stream that represents the impact of the waiting time on user’s experience; and \( a_1 \) and \( a_2 \) are the weights for \( U_q \) and \( U_d \), respectively. Subjective test methods could be applied in order to determine the appropriate value for the weights. Currently, we assume that these values are known in advance and leave the problem for future research.

In our model, \( U_q \) is represented in terms of Peak Signal to Noise Ratio (PSNR). For solving the optimization problem, the utility values of video versions at different bitrates and GOP sizes are provided as empirical data in the form of metadata (MPD) [4][14]. Fig. 1 shows some utility curves for video streams recorded by a camera. At each camera, the curves corresponding to five GOP sizes, which are 1, 2, 4, 8 and 16, are provided. For the same perceptual quality, we can notice that the selection of GOP size equal to 1 results in the highest video bitrate. Meanwhile, video version at GOP size of 16 requires the smallest bitrate to achieve equivalent quality result. Obviously, the curves are different for cameras depending on the scene characteristics. The normal trend is that encoding efficiency increases rapidly as the GOP size increases up to around 32. A GOP size exceeding 32 usually does not provide any significantly improvement in encoding efficiency.

Meanwhile, \( U_d \) expresses the impact of the end-to-end delay on the utility of the video version. Evaluating the influence of the end-to-end delay on streaming services is a complicated task. In [5], a study on the influence of the initial delay on Web-based services can be found. A logarithmic relationship between the initial delay and user’s Quality of Service is introduced and confirmed by performing real test. In this paper, the relationship between the end-to-end delay and the delay utility is modeled using the zapping time – MOS relationship in IPTV services that has been studied in [11]. This delay utility model is shown in Fig. 2.

Since the problem space of home surveillance scenario is small, a full-search procedure is applied to solve the optimization problem. Firstly, the utilities of all video versions from cameras are calculated using (4). Then, the client searches for video versions that maximize the overall utility \( U \) calculated using (2) and satisfy the constraints (3). In large scale network such as smart city, fast algorithm should be used to reduce the searching time, such as dynamic programing [13][14].

\[ \text{Fig. 1. } U_q \text{ as the function of bitrate at different GOP sizes} \]

\[ \text{Fig. 2. } U_d \text{ as the function of delay} \]

V. EXPERIMENT RESULTS

The general architecture of our system is shown as in Fig. 3. Video content from camera sensors is prepared at the server by Media preparation block. In addition, event detection block
will detect any changes in the scenes. Information related to video version is described in MPD file that is sent to the client at the starting of streaming session. Real-time information about the amount of activity of each camera is updated regularly and retrieved by the client in the same way as media segments. It should be noted that the mapping from events or activities to the weight value of each camera is reserved for our future work.

![General architecture of our developed system](Image)

Our test-bed consists of a number of cameras and a client connecting to an access network. The client is implemented using Java and C language and run on a Windows Vista Home Basic computer with Core2Duo processor and 2G RAM. We used the common network emulator, i.e. Dummy Net for network simulation. Each camera is connected to a home gateway (server), which has the ability to provide video streams at different quality levels and GOP sizes. Video component has frame rate of 30fps and resolution of 177x144. FFmpeg tool [16] is used to create video content at the server in real time. The network delay between server and client is set to 40ms and end-to-end delay constraint to 1s. Video content is provided at 16 bitrates from 64 Kbps to 2048 Kbps with step size of 128 Kbps and at 5 GOP sizes which are 1, 2, 4, 8 and 16. The GOP size exceeding 16 results in high end-to-end delay that is not suitable for surveillance system.

In the first part of the experiments, we investigate performance of our method compared with the simple method where GOP sizes are fixed and the same for every video encoded by the cameras. We use 4 cameras in this experiment. The weights for the cameras are set to 1 for every camera. The utility model parameters \( a_1 \) and \( a_2 \) are set to 0.8 and 0.2, respectively. In the model, both perceived quality and delay utility are normalized into \([0, 1]\) interval.

Fig. 4 shows the experiment result of our method and two fixed GOP size cases, i.e. GOP size of 1 and GOP size of 16, which are the two extreme cases. The former has the smallest end-to-end delay while the later provides the best bandwidth utilization. We can note that the proposed method outperforms both of the fixed GoP methods by a margin of 0.1 to 0.3, equivalent to 0.4 dB to 1.2 dB in PSNR scale. When the available throughput is low, the selection of GOP size = 16 provides an overall utility similar to that of GOP size = 1. But its performance is the lowest when the throughput is higher than 3000 Kbps. The selection of GOP size = 1 shows poor performance as the throughput is low. But the obtained utility increases steadily w.r.t. the available network throughput. When the throughput is around 12Mbps, it is the same with the optimal result obtained by the proposed method. This is because both methods select the GoP size of 1 at this high throughput.

![Utility comparison of the proposed method with fixed GOP size methods. Four cameras are used in this case.](Image)

<table>
<thead>
<tr>
<th>Video #1</th>
<th>Video #2</th>
<th>Video #3</th>
<th>Video #4</th>
</tr>
</thead>
<tbody>
<tr>
<td>priority</td>
<td>Bitrate</td>
<td>GOP size</td>
<td>Bitrate</td>
</tr>
<tr>
<td>0.5</td>
<td>704</td>
<td>16</td>
<td>1984</td>
</tr>
<tr>
<td>1</td>
<td>1216</td>
<td>8</td>
<td>1472</td>
</tr>
<tr>
<td>1.5</td>
<td>1984</td>
<td>2</td>
<td>1472</td>
</tr>
<tr>
<td>2</td>
<td>1984</td>
<td>2</td>
<td>1216</td>
</tr>
<tr>
<td>2.5</td>
<td>2880</td>
<td>1</td>
<td>1088</td>
</tr>
<tr>
<td>3</td>
<td>2880</td>
<td>1</td>
<td>1088</td>
</tr>
</tbody>
</table>

In the second part of the experiment, we investigate the behavior of the client in order to cope with the changes in system condition. We used the same configuration as in the previous experiment except camera’s priority values. Now, we fix the priority of the video cameras #2, #3, and #4 to 1 and adjust the priority of the video camera #1. The selected video versions at each video camera when the network throughput is 6Mbps are shown in Table I. For video camera #1, we can note that the GOP size of the selected video version decreases as the priority increases. For instance, when the priority value is set to 2, the client selects the video version at bitrate of 1216 Kbps and GOP size of 2. The client selects the video version at the lowest GOP size, i.e. GOP size equal to 1, when the priority value is 3. On the other hand, when the priority value is reduced to 0.5, the GOP size of 16 is chosen to save more resource for other cameras. The results indicate that important cameras, i.e. high priority, will have small end-to-end delay that may help the user to quickly follow unusual events, while for less important camera, a larger GoP size is selected for saving network resources.
The result in case of 8 cameras are used is shown in Fig. 5. We can see that due to the increased number of cameras, the overall utility is also increased as well. Nonetheless, the benefit of the proposed method is similar to the previous case.

![Utility comparison of the proposed method with fixed GOP size methods. Eight cameras are used in this case.](image)

**VI. CONCLUSION**

In this paper, we have studied the use of HTTP streaming in home surveillance system. A general problem formulation for bitrate adaptation problem was introduced with consideration to not only the perceptual quality but also the impact of the GOP size on the system performance. Experiment results showed that the proposed method has better performance in comparison with existing methods. Especially, video versions of appropriate GoP sizes and bitrates were selected from each camera according to the importance of the cameras and available throughput. In the future work, we will focus on large scale surveillance system where fast algorithms are needed in solving the optimization problem.

**REFERENCES**


