

# Implementation of a Rate-Adaptive Approach for H.264 Streaming Traffic Control on Da Vinci Dual Core Platform

Y.J. Chen

Ming Chi University of Technology  
84 Gungjuan Rd., Taishan Dist.  
New Taipei City, Taiwan

D.Y. Shih

Ming Chi University of Technology  
84 Gungjuan Rd., Taishan Dist.  
New Taipei City, Taiwan

J.J. Huang

Ming Chi University of Technology  
84 Gungjuan Rd., Taishan Dist.  
New Taipei City, Taiwan

**Abstract**—Video surveillance becomes far-reaching thanks to the Internet. The IP camera having both photography and video output functions becomes the primary component in network video surveillance; the bandwidth of the network passageway is fixed but the traffic is changing. In the event of network jam, there must be ways to avoid missing surveillance screen or lag. Therefore, this study proposes a practical approach to make H.264 video encoder dynamically adjust the compression parameters, such as the size of the quantization value, the video frame resolution, and the number of frames generated in each second and more, through a traffic control program to produce adequate video traffic. By doing so, even in the event of insufficient bandwidth at the user-end, the packet dropping and delaying will be avoided and video frames are able to be transmitted to the user in time.

**Keywords**—rate-adaptive traffic control; H.264; streaming, RTP; LIVE555; DaVinci platform

## I. INTRODUCTION

When multiple users are using the network with fixed and limited bandwidth, as there might be a massive amount of packets to be transmitted at the same time, the network equipment will put the current packet into the buffer for the next clock. In a more serious condition this packet will be discarded because the influx of a large number of packets instantly will cause unstable network traffic. The video bit stream generated from a video being compressed by H.264 algorithm has two models of source rate [1-3]. One as Variable Bit Rate (VBR) and the other is Constant Bit Rate (CBR). General IP camera (IP Cam) has the video compressed output flow default as CBR, which will dynamically adjust the Quantization Parameter (QP) value according to the complexity and change degree of video frames to make the fragments of complicate or frequently changing frames have thicker quantization scale and simple and stable fragments have finer quantization scale; thus to maintain the source rate of the video. Therefore, when the network becomes congestion that the available bandwidth is insufficient to support the fixed traffic of the camera, it will lead to a delay of video packet

transmission causing incomplete or missing frames at the target-end. VBR fixes the QP value of the video compression therefore the compressing speed of fragments without too many changes is low and vice versa. Thus the source rate is commensurate with the complexity and change degree of the screen. This study focuses on the CBR mode and proposes a rate-adaptive approach, predicting the next-cycle available bandwidth and then requesting the H.264 encoder to change its output rate to the predicted available bandwidth by adjusting the compression parameters, such as the QP value, the frame resolution, and the frames-per-second value and more. The proposed approach has been implemented in a client-server environment; the server end is a TI (Texas Instruments) DaVinci dual core platform with H.264 video encoder and LIVE555 streaming server [4-7], and the client end includes a VLC (Video LAN Client) player and a rate-control program of bandwidth detection and prediction.

## II. LITERATURE REFERENCES

There are many products and literatures related to network video surveillance. Most of IP Cam products only emphasize the techniques of video image analysis and compression. IP Cams expend large power for compression calculation; however, whether the CBR or VBR mode an IP Cam is set to, its source rate is produced according to the video frame but not the available bandwidth of the network. This also ignore the network traffic that may have tremendous change so when the images are being compressed, some delay occurred. This is the reason why the network surveillance system that has diverse functions and clear view on the target is not being used by the general public. Mainly it is because IP Cams cannot immediately or effectively provide dynamically video compression control aiming on the network traffic. Thus, when the network traffic is congestion, it cannot provide effective images, not even to transmit the images to the user end immediately.

### III. H.264 INTRODUCTION

H.264 video encoding is the advance video coding standard established on the basic of MPEG-4 technique. The formal name is H.264/AVC. The coding process mainly comprise of five parts: inter-prediction and intra-prediction, transform, quantization, loop filter, and entropy coding [4]. The video encoding mechanism of H.264 uses blocked-based unit. In other words, it cuts the whole frame into many rectangle zones called macro block (MB) and then encodes these macro blocks, as shown in Figure 1. First adopt intra-prediction and inter-prediction technique to eliminate the similarity among images to get the so-called residual frame, which will be transformed and quantized to eliminate visual redundancy [2]. Finally entropy coding, in video coding layer (VCL), will output the encoded bit stream and then pack as NAL-unit packet (NAL, network abstract layer) to be transmitted through network to the remote end or store in the storage media.

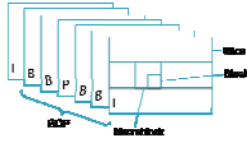


FIGURE 1. H.264GOP AND THE CODING MECHANISM [3].

### IV. RTP PROTOCOL

The NAL-unit packet of H.264 is packetized in the packet of Real-time Transfer Protocol (RTP). RTP is an end-to-end communication protocol of OSI Layer 5. RTP is used in conjunction with the RTP Control Protocol (RTCP). While RTP carries the media streams (e.g., audio and video), RTCP is used to monitor transmission statistics and quality of service (QoS) and aids synchronization of multiple streams. As RTP uses UDP protocol to transmit (Figure 2), it only has the basic function of providing the real-time information such as timestamp and sequence number[4].

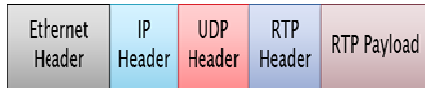


FIGURE II. RTP FORMAT.

Real Time Streaming Protocol (RTSP) is a communication protocol controlling the multimedia service. The syntax of RTSP is very similar to that of HTTP, a pure text communication protocol. While HTTP is stateless, RTSP has state; an identifier is used when needed to track concurrent sessions. Like HTTP, RTSP uses TCP to maintain an end-to-end connection and, while most RTSP control messages are sent by the client to the server, some commands travel in the other direction (i.e. from server to client). The relationship among RTP, RTCP, and RTSP is shown in Figure 3.

In this study, we use LIVE555 program [7] to set up the streaming server. This program is able to produce RTP, RTCP, RTSP, and SIP standard agreement and support MPEG, H.264, H.263, DV, and JPEG formats. The framework of transferring the H.264 frame with RTP packets by using LIVE555 and transmit them to the user is shown as Figure 4.

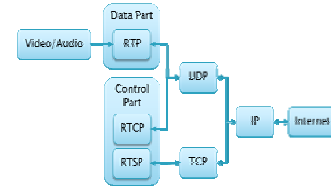


FIGURE III. FRAMEWORKS OF RTP, RTCP, RTSP[5].

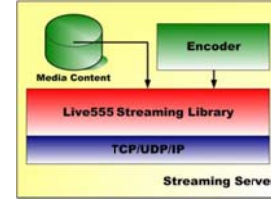


FIGURE IV. LIVE555 FRAMEWORK.

### V. RESEARCH METHOD

In the network, a shared environment, the traffic from the video of H.264 source end, either in VBR or CBR mode, is interfered by other and thus the fluency and clarity of video frames are unpredictable in receiver end. Therefore, the source rate of the video must be adjusted to meet the available bandwidth to maintain the immediateness of the video; however, even immediate condition is met, it must achieve effectiveness. That is, transmit the clearest frame within the limited bandwidth.

When the user starts the play program and traffic control program, the play program will send the RTSP packets to LIVE555 server to establish the streaming passageway between each other. Also, it will start executing the traffic control program to calculate the UDP packet size comes from the server in one cycle. This value is the current actual traffic. In addition to using this bandwidth to predict the available bandwidth of the next cycle, the control program also feeds back this rate to the server as the preprogrammed source rate. This data reflects the predicted bandwidth of the video gained on the next cycle. When the traffic control program run by the ARM core receives the feedback value, it will provide the source rate to the DSP core to have H.264 encoder dynamically adjust the internal parameter of video coding layer (VCL) bit rate and pack it as the RTP packet to transmit to the client to make the data output rate after compression achieve the preprogrammed source rate. Denote the actual video traffic measured by the client on the  $n$  cycle by  $B_t(n)$  and the preprogrammed source rate of the same cycle by  $B_p(n)$  to predict the source rate of the next cycle, the  $n+1$  cycle, denoted by  $B_p(n+1)$ . The equation is shown in Eq. (1). The calculated predicted traffic value of this equation is related to the history actual traffic value [6]. The proportion is related to parameter  $\alpha$ . Finally, this predicted traffic value will make H.264 encoder to dynamically adjust the quantization parameter to accommodate the actual bandwidth available.

$$B_p(n+1) = \alpha B_t(n) + (1 - \alpha) B_p(n), 0 < \alpha < 1 \quad \dots \text{Eq. (1)}$$

$$\begin{aligned}
B_p(n+1) &= \alpha B_t(n) \\
&+ (1-\alpha)[\alpha B_t(n-1) \\
&+ (1-\alpha)B_p(n-1)] \\
B_p(n+1) &= \alpha B_t(n) + \alpha(1-\alpha)B_t(n-1) \\
&+ (1-\alpha)^2 B_p(n-1) \\
B_p(n+1) &= \alpha B_t(n) + \alpha(1-\alpha)B_t(n-1) \\
&+ \alpha(1-\alpha)^2 B_t(n-2) \\
&+ \alpha(1-\alpha)^3 B_t(n-3) \\
&+ \alpha(1-\alpha)^4 B_t(n-4) + \dots
\end{aligned}$$

The  $B_p(n+1)$  of Eq. (1) is the relation developed from the viewpoint of network congestion. It is the case that the actual traffic  $B_t(n)$  is much smaller than the predicted traffic  $B_p(n)$ . If the actual traffic has small gap with the predicted traffic, the network traffic is determined as smooth. When the network is smooth,  $B_p(n+1)$  must capture the residue bandwidth to normalize the video quality; in theory,  $B_p(n)$  and  $B_t(n)$  shall be in consistence when the network is smooth. In practice, it will have some deviation. If the deviation is smaller than the constant  $\varepsilon$ , they are deemed as in consistence. At this time, the predicted  $B_p(n+1)$  must add one addition proportion  $\delta$  to capture more available bandwidth. Therefore, after calculating the  $B_p(n+1)$  of Eq. (1), Eq. (2) must be calculated to determine whether  $B_p(n+1)$  need to be added.

$$\begin{aligned}
B_p(n+1) &= B_p(n+1)(1+\delta), \text{ if } (B_p(n) - B_t(n))/B_p(n) \\
&\leq \varepsilon \dots \text{Eq. (2)} \\
0 < \delta < 1; 0 < \varepsilon < 1
\end{aligned}$$

## VI. SYSTEM ARCHITECTURE AND IMPLEMENTATION

This study uses TI DaVinci TMS320DM6446 dual core platform to setup a network surveillance system, providing the users to remotely view real-time and effective video frames. System framework is shown in Figure 5. The operational procedure in the DM6446 platform is divided into two parts. One is that DSP core compresses the video frames to H.264 format and transmits the frame bit to ARM core. The other is that ARM core executes the operation system to run the LIVE555 server [7] and traffic control program for forming the DM6446 server end.

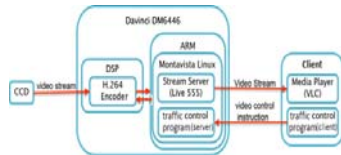


FIGURE V. SYSTEM FRAMEWORK.

In the DM6446 platform, while the H.264 Encoder is run on the DSP core, the operating system, MontaVistaLinux, is run on the ARM core to execute two programs: LIVE555 and traffic control program (server). LIVE555 is a video streaming program. When the client starts the VLC media player and connect to LIVE555 streaming server through RTSP URL, LIVE555 will pack the video frames compressed by H.264 Encoder as RTP packets and transmit to the media player at the client end, as shown in Figure 6. The traffic control program(server)receives the feedback from the traffic control program (client) to request the H.264 Encoder to adjust its

output rate. In the Client end, the traffic control program (client) captures RTP packets to calculate the received flow of one single cycle, and through Eq. (1) and (2) to predict the flow rate of the next cycle. This flow rate value will be fed back to the server end. The traffic control program (server) will receive the predicted flow rate value from the client end as the next-cycle programmed source rate. Once the predicted value is received, it will request the H.264 Encoder to adjust the out put bit rate by re-compressing the frames to fit the available band width. This adjustment processing information is shown in Figure 7.



FIGURE VI. VLC PLAYING SURVEILLANCE SCREEN.

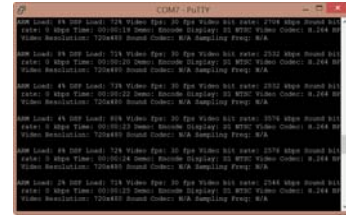


FIGURE VII. H.264 ENCODE COMPRESSED INFORMATION.

## VII. EXPERIMENT RESULTS

This study focuses on traffic control. The passageway bandwidth is initially set at 2Mbps and adjusted to 1.5Mbps after 1 minute and then to 1Mbps after another 1 minute. Therefore the RTP packets of the video stream received by the client end reduce. In order to allow the H.264 Encoder program to adjust the compressing rate immediately, the predicted cycle of the video stream traffic at the client end is 5 seconds, and the default bit rate of the H.264 Encoder program and predicted traffic amount are 2Mbps. When the bandwidth of the passageway becomes 1.5Mbps, the traffic control program executed at the client end will capture the RTP packets transmitted from the server to get the actual average flow rate as 1.406Mbps. Through Eq. (1), it calculates the predicted traffic is about 1.5Mbps and return this value to the server end, which will adjust the bit rate to 1.5Mbps. However, the video traffic does not meet immediately the available bandwidth, and 5 packets are lost. If the bit rate is not adjusted and keeps transmitting packets in 2Mbps, in theory, 0.5Mbps traffic will be lost. 312.5Kbytes are loss in 5 seconds, divided by 1100bytes of one RTP packet in average, 284 packets will be loss. Therefore, by dynamically adjusting the video source rate, the packets might not affected by the changing of the network traffic and transmit to the client normally to play the frame smoothly.

TABLE I .THE RELATIONSHIP OF FLOW AND RTP PACKETS.

Channel Bandwidth(bps)	1M	1.5M	2M
Source address	192.168.1.74	192.168.1.74	192.168.1.74
Source Port	6970	6970	6970
Destination address	192.168.1.123	192.168.1.123	192.168.1.123
Destination Port	54092	49520	62118
RTP Packets	659	930	1202
Lost Packets	1	5	1

### VIII. SUMMARY

The change in network traffic affects the transmission of the video packets; therefore the main feature of this system is that the client end will feedback the predicted video flow rate in the next cycle to the server within a certain period. According to the client's feedback, the server will dynamically request H.264 Encoder to adjust its output bit rate to fit the available band width. The rate-adaptive traffic control approach has been implemented on TI DaVinci dual core platform and gives a practical experience for dynamically controlling the video surveillance traffic by tracking the available network bandwidth. The future study will research the length of the system feedback cycle to make optimization analysis on the rate-adaptive process. It is clear that a large feedback cycle will make the H.264 Encoder not able to react to the change of available network bandwidth speedily while a small cycle will make it adjust its output bit rate frequently and cause a traffic oscillation.

### ACKNOWLEDGEMENT

This work was supported in part by R.O.C. (Taiwan) Ministry of Science and Technology under Grant MOST103-2815-C-131-005-E.

### REFERENCES

- [1] Koumaras H., Skianis C., Gardikis G., KourtisA., "Analysis of H.264 video encoded traffic", INC 2005 Fifth International Network Conference, Samos Island, Greece, July 2005.
- [2] Patrick Seeling and Martin Reisslein, "Video Transport Evaluation With H.264 Video Traces", IEEE Communications Surveys& Tutorials (2012) 1142-1165.
- [3] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," IEEE Trans. Circuits Syst. Video Technol. (2003) 560-576.
- [4] Introduction of Real-time Streaming Protocol on [http://fanli7.net/a/bianchengyuyan/\\_NET/20120615/171925.html](http://fanli7.net/a/bianchengyuyan/_NET/20120615/171925.html)
- [5] RTSP(Real Time Streaming Protocol) on <http://b8807053.pixnet.net/blog/post/3611245-rtsp%E4%BC%88real-time-streaming-protocol%E4%BC%89>
- [6] You Jun Chung, JongWon Kim, and C.-C. Jay Kuo, "Real-Time Streaming Video With Adaptive Bandwidth Control and DCT-Based Error Concealment", IEEE Transactions on circuits and systems-II: analog and digital signal processing, (1999).
- [7] LIVE555.COMTM on <http://www.LIVE555.com/>