Design and Simulation of a H.264 AVC Video Streaming Model

Doggen, Jeroen
Van der Schueren, Filip

January 31, 2008

"University College of Antwerp", Department of Applied Engineering
Paardenmarkt 92, B–2000 Antwerpen, Belgium
j.doggen@ha.be

Abstract

This paper explains the design process of a traffic simulation model for H.264 AVC video streaming. The simulation model for H.264 video streaming was developed using OPNET Modeler, an advanced network modeling and simulation tool. The gamma distributions used in the model are based on empiric mean and variance values. The stream generator was used in several simulation scenarios in conjunction with Ethernet and wireless LAN node and network models. The simulation results show that based on the high level characteristics in the time domain, a H.264 stream is very similar to a MPEG2 stream. Under stressed network conditions, the more advanced H.264 standard shows better results: lower queuing delays and less packet-loss.
Keywords: *H.264 AVC, video streaming, simulation, OPNET Modeler*

1 Introduction

The introduction of DVD-Video and the arrival of digital television have revolutionized the world of home entertainment and broadcast television. These applications and many more were made possible by the standardization of video compression technologies. New standards are currently enabling a new generation of Internet based applications. H.264 is one of the most promising standards to be used in a very wide field of applications, ranging from low-bandwidth / low-resolution cellular phones to High Definition cinema systems.
When deploying new video streaming implementations it is very important to test different design options. Using a network traffic model it is possible

to simulate and evaluate all design choices prior to performing real-world tests. The H.264 model, designed using OPNET Modeler [1], was based on some of the theoretical concepts used in an existing MPEG2 model. By adding new features to this model it was possible to send H.264 streams over a broad range of networks, using different underlying protocols. The model was tested in an Ethernet, token-ring and a Wireless LAN environment. The simulation results show that based on the high level characteristics in the time domain, a H.264 stream is very similar to a MPEG2 stream. This behavior was expected because H.264 is essentially an improved version of the MPEG2 standard. Depending on the end application, the parameters used by the model can be changed to test specific network scenarios.

2 Limitations of network simulation

The analysis of modern telecommunication systems can be extremely complex, as most standard modeling techniques analyze each component and do not necessarily take into account the relationships that exist between the components within the system. Simulation is an approach which can be used to predict the behavior and performance of large, complex stochastic systems [2]. The development of an accurate simulation model requires extensive resources. When a model is not very accurate, one can make the wrong conclusions from the simulation results. The basic problem is that every simulation model is inherently wrong, ranging from lightly flawed up to totally wrong. As a result the simulation outcome is only as good as the model and it is still only an estimate of a possible projected outcome.

3 MPEG2 Model

The original OPNET simulation model for MPEG2 streaming was developed in the year 2000 by Srinivas Kandala and Sachin Deshpande[3], who were both working at Sharp Laboratories of America at that time. This model is available freely for maintained OPNET customers and University program users.

3.1 Theoretical concepts

The model itself is based on a traffic model developed by M. Krunz, and H. Hughes[4]. They analyzed several test streams and measured the number and size of the three different types of frames (I,P and B). They observed that the correlations between their measured statistics were very complex because of the fact that one stream holds three types of frames with varying sizes. By decomposing the stream into three separate streams, each holding just one type of frame, an accurate approximation of the streams could be modeled. The following three variables determine the nature of the stream: scene length distribution, frame size distribution and stream structure.

3.1.1 Scene length distribution

A scene is one part of a movie filmed using just one shot. They observed that a sudden change in I-frame size was a way to detect the start of a new scene. In their paper they show several figures of the probability density distribution of the scene length. Apparently, in 95% of the tests the lenght of one scene is not dependent on the length of other scenes. The scene length is modeled as a sequence of *iid* random variables with a geometric distribution.

3.1.2 Frame size distribution

The frame size distribution of the complete stream (with I,P and B-frames) was not studied thoroughly because the impact of the frame type is an essential aspect of the model. They examined three different probability density functions: Gamma, Weibull and Lognormal distribution. The conclusion was that a lognormal distribution provided the best fit for the frame size histograms of the three streams. Probability density function of the lognormal distribution:

$$F(x;\mu,\sigma) = \begin{cases} \frac{1}{x\sigma\sqrt{2\pi}} e^{-(\ln x-\mu)^2/2\sigma^2} & x > 0 \\ 0 & otherwise \end{cases}$$

With:

$e^{\mu+\sigma^2/2}$: the expected value

$(e^{\sigma^2} - 1)e^{2\mu+\sigma^2}$: the variance

3.1.3 Stream structure

An example stream structure can be seen in Figure 1. At the lowest level, we have 15 frames, they form one Group of Pictures (GOP). Several of these GOPs form one scene. This means the scene length equals N*d. (with N: the number of frames in one GOP and d: the number of GOPs in one scene)

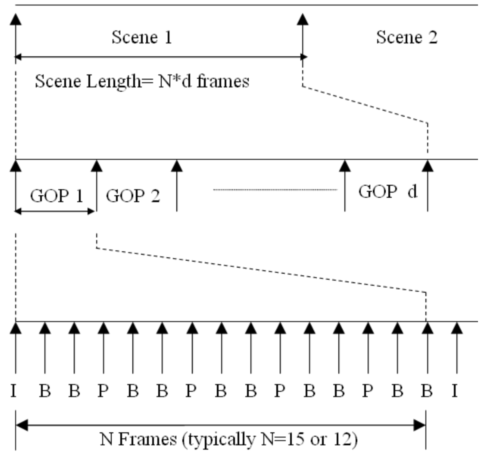

Figure 1: Structure of the modeled MPEG2 stream

3.2 Node model design

The first test scenario was a completely custom made simulation design. The design includes: a packet format, a generic node model to support transmitting and receiving packets, a link model and the MPEG2 process model. The node model consists of the stream generator, a sink module and two transceivers. The only application supported is MPEG2 streaming, no real world links nor protocols can be used. This basic node model was used to test the key functionality of the MPEG2 & H.264 models without having any problems with the configuration and limitations of other protocols.

4 MPEG4 Part 10 - H.264 Model

The global structure of a H.264 video stream is similar to a MPEG2 video stream, but there are some major differences: the frame size distribution and the stream structure. One new feature specific to H.264 is the use of Si and Sp frames. Further, H.264 has an sophisticated rate control algorithm to produce constant bitrate streams (CBR). The H.264 codec can approximate a target bitrate by changing the quantizer parameter (QP). When CBR is used, the process model in OPNET is very simple, just use the standard *simple_source* module and set the frame size and frame interarrival time to the desired values. When a variable bitrate (VBR) stream with a fixed QP is used, things get more interesting. For some environments CBR can be very efficient (i.e. connection oriented networks: ATM, Frame relay,...), but VBR is much more efficient in packet switched networks. When a VBR stream is used, scenes with a high entropy get more bandwidth, resulting in a higher visual quality. Because different streams tend to have high entropy scenes at different moments in time, the bitrate is divided very efficiently when several streams are transmitted over a packet switched network. All the following information is based on VBR streams.

4.1 Theoretical concepts

The scene length is independent of the video codec and can be modeled as a sequence of *iid* random variables with a geometric distribution.

4.1.1 Frame size distribution

The simulation model for H.264 streaming is based on a traffic model that was developed by H. Koumaras, C. Skianis, G. Gardikis and A. Kourtis [5]. The methods they used are the same as the methods used to develop the MPEG2 traffic model. They fitted a gamma distribution to the histogram of the frame sizes of their test data. These empiric mean and variance values are used in the simulation models. Probability density function of the gamma distribution:

$$F(x;k,\theta) = \begin{cases} x^{k-1}\frac{e^{-x/\theta}}{\theta^k\,\Gamma(k)} & x > 0 \\ 0 & otherwise \end{cases}$$

With:

$k\theta \ = M$: Expected Value (Mean)

$k\theta^2 \ = V$: Variance

$V/M = \theta$: Scale Factor

$M^2/V = k$: Shape Factor

The model only provides mean and variance values for just one resolution (528x384), using linear extrapolation, the corresponding values for SDTV and HDTV resolutions can be calculated. With this new values, the necessary scale and shape factors can be found. These factors are used in the OPNET simulations. Table I shows some of the calculated values. These values are all calculated using the mean and variance values found by H. Koumaras and his colleagues after analyzing a video sequence encoded with a QP of 30. The I,B and P mean and variance values are the values per frame in bits. Depending on the number of frames per second (fps) and the GOP this results in a different total bitrate. The mean bitrate

		M (10^3)	V (10^9)	θ	k (10^3)
525*384	I	53,91	659,6	4,406	12,24
	B	7,860	75,90	0,814	9,660
	P	16,33	194,0	1,374	11,88
Mean Bitrate:	438 Kbps				
720*576	I	110,3	1,350	9,013	12,24
	B	16,08	0,155	1,665	9,660
	P	33,40	3,968	2,812	11,88
Mean Bitrate:	896 Kbps				
1280*720	I	245,1	34,64	20,03	12,24
	B	35,73	15,12	3,700	9,660
	P	74,23	19,61	6,248	11,88
Mean Bitrate:	1990Kbps				
1920*1080	I	551,3	77,94	45,06	12,24
	B	80,39	34,01	8,325	9,660
	P	167,0	44,11	14,06	11,88
Mean Bitrate:	4480Kbps				

Table 1: H.264 parameters (Mean & Variance in bits)

mentioned in the table is calculated using a frame rate of 30fps and the following GOP:'IPBPBPBPBPBPB'.

4.1.2 Stream structure

The structure of a simple H.264 stream is similar to an MPEG2 stream. More complex H.264 inplementations use other GOPs because differing applications demand different GOPs. Real-time applications use a very short GOP with I and P frames (*IPI* or *IPPI*), because of time and processing power constraints. Very complex multi-pass algorithms and much larger GOPs can be used for offline encoded content. I, P, B, Si and Sp frames can be used. With the help of a simple program, written in plain C code the generation process of different GOPs was tested. This was the easiest and fastest way to test new GOPs. By replacing the procedure used by OPNET to generate a packet with a simple *printf* command and by replacing OPNETs looping process with a *for* loop, the code could be interchanged between OPNET and Visual Studio without too much problems. Extra code to support the following GOPs was added: *IBBPBBP..PBB*, *IP*, *IP...P*, *IPBPB..PB*, *..PBBPBBIBB* and *ISiSpPBB..B*. Their typical use (in order):

	Calculated BR	Target BR	File size(KB)	final BR
525*384	438	432	6465	544
720*576	896	864	11041	930
1280*720	1991	2016	25048	2109

Table 2: H.264 test sequences: all bit rates (BR) in Kbps

MPEG2 display order, real-time H.264, offline encoded H.264 content, offline encoded H.264 content using B frames, real MPEG2 transmission order and H.264 extended profile. GOP properties can be configured using two parameters: NGOP and MGOP. To get this code converted back to OPNET, the *printf* commands have to be replaced by the appropriate command to schedule and generate a packet. (i.e. *next_pk_evh = op_intrpt_schedule_self (op_sim_time() + next_intarr_time,SSC_IGENERATE);*)

4.2 Real-world tests

By encoding several video sequences at bit rates approximating the values calculated in Section 4.1.1, the credibility of these values was evaluated. Table II shows an overview of calculated bit rates, target bit rates, file sizes and final bit rates. The source video for these tests was the first chapter from the high definition version of the movie 'Constantine'. The encoding of the video sequences was done using the opensource x264 encoder[8]. Properties of the video sequence: 2853 frames, 29,95 frames per second, 95,25 seconds, audio disabled, H.264 main profile level 2 (level 4 for HDTV resolutions). These rather low values were chosen based on the calculated values and on bitrate values that are popular in the "video-copying-community". The output files were shown to several test persons. Based upon the subjective opinion of these test persons, the quality at these bit rates was found adequate.

4.3 Process model design

The process model has three different states (init, generate and stop). Figure 2 gives a graphical overview of the transitions between the different states. There are eight different conditional transitions and eight corresponding actions. The state variables and the temporary variables are declared outside of the finite state machine (FSM) diagram. All the interrupt codes and header definitions are defined in the header block. The code used to generate packets, calculate packet sizes and generate statistics resides in the function blocks.

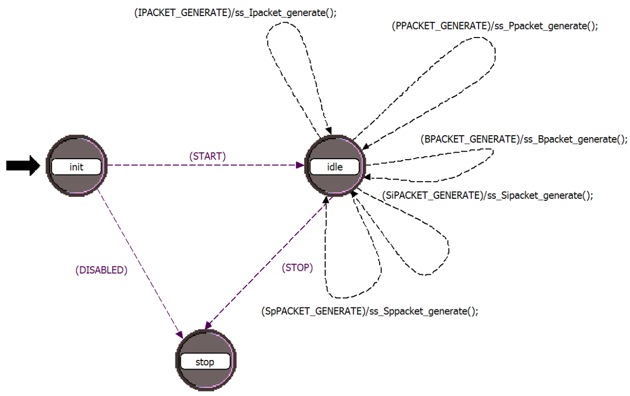

Figure 2: H.264 process model

4.3.1 The init state

In the *init* state, all the values that are selected at the node level are read. Some other variables are registered and some low level checks are done to ensure correct simulation. In case of a '*start*' event, the FSM switches to the idle state.

4.3.2 The idle state

The arrival time of the next packet is added to the event queue. The GOP type is determined, an algorithm determines the next packet type depending on past events. When all the frames of a scene are sent, a new scene length is calculated. The idle state has a transition for six different interrupts, one for each packet type and the '*stop*' interrupt. When a '*packet_generate*' interrupt arrives, the corresponding function in the function block is activated.

Si and Sp frame support Si and Sp frames are used to switch between streams in the H.264 extended profile. Because Si and Sp frames are a very new concept, and their exact implementation is still being discussed, there are no research papers available about possible traffic models for the Si and Sp frames. When new traffic models become available, they can be applied to this model without any problems.

4.3.3 The stop state

No more traffic is generated by this module until the simulator stops.

4.4 Node model design

The custom made node model (Section 3.2) was used to test the basics of the H.264 process model. For more complex scenarios the 'Adapted Ethernet station' node model was used.

4.4.1 Adapted Ethernet station

The *Ethernet_station_adv* node module contains four process modules, one of them is called *bursty_gen*. By replacing the *bursty_gen* module with the H.264 module it is possible to send H.264 packets directly into the data link layer. This method completely ignores the Real-time Transport Protocol (RTP), User Datagram Protocol (UDP) and Internet Protocol (IP) typically found in a real-world implementation.

Packet segmentation The frames generated by the H.264 generator are very big, as a result, when using this module directly over Ethernet, all packets larger than Ethernet's 1500 bytes maximum tranfer unit (MTU) get dropped. Packet segmentation was added to the model to solve this problem. Instead of sending the frame direcly to the Ethernet layer, the frame is first placed in a buffer. This buffer is emptied in small steps, with a step size equal to the MTU. These small chunks of data are send to the lower layer.

Protocol overhead Because the packets are injected directly into the data link layer, the packet-header overhead will be lower than in reality. The typical overhead in this implementation is 26 bytes for every 1500 bytes send (98,2% efficiency). The minimal overhead in a Network Adaptation Layer (NAL), RDP, UDP, IP over Ethernet implementation is 71 bytes (1+12+12+20+26) for every 1500 bytes send (95,5% efficiency). This is the best case scenario, in practical implementations, the overhead will be much larger. By adding an extra 52 bytes field to the Ethernet packet, it would be possible to lower the efficiency to have more realistic results. This was not done because it does not really make sense. Enhancing the generator in the future to use RTP,UDP and IP is more interesting.

4.4.2 Audio streaming

To complete the node model, a slightly adjusted *bursty_gen* model was added. It can be configured to generate typical audio streams: 128 Kbps, 192 Kbps & 448Kbps.

5 Simulation scenarios

The proces model was tested using several network models and simulation scenarios created to simulate different network environments.

5.1 Switched Ethernet

To test a switched Ethernet network, a network with four servers, three switches and seven clients was configured. All servers can send one H.264 stream. Figure 3 shows the network layout. The links between the servers and the switches are 100Mbps Ethernet links (thin arrows), the links between switch3 and the clients are 10Mbps links (thick arrows). The four servers all send a H.264 stream using an Ethernet broadcast. They all have different start and stop times. It was made sure that at some time all the servers were sending data. Figure 4 shows that the 100 Mbps links have no problems with the load, while the 10 Mbps links get saturated when all servers are active. At that moment queue overflows result in increased Ethernet delay and packet loss. Transmission of the same video essence with a lower compression ratio using the MPEG2 codec results in higher link utilization and earlier link saturation.

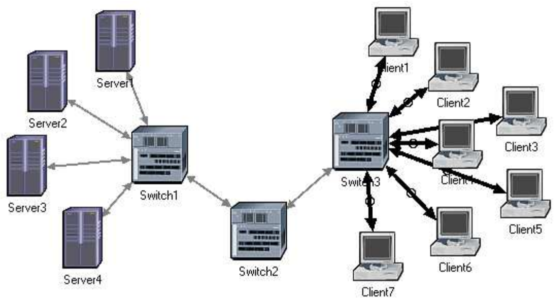

Figure 3: Switched Ethernet test scenario: Network layout

5.2 Wireless LAN

OPNET provides several Wireless LAN models. The *wlan_station_adv* module is very similar to the *eth_station_adv* Ethernet module. The H.264 model works without any problems in all the tested environments. The model was tested using a basic WLAN scenario, with a direct sequence

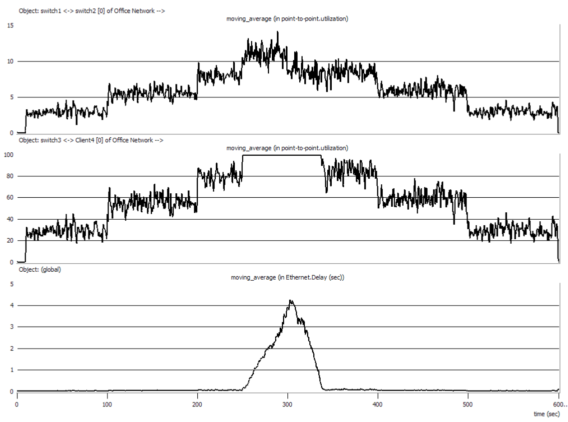

Figure 4: Switched Ethernet test scenario: link utilization and Ethernet delay

11Mbps link, one sending station (using broadcast), and one receiving station. All simulation results were as expected, similar to the Ethernet simulation results.

5.3 SDTV & HDTV

Using the parameters shown in Table I, some tests were done to compare several streams generated by the process model. Figure 5 shows the frame size histogram of two simulations. On the left side the B-frames of a HDTV (1920*1080) stream are shown, on the right side the histogram of the I frames of a SDTV stream (720*576). The simulated values correspond to the calculated values. The simulation results show that, when doing a time domain comparison of a H.264 stream and a MPEG2 stream, using comparable bit rates both streams are very similar. Depending on the end application, more parameters can be calculated to test the capacity of any Ethernet or Wireless LAN network.

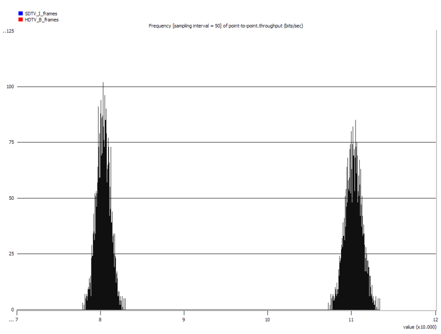

Figure 5: H.264 frame size histogram

6 Conclusion

The design process of a traffic simulation model for H.264 AVC video streaming was explained in this paper. OPNET Modeler was used to program the low level process models. A model of a MPEG2 stream generator was explained first, because the H.264 model is based on similar theoretical concepts. The H.264 model uses an other traffic model and has some extra features: Si/Sp frame support and packet segmentation. The packet segmentation support allows the model to be used over several underlying protocols. The model was tested in an Ethernet and a Wireless LAN environment. The simulation results show that based on the high level characteristics in the time domain, the behavior of H.264 and a MPEG 2 stream are very similar. Depending on the end application, several parameters can configured to simulate a broad range of practical applications.

References

[1] OPNET Modeler - OPNET Technologies, Inc. http://www.opnet.com

[2] Flood, J.E. (1998) "Telecommunications Switching, Traffic and Networks", New York: Prentice-Hall, 1998

[3] S. Kandala and S. Deshpande (2000) "Models for MPEG2". Sharp Laboratories of America

[4] Marwan Krunz and Herman Hughes (1995) "A traffic model for MPEG-coded VBR streams". in Proceedings of the ACM SIGMETRICS '95 Conference pp 47-55.

[5] H. Koumaras and C. Skianis and G. Gardikis and A. Kourtis (2005) "Analysis of H.264 video encoded traffic". in Proceedings of the 5th International Network Conference (INC2005) pp 441-448

[6] M. Krunz, S. K. Tripathii (1997) "On the characterization of VBR MPEG streams". in Proceedings of ACM SIGMETRICS97 Conference pp 92-20

[7] A. Matrawy, L. Lambadaris, Changcheng Huang (2002) "MPEG4 Traffic Modeling Using The Transform Expand Sample Methodologys". in Proceedings of the IEEE 4th International Workshop on Network Appliances pp 249 - 256

[8] x264 - a free h264/avc encoder. http://www.videolan.org/x264.html