MPEG4 Part 10 - H.264 Modeling in OPNET

Doggen Jeroen [1], Van der Schueren Filip [2]
[1]doggen.jeroen@student.ha.be
[2]F.VanderSchueren@ha.be

June 12, 2006

***Abstract*—This paper explains the design process of a traffic simulation model for MPEG4 Part 10 / H.264 AVC video streaming. H.264 is one of the most promising standards to be used in a very wide field of telecommunication and entertainment applications. These applications range from low-bandwidth/low-resolution cellular phones to high definition cinema systems. The simulation model for H.264 video streaming was designed using OPNET Modeler, an advanced network modeling and simulation tool. The gamma distributions used in the model are based upon 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. Depending on the end application, the parameters to configure the stream can be changed.**

I. 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, 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 using different underlying protocols.

The model was tested in an Ethernet and in 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. Depending on the end application, the parameters used by the model can be changed.

II. NETWORK SIMULATIONS

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 model large, complex random systems to make educated predictions or for performance measurement purposes.

The development of an accurate simulation model requires extensive resources. When a model is not very accurate, one can take the wrong conclusions out of the simulation results. The basic problem is that every simulation model is 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.

III. OPNET MODELER

OPNET technologies[1] provides one of the most advanced environments for network modeling and simulation. It is used to accelerate research and development. Modeler has been a very useful tool during the development of wired and wireless networks, devices and communication protocols. Modeler uses an object-oriented modeling approach, it consists of a series of hierarchical editors that directly parallel the structure of real networks, equipment and protocols. On the lowest level, a finite state machine approach is used to mirror the functionality of the real-world devices.

IV. MPEG2 MODEL

The OPNET simulation model for MPEG2 streaming was developed in the year 2000 by Srinivas Kandala and Sachin Deshpande[2], 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.

A. Theoretical concepts

The model is based upon a traffic model that was developed by M. Krunz, and H. Hughes[3]. They analyzed several test streams. The number and size of the three different types of frames (I,P and B) were measured. 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. They decomposed the stream into three separate streams, each holding just one type of frame. Then they developed a traffic model for each stream. The following three variables determine the nature of the stream: scene length distribution, frame size distribution and stream structure.

1) Scene length distribution: A scene is one part of the movie that is filmed using one camera angle. 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 all cases 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.

2) Frame size distribution: The frame size distribution of the complete stream (with I,P and B-frames) was not studied thoroughly by Krunz and Hughes because they believed that 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) Stream structure: An example stream structure can be seen in Figure 1. At the lowest level, we have 15 frames, they form one 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)

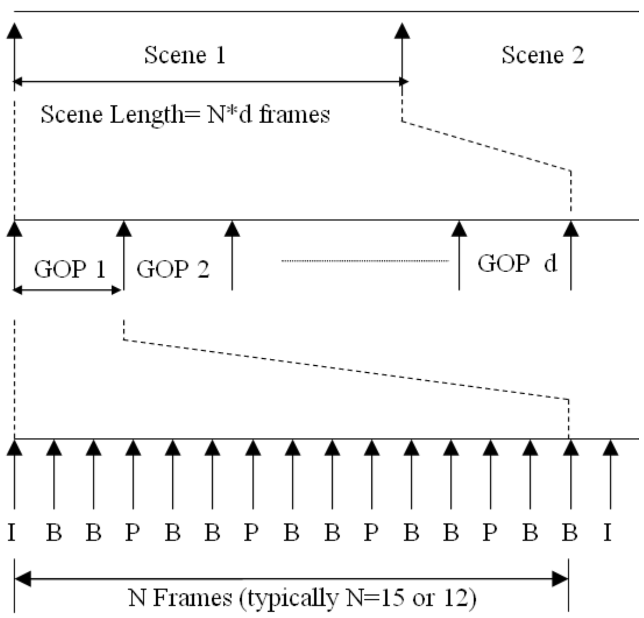

Fig. 1: Structure of the modeled MPEG2 stream

B. 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 is very useful to test the key functionality of the MPEG2 model without having any problems with the configuration and limitations of other protocols.

V. MPEG4 PART 10 - H.264 MODEL

The high level 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.
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 applications CBR can be very efficient (i.e. connection oriented environments: 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. This results 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 one packet based link. All following information is based upon VBR streams.

A. Theoretical concepts

1) Frame size distribution: The simulation model for H.264 streaming is based upon a traffic model that was developed by H. Koumaras, C. Skianis, G. Gardikis and A. Kourtis [4]. 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. Their 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 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 by H. Koumaras and his colleagues after analyzing a video sequence encoded with a QP of 30. The I,B and B 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 mentioned in the table is calculated using a frame rate of 30fps and the following GOP:'IPBPBPBPBPBPB'.

2) Stream structure: The higher level structure of an H.264 stream is identical to an MPEG2 stream. The variable part is the GOP. Different applications demand different GOP's. Real-time applications use a very short GOP with I and P frames (IPI or IPPI), because of the limited time and processing power available. Very complex multi-pass algorithms and much larger GOP's can be used for offline encoded content. I,P and B frames can be used.
With the help of a program, written in Microsoft Visual studio using the C programming language, the generation process of different GOP's was tested. This was the easiest and fastest way to test new GOP's. By replacing the procedure used by OPNET to generate a packet with a simple *printf* command and by replacing OPNET's looping process with a for loop, the code could be interchanged between OPNET and Visual Studio. Extra code to support the following GOP's was added: IBBPBBP..PBB, IP, IP...P, IPBPB..PB, ISiSpPBB..B and ..PBBPBBIBB. Their purpose (in order): MPEG2 display order, real-time H.264, offline encoded H.264 content, offline encoded H.264 content, H.264 extended profile and the real transmission order used by MPEG2. GOP properties can be changed 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);*)

B. Real world tests

By encoding several video sequences at bit rates approximating the calculated values (Table I), the feasibility 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 super© encoder from eRightSoft[5]. Super© is a graphical user interface for the x264 command line encoder[6]. 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 1280*720 resolution).
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.

C. 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 conditional transitions and eight corresponding conditions. 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 block.

1) The init state: In the initial 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*', the FSM switches to the idle state.

2) The idle state: The arrival of the next packet is scheduled. 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.

		M (10^3)	V (10^9)	θ	k (10^3)
525*384	I	53,91	659,6	4,406	12,24
	B	7,86	75,90	0,814	9,660
	P	16,33	194,0	1,374	11,88
Mean Bitrate:	438 Kbps				
720*576	I	110,27	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 I: H.264 parameters (Mean & Variance in bits)

	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 II: H.264 test sequences: all bit rates(BR) in kbps

3) The stop state: All packet generation is canceled. No more traffic is generated by this module until the simulator stops.

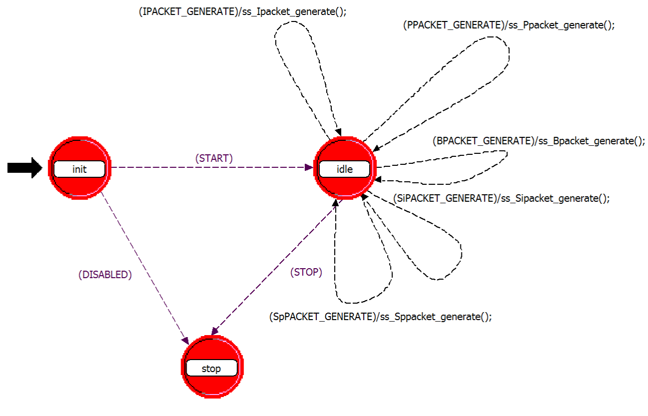

Fig. 2: H.264 process model

4) Si and Sp frame support: Si and Sp frames are used to switch between streams in the H.264 extended profile. By adding extra code to the process model, these frames could be supported. Because Si and Sp frames are a very new concept, and their actual 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.

D. Node model design

The custom made node model was used to test the basics of the H.264 process model. For more complex scenarios the 'adapted ethernet station' node model was used.

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 datalink layer. This method completely ignores the RTP, UDP and IP protocols typically found in a real-world implementation.

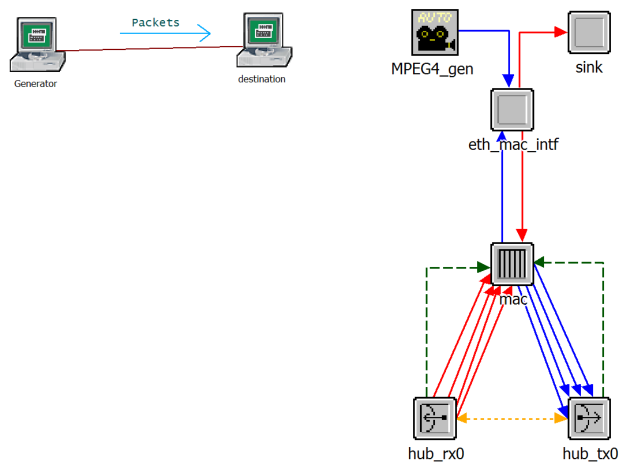

Fig. 3: MPEG test scenario: node and network model

a) Packet segmentation: By adding an extra option to the process model, some problems concerning the Ethernet maximum transfer unit were solved. The packets generated by the H.264 generator are very big, but when using this module directly over ethernet, no packet can be bigger than the MTU (usually 1500 bytes), all larger packets are dropped by the Ethernet MAC layer.
The steps needed to implement packet segmentation: add the *segmentation size* model attribute, add state variables: *segmentation size* and *segmentation_buf_handle*, add init enter execs code, change the code in the frame generation functions.
The frame is placed in a buffer, the buffer is emptied in small steps, with a step size equal to the *segmentation size*. These small chunks of data are send to the lower layer.

b) Packet header overhead: Because the packets are injected directly into the datalink layer, the packet-header overhead will be lower than in reality. The typical overhead in this implementation is 208 bits for every 1500 bytes send (98,2% efficiency). The typical overhead in an NAL, RTP, UDP, IP over ethernet implemention is 71 bytes (1+12+12+20+26) for every 1500 bytes send (95,5% efficiency). This is the best case scenario, when the segmentation size is set to a lower value, the overhead will be much larger. By adding an extra 52 bytes field to the ethernet packet, it is possible to lower the efficiency to have more realistic results. This was not done because it does not really make sense. Enhancing the generator to use RTP,UDP and IP is more interesting.

2) Adapted ethernet workstation: The *ethernet_wkstn_adv* node module contains eleven process modules. The complete TPAL, TCP, IP, ethernet protocol stack is supported. If the application module of such a workstation could be changed, it would be possible to send an H.264 stream over any network supported by OPNET. When this is working, (especially the connection between the application and TPAL process: ICI interfaces) it will be relatively easy to change this node model to support other underlaying protocols (such as ATM, frame relay). Some adjustments have been made to the model, but because of time constraints, this job was not finished completely. The following simulation scenarios are done using the 'adapted ethernet station' node model.

VI. SIMULATION SCENARIOS

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

A. Wireless LAN

OPNET provides several Wireless LAN modules. 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 11Mbps link, one sending station (using broadcast), and one receiving station. All simulation results were as expected, similar to the ethernet simulation results.

B. Switched ethernet

To test a switched ethernet network, a network with four servers, three switches and seven clients was build. All servers can send one H.264 stream. Figure 4 shows the network layout. The links between the servers and the switches are 100Mbit ethernet links (red), the links between switch3 and the clients are 10Mbit links (blue).
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 5 shows that the 100 Mbit links have no problems with the load, but the 10 Mbit links get saturated when all servers are active, this results in increased ethernet delay.

C. SDTV & HDTV

Using the parameters shown in Table I, some tests were done to compare several streams generated by the process model. Figure 6 shows the frame size histogram of two simulations. On the left side the histogram of 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, coaxial ethernet or Wireless LAN network.

VII. CONCLUSION

The design process of a traffic simulation model for MPEG4 Part 10 / H.264 AVC video streaming was explained in this paper. OPNET Modeler and Microsoft Visual studio were 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

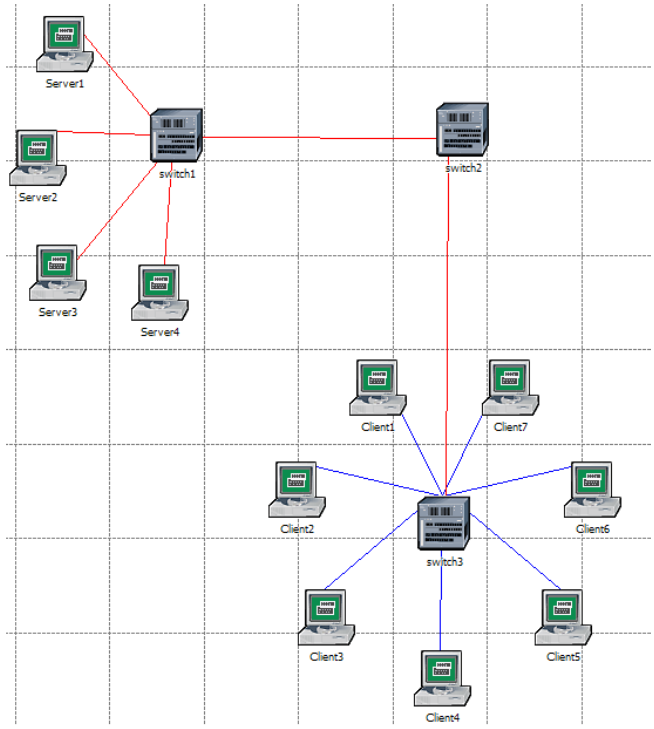

Fig. 4: Switched ethernet test scenario: Network layout

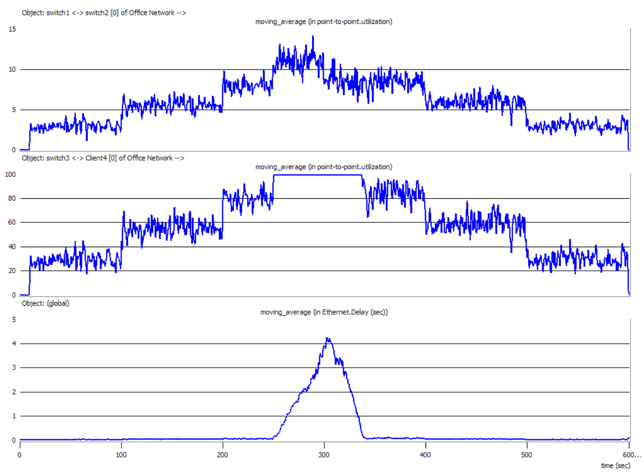

Fig. 5: Switched ethernet test scenario: link utilization and ethernet delay

segmentation. The packet segmentation support allows the model to be used over several underlying protocols. The model was tested in an Ethernet and in 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. Depending on the end application, the parameters to configure the stream can be changed.

VIII. Acknowledgments

The authors gratefully acknowledge the support of Jaume Abella and his colleagues at Universitat Ramon LLull, La Salle.

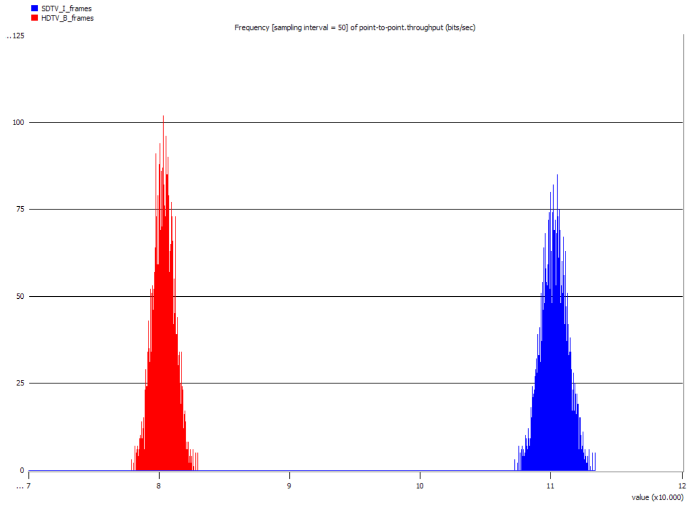

Fig. 6: H.264 frame size histogram

References

[1] "Opnet modeler," OPNET Technologies, Inc. [Online]. Available: http://www.opnet.com
[2] S. Kandala and S. Deshpande, "Models for mpeg2," Sharp Laboratories of America, January 2001. [Online]. Available: http://www.opnet.com
[3] M. Krunz and H. Hughes, "A traffic model for mpeg-coded vbr streams," *in Proceedings of the ACM SIGMETRICS '95 Conference*, pp. 47–55, June 1995.
[4] H. Koumaras, C. Skianis, G. Gardikis, and A. Kourtis, "Analysis of h.264 video encoded traffic," *Proceedings of the 5th International Network Conference (INC2005)*, pp. 441–448, 2005.
[5] "Super©," eRightSoft, April 2006. [Online]. Available: http://www.erightsoft.com
[6] "x264 - a free h264/avc encoder," Released under the GPL, April 2006. [Online]. Available: http://developers.videolan.org/x264.html
[7] M. Krunz and S. K. Tripathii, "On the characterization of vbr mpeg streams," *Performance Evaluation Review (Proceedings of ACM SIGMETRICS'97 Conference)*, vol. 25, no. 1, pp. 192–20, June 1997.
[8] O. Rose, "Statistical properties of MPEG video traffic and their impact on traffic modeling in ATM systems," vol. 20, 1995, pp. 397–406. [Online]. Available: citeseer.ist.psu.edu/rose95statistical.html
[9] A. Matrawy, I. Lambadaris, and C. Huang, "Mpeg4 traffic modeling using the transform expand sample methodologys," Networked Appliances. IEEE 4th International Workshop on Network Appliances, 2002.
[10] I. E. G. Richardson, *H.264 and MPEG-4Video Compression: VideoCoding for Next-generation Multimedia*. John Wiley & Sons Ltd, 2003.
[11] A. S. Tanenbaum, *Computer Networks*, 4th ed. Pearson Education Inc, 2002.
[12] Z.-N. Li and M. S. Drew, *Fundamentals of Multimedia*. Pearson Education Inc, 2004.

If you wish to cite this paper, please use the following code: *Jeroen Doggen and Filip Van der Schueren, MPEG4 Part 10 - H.264 Modeling in OPNET, Masters Thesis, Department of Industrial Sciences and Technology, University College of Antwerpen, Belgium, July 2006*