Evaluation of Erlang Models in IP Network

M. Kavacký, E. Chromý, J. Šuráň*
* Institute of Telecommunications, FEI STU Bratislava, Slovakia
matej.kavacky@stuba.sk, erik.chromy@stuba.sk, jan.suran@stuba.sk

***Abstract*—Our paper deals with utilization and examination of Erlang traffic theory in asynchronous networks based on IP technology. We have proposed test network model with video traffic source. The obtained results have been compared with calculations performed in Matlab environment. We have aimed to loss parameter in our comparisons.**

I. INTRODUCTION

The IP networks were not designed for providing multimedia services at the beginning. These networks were intended for data transfer without predefined paths, so that in case of breakdown of one of the routers, other path could be chosen [1]. Path of packet can change and so can change transfer parameters.

If we want to use IP networks for voice, video and multimedia transmission, we have to define transfer parameters. These parameters are covered under QoS (*Quality of Service*). For observance of QoS [2]-[5] parameters it is necessary to estimate network performance under load and link overflow.

This problem is handled by theory of queuing systems described in detail in [6]-[9]. Therefore we will try to use Erlang models well known from synchronous telecommunication networks. Erlang B model deals with data loss in relation with network load. Erlang C model defines probability of waiting of data in the waiting queue and so resembles to IP network behavior, where data queuing occurs before sending further.

II. PROPOSED MODEL

The proposed model is based on Linux open-source operating system. In this system every part of the core is documented and available [10]-[13]. Thanks to this attribute, OS Linux or its core is implemented in many devices (routers). In our case we used linux distribution Debian 4.0, installed it on a PC that serves as a router, web server, file server and other applications server. There will also be installed a simulation program for traffic simulation, that will send video flows of defined parameters. Linux is capable of packet modification from its core and we will use this feature. Labeled packets of video flow then will be concentrated into one link or divided. We will install a custom package for traffic engineering, where classes, queues and filters can be created. These three main elements are defining routing mechanisms in router. This type of server with OS linux installed can be a low price alternative to routers used in commercial networks.

A. IP Tables

IP tables is a mechanism implemented directly into core of Linux operating system and serves for setting of IP traffic rules. It can change headers or drop packets. In our case, it will be used only for packet labeling. Labeled packets can be separated into individual data flows and then formed by traffic engineering package. IP tables consist of three parts: incoming, outgoing and forwarding. We can insert rules into these groups. If packets passes first rule, other rules in chain are not tested and packet is processed with the first rule definitions. For example, it can be labeled or dropped.

B. Classes

Classes serve for keeping our defined bandwidths. If we want to reduce speed to 2Mbit/s, we can create a class that ensure, that overflowed data will go to a queue and then will be sent. Linux uses two common classes:

- CBQ (*Classfull Based Queuing*).
- HTB (*Hierarchy Token Bucket*) [14].

C. Queues

Queues in traffic engineering influence which data will be send and which will have to wait in queue. There exist many mechanisms for queue managing and they are assigned to CBQ or HTB. Queues are class independent. Basic configuration of *Traffic control* package includes following types of queues:

- FIFO (*First in First Out*).
- PFIFO *(Priority* FIFO).
- SFQ (*Stochastic Fairness Queuing*).
- RED (*Random Early Detection*).

D. Filters

Last and very important part of our flow regulation package are filters. Filter serves for inserting data into the right class. In our case, filter chooses class by labels that were added by *IP tables*. But also these filters can analyze packet headers and assign data to right classes. These filters are paired with main classes and classes then insert data correctly based on corresponding filters.

E. Proposed Network Model

Our designed model of simulation environment is shown in Figure 1. We used one client for creation of video traffic and two clients for receiving and video data processing. Whole communication ran through server, which stood as a router and traffic shaper. This network was built and configured based on knowledge from literature [14]-[18]. All network elements were capable of 1GBit/s (if we did not use server for link limitation, we would not be able to correctly measure IP network behavior under load).

978-1-4577-0052-1/11/$26.00 ©2011 IEEE

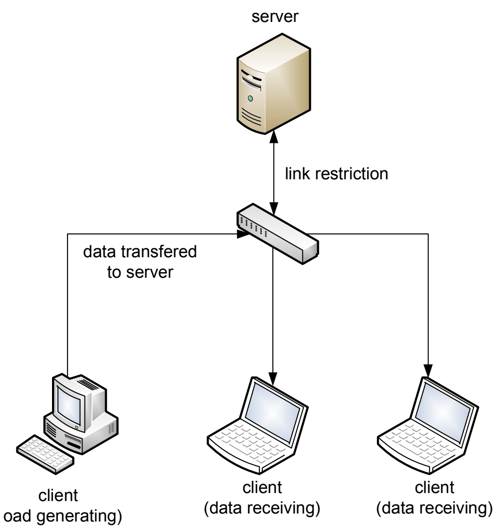

Figure 1. Proposed network model.

F. VBR Video Source

For real traffic simulation in IP network, simple model with JPG frame compression is sufficient for our purposes. As video source was used a webcam. It captured one minute long recording, all measurements used same data and so results were comparable. Our model is shown in Figure 2. and was designed based on information from [19]-[21].

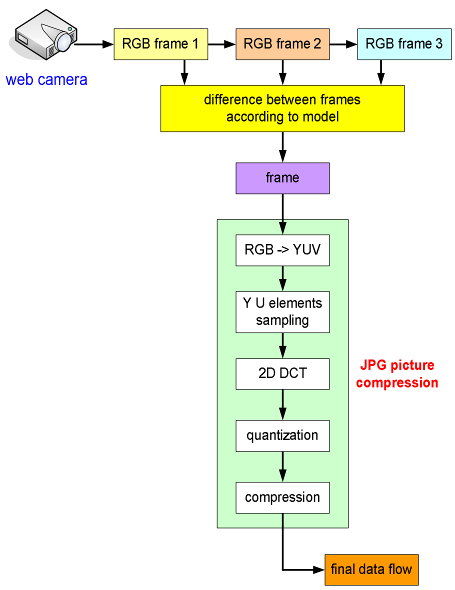

Figure 2. Video model with VBR data flow.

III. Erlang Models in Matlab

In 1917 Agner Kralup Erlang published *Solution of some Probléme in The Theory of Probabilities of Signifiance in Automatic Telephone Exchanges*. In this publication he stated his models for loss and waiting time in telephony traffic.

A. Erlang B Model

Erlang B model deals with probability of data loss under link load. Its basic form is (2):

$$B = E_{1,C}(\rho) = \frac{e^{-\rho} * \dfrac{\rho^C}{C!}}{\sum_{x=0}^{C} e^{-\rho} * \dfrac{\rho^x}{x!}} = \frac{\rho * E_{1,C-1}(\rho)}{C + \rho * E_{1,C-1}(\rho)} \quad (2)$$

where:

- ρ - total load [%],
- C - number of links.

We revised model (2) into form (3), which is more appropriate for calculations. From this model results relation of two parameters:

- ρ - link load in [%],
- C - link speed in Mbit/s.

$$E1 = \frac{\dfrac{\rho^C}{C!}}{\sum_{k=0}^{C} \dfrac{\rho^k}{k!}} \quad (3)$$

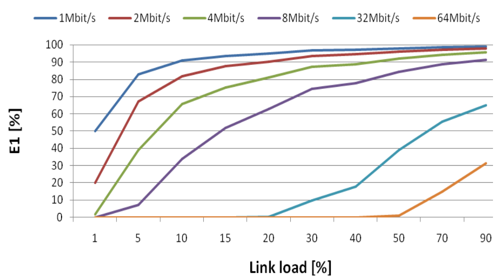

Figure 3. Data loss probability in the case of constant bandwidth.

From Fig. 3 we can see following results:

- Probability of loss increases with link load.
- With increasing bandwidth data loss is decreasing.

After revision and modification of Erlang model with parameter *m*, (number of sources in common path) we used model (4) for further calculations. Other parameters stayed the same.

$$E_1(\rho, c, m) = \frac{\dfrac{(m.\rho)^c}{c!}}{\sum_{i=0}^{c} \dfrac{(m.\rho)^i}{i!}} \quad (4)$$

Common path means that more flows are transferred through one way. Link bandwidth requirements are then summarized together.

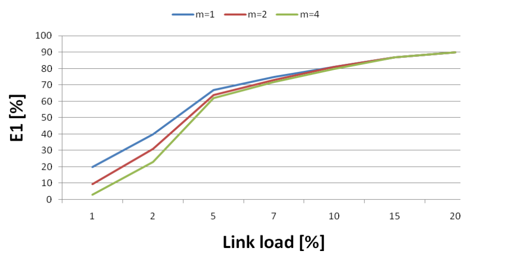

Figure 4. Data loss probability in the case of constant bandwidth 2 Mbit/s and m flows.

From Fig. 4 we can see following results:

- Under low link load with increasing number of sources is data loss probability reduced.
- Under higher link load (5%) this difference is not as much notable and the loss is gradually flatten.

B. *Erlang C Model*

Probability of waiting the requests in queue is described by second Erlang model (5).

$$E_{2,C}(\rho) = \frac{\frac{\rho^C}{C!} * \frac{C}{C-\rho}}{\sum_{x=0}^{C-1} \frac{\rho^x}{x!} + \frac{\rho^C}{C!} * \frac{C}{C-\rho}} \tag{5}$$

For calculations form without factorials is preferable. In case that $\rho < \mathrm{C}$, we can rewrite model to (6):

$$E_{2,C} = E_{1,C} * \frac{C}{C - \rho[1 - E_{1,C}(\rho)]} \tag{6}$$

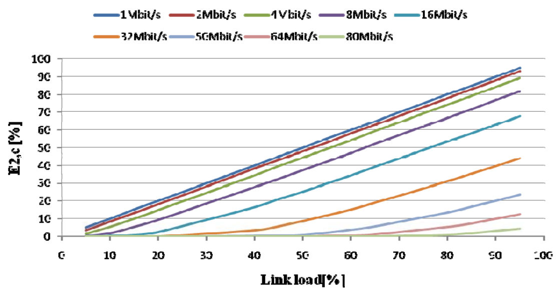

Figure 5. Probability of enqueue in case of constant bandwidth.

From Fig. 5 we can see following results:

- With increasing link load, probability of enqueue also increases.
- With increasing bandwidth, probability of enqueue decreases.
- With low link speeds (under 16 Mbit/s), the behaviour of probabilities is nearly linear.

Model (6) extended by *m* parameter:

$$E_{2,C}(m*\rho) = \frac{\frac{(m*\rho)^{Ccp}}{Ccp!} * \frac{Ccp}{Ccp-(m*\rho)}}{\sum_{x=0}^{Ccp-1} \frac{(m*\rho)^x}{x!} + \frac{(m*\rho)^{Ccp}}{Ccp!} * \frac{Ccp}{Ccp-(m*\rho)}} \tag{7}$$

where:

- ρ - link load [%].
- C_{cp} – link speed (for common path) in Mbit/s.
- m – number of sources.

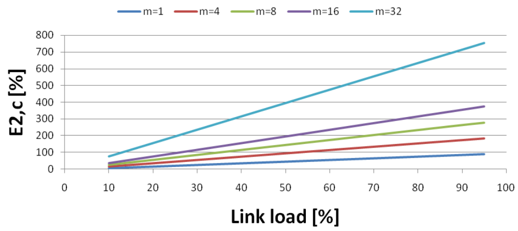

Figure 6. Probability of enqueue in case of common utilization of 4 Mbit/s link.

From Fig. 6 we can see following results:

- With increasing link load and constant bandwidth, the probability of enqueue increases.
- With higher common path utilization, the probability of enqueue is also higher.
- With increasing of common path bandwidth, it is possible multiple use of traffic links also with higher load.
- Common path bandwidth affects slope of probability of enqueue.

IV. Discussion of Results

When comparing results from MATLAB simulations and real measurements, it is necessary to consider VBR video characteristics and used queues for traffic shaping. For simulation of model B we used RED mechanism, because packets were dropped after reaching maximum queue length. Model C uses SFQ mechanism, which does not drop packets, but inserts them into waiting queue.

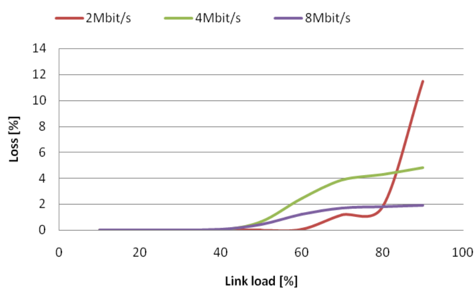

Figure 7. Loss probability in case of constant bandwidth.

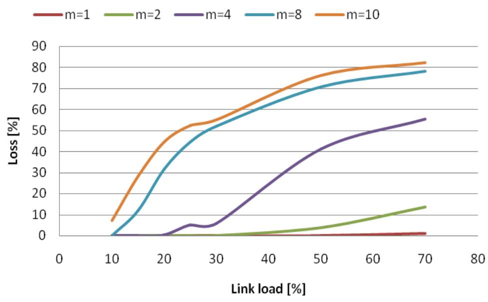

Figure 8. Loss probability in case of 2 Mbit/s link and *m* flows.

A. *Erlang B Model*

Firstly we measured loss in relation to link load at 2 Mbit/s link capacity. In Figure 7 we can see that graphs look like in simulations (section III.A), but values of loss are much lower (loss decreases with link capacity). This is caused by VBR video characteristics, where loss does not occurs until reaching link capacity.

The figure 8 shows Erlang B model with common link. Each flow was transferred with some time delay against other flows. This way we prevented overloading of the link by VBR video. From this measurement results:

- With constant network load and increasing bandwidth it is possible to raise the number of sources.
- With constant load and bandwidth, loss increases with number of sources per link.
- With increasing link capacity we can add number of sources per one path.

B. *Erlang C Model*

The method was the same as with Erlang B model, except for the fact, that data which came later, were labeled as lost, because they went to waiting queue.

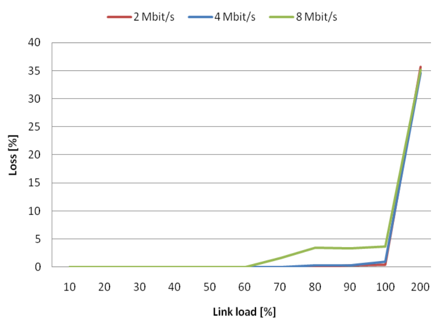

Figure 9. Loss probability in case of constant bandwidth.

From Fig. 9 we can see following results:

- SFQ waiting queue is a better choice for VBR video, because of its better handle with high-peak rate.
- Compared to MATLAB simulations, loss is under 5% even until full network usage. This is caused by SFQ queue and VBR data model.

We also tried the common path utilization with Erlang C model. Flows were put together in the same way as with Erlang B model. The results are depicted in the Fig. 10:

- Graphs are now comparable with MATLAB simulations, but with lower probability of loss.
- Similar to Erlang B model, it is useful to put together more flows into one path.
- With increasing bandwidth, we can put together more flows into one path and assure acceptable loss (based on SLA).

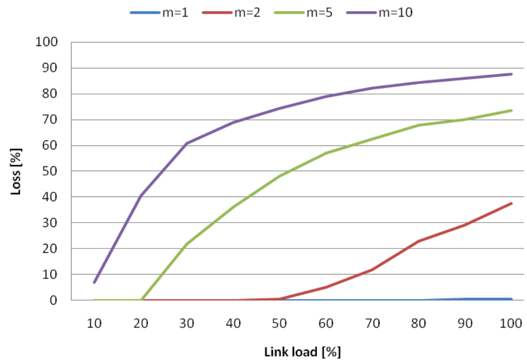

Figure 10. Loss probability in case of *m* flows and 2 Mbit/s link.

V. Conclusion

The purpose of our work was to evaluate utilization of Erlang B and C models in IP networks with VBR video traffic source. We have pointed out the difference between two link shaping methods and we have found out that SFQ is more suitable for this utilization. Erlang model is then a good choice for VBR traffic calculations with respect to losses. It is possible to estimate network behavior under load with these models. These models give us only approximately estimation. Loss probabilities were not exactly as predicted by calculations, but lower. It is caused by VBR video characteristics and data transfer method in IP networks. Calculations and real measurements have shown advantages of effective link load. Putting more VBR flows together into one link has greatly improved effectiveness of data transfer.

Acknowledgment

This work was supported in part by the Grant VEGA no. 1/0565/09 “Modelling of traffic parameters in NGN telecommunication systems and networks”.

References

[1] Dostálek, L., Kabelová, A. Understanding TCP/IP, Packet, Birmingham, 2006, ISBN 1-904811-71-0.

[2] Kováč, A., Halás, M. Vozňák, M. “Impact of Jitter on Speech Quality Estimation in Modified E-Model,” *12th International Conference Research in Telecommunication Technology RTT 2010*, Velké Losiny, Czech Republic, VŠB Ostrava, September 8 – 10, 2010, 225 p. ISBN 978-80-248-2261-7.

[3] Mišuth, T., Baroňák, I. "Performance Forecast of Contact Centre with Differently Experienced Agents," *Elektrorevue*. ISSN 1213-1539. Vol. 15, 16.11.2010 (2010), art. no. 97.

[4] Baroňák, I., Mičuch, J. "Preventive Methods Supporting QOS in IP," *New Information and Multimedia Technologies. NIMT - 2009: Brno*, Czech Republic, 17. - 18. 9. 2009. Brno: Brno University of Technology, 2009. ISBN 978-80-214-3930-6. pp. 15-23.

[5] Balogh, T., Medvecký, M.: Comparison of Priority Quauing Based Scheduling Algorithms. In: Elektrorevue, vol. 15, 16.11.2010 (2010), art. no 93. ISSN 1213-1539.

[6] Wootton, C. "A practial guide to video and audio compression," *Focal press*, USA 2005, ISBN 0-240-80630-1.

[7] Peško, S. "System queuing theory," 2001, http://frcatel.fri.uniza.sk/users/pesko/OA2/oa2m.pdf

[8] Hayes, J. F. et al., "Modeling and Analysis of Telecommunications Networks," Wiley Interscience, 2004.

[9] Iversen, V. B. "Teletraffic Engineering Handbook," COM Centre, Technical University of Denmark, 2001.

[10] Graf, T., Maxwell, G., Mook, R. "Linux Advanced Routing & Traffic Control," 2004 http://lartc.org/

[11] Thomas, K. "Beginning Ubuntu Linux From Novice to Professional," *Apress*, 2006, ISBN 1-59059-627-7.

[12] Bauer, M. "Building Secure Servers with LINUX," O'Reilly, 2003, ISBN 0-596-00217-3.

[13] Hunger, S. "Debian GNU-Linux Bible, Hungary Minds," New York, 2001, ISBN 0-7645-4710-0.

[14] Kist, A. "Erlang B as a Performance Model for IP Flows Networks," *15th IEEE International Conference*, 2007, ICON 2007.

[15] Hofmann, M., Beaumont, L. "Content Networking," *MK Publishers*, San Francisco, 2005, ISBN 1-55860-834-6.

[16] Davies, G. "Designing and Developing Scalable IP Network," *John Wiley & Sons Ltd*, England, 2004, ISBN 0-470-86739-6.

[17] Held, G. "Ethernet Networks – Fourth Edition," *John Wiley & Sons Ltd*, England, 2003, ISBN 0-470-84476-0.

[18] Plumley, S. "Home Networking Bible," *John Wiley & Sons Ltd*, England, 2004, ISBN 0-7645-4416-0.

[19] Pirogov, V. "Mastership in assembler language," *Computer Press*, Brno, 2005, ISBN 80-251-0888-0.

[20] Pirkl, J. "Network programming in Windows and internet programming," *Kopp*, České Budějovice, 2001, ISBN 80-7232-145-5.

[21] Chalupa, R. "1001 tips and tricks for Visual C++," *Computer Press*, Brno, 2003, ISBN 1-904811-71-0.

[22] Tisovský, A.: Method for Throughput Interpolation of Computationally Intensive IPsec Process. In: Telecommunications and Signal Processing TSP-2010 : 33rd International Conference on Telecommunications and Signal Processing. Baden near Vienna, Austria, 17.-20.8.2010. - Budapest : Asszisztencia Szervezö Kft., 2010. - ISBN 978-963-88981-0-4.

[23] Balogh, T., Medvecký, M.: Performance Analysis of Priority Quauing Based Scheduling Algorithms. In: Telecommunications and Signal Processing TSP-2010 : 33rd International Conference on Telecommunications and Signal Processing. Baden near Vienna, Austria, 17.-20.8.2010. - Budapest : Asszisztencia Szervezö Kft., 2010. - ISBN 978-963-88981-0-4. pp. 228-233.

[24] Tisovský, A., Kľúčik, S. Method for Calculation the Packet-Size Dependent throughput of a Computationally Intensive IPsec Process. In: Elektrorevue, vol. 15, 16.11.2010 (2010), art. no 98, ISSN 1213-1539.

[25] Voznak, M., Rozhon, J. Performance testing and benchmarking of B2BUA and SIP Proxy. In: 33rd International Conference on Telecommunication and Signal Processing, AUG 17-20, 2010 Vienna, Austria, pp. 497-503.