TFRC and RTT Thresholds Interdependence in a Selective Retransmission Scheme

Árpád Huszák, Sándor Imre

Budapest University of Technology and Economics, Department of Telecommunications
Budapest, Hungary
Email: huszak@hit.bme.hu, imre@hit.bme.hu

Abstract—**Retransmission based error recovery is not always applicable for multimedia applications; however it can be used if the transmission characteristics and network conditions are examined before. Retransmissions can be carried out only if there is no congestion in the network and the network delay is low enough to receive the retransmitted packet before the playout. These two attributes are depending and one can be estimated from the other. In this paper we introduce a TFRC and RTT threshold interdependence model to determinate network delay thresholds for the retransmission decision algorithm. Using the calculated sending rate of the TFRC (TCP Friendly Rate Control) to estimate the congestion level in the network, we can set up a threshold of disabling/enabling the retransmissions. Our goal was to determinate a matching RTT threshold to gain similar retransmission behavior as using the sending rate threshold. We found that using our threshold estimating model, the number of retransmitted packets was similar, using the TFRC-based sending rate threshold and the RTT retransmission threshold.**

Keywords: TFRC, Round-Trip-Time, retransmission

I. INTRODUCTION

Real-time multimedia applications are already popular in reliable wired networks, but in wireless environment the obstacles of the expansion are the higher bit error ratio of the radio link and the limited bandwidth of the mobile links. Packet losses generally degrade the performance of any Internet data transfer especially on compressed data. Inter-frame-video compression algorithms such as MPEG exploit temporal correlation between frames to achieve higher compression therefore errors in a reference frame will propagate to the dependent difference frames. To minimize the end-to-end packet loss ratio the packet loss should be either prevented or subsequently handled. Despite its latency, retransmission can be used successfully in many cases, especially if playout buffering is employed.

Late retransmissions in real-time application are undesirable because the late packets will be dropped at the receiver. For the retransmission to be successful, retransmitted packet must arrive at the receiver in time for playback and should not be lost again due to congestion. When the network is in congested state or the RTT (round-trip-time) is so high that the retransmitted packet will not arrive in time, the retransmission will not improve the quality; moreover will increase the load and latency. The correlation of congestion level and RTT is obvious and already studied intensively. Numerous researches focus on end-to-end congestion avoidance algorithms that use round-trip-time fluctuations as an indicator of the level of network congestion. [4]–[8]. In these works RTT variations are assumed to be caused by changes in queuing delays, therefore the congestion level can be estimated from the current RTT and its variation history.

Loss-tolerant real-time multimedia applications prefer unreliable transport protocols, like UDP or UDPLite, but the DCCP (Datagram Congestion Control Protocol) [1] is also very advantageous, because it uses sequence numbering, acknowledgements and congestion control algorithms. From the retransmission point of view sequence numbers and acknowledgement are needed to identify the lost packets, while the congestion control algorithms (TCP-Like Rate Control [2]and TCP Friendly Rate Control [3]) manage the actual round-trip-time measurements. Due to these advantages, in our measurement we applied DCCP to get the necessary information for the retransmissions.

In our previous works we have investigated cross-layer based retransmission methods. First, a TFRC-based retransmission scheme [9][10] was implemented, which disable or enable the retransmission according to a given threshold and the calculated TFRC [11] sending rate. Secondly, we have studied a RTT-based retransmission decision algorithm based on the actual RTT, in order to determinate an appropriate threshold for successful retransmissions [12].

In this paper we introduce a TFRC and RTT thresholds interdependence model, which utilize the correlation of the TFRC sending rate and the RTT. The model estimates the correspondent RTT threshold using the TFRC-based sending rate threshold gaining similar retransmission behavior. Using the proposed model the number of retransmitted packets is similar in both cases, because we use corresponding RTT and TFRC throughput thresholds. In order to analyze the proposed method we used the Ns-2 simulation tool to examine the retransmission behaviors.

The rest of the paper is organized as follows. The background work is presented in Section II. In Section III we introduce our TFRC and RTT thresholds interdependence model for real-time applications. Performance evaluation of our method follows in Section IV. Finally, we summarize our paper and make the conclusions in the last section.

II. BACKGROUND

In order to efficiently retransmit the lost packets without late delivery or causing congestion, we have analyzed two retransmission approaches. Both cross-layer based methods rely on the information getting from the transport layer (DCCP). The DCCP protocol continuously measures the RTT and calculates a sending rate to avoid congestion.

A selective retransmission scheme is considered which disable or enable the retransmission of lost packets according to the current state of the network. In this scheme the retransmission should be enabled or disabled according to the stream bitrate (threshold level) and the calculated TFRC sending rate [9]. When the network is in congested state or near to this state the retransmissions should be disabled. When the buffers of the network routers are overloaded the retransmitted packet will be dropped at the routers or it will cause the loss of other packets. The method uses the TFRC congestion avoidance algorithm to decide whether the lost packet should be retransmitted. It disables the retransmission when the TFRC rate is under a given threshold ($X_{Threshold}$), e.g. real-time video coding bitrate.

$$X_{Threshold} > \frac{s}{R\sqrt{\frac{2p}{3}} + 4R(3\sqrt{\frac{3p}{8}} \cdot p \cdot (1 + 32p^2))} \quad (1)$$

Variable p stands for the packet loss probability and R is the RTT moving average and s is the packet size in bytes, as given in RFC 3448.

On the other hand we can consider a delay-based retransmission technique [12], where the available time for the first transmission, retransmission and loss detection is equal to the playout buffer delay (T_{bd}), therefore the stipulation to successfully retransmit a packet is

$$\frac{3}{2}RTT + \delta < T_{bd}, \quad (2)$$

where δ is the loss detection delay. If (2) is true the retransmission is enabled, considering that only one retransmission attempt is possible, however the equation can be easily modified for multiple retransmission attempts. If (2) is false the retransmission is disabled, because the retransmitted backet will not arrive in time. The playout buffer delay (T_{bd}) is a time constrain of the retransmissions, determining the maximal acceptable round-trip-time ($RTT_{Threshold}$):

$$RTT_{Threshold} = \frac{2}{3}(T_{bd} - \delta) \quad (3)$$

$RTT_{Threshold}$ is used to decide whether to retransmit a lost packet or not, e.g. if the current RTT is lower than the $RTT_{Threshold}$ the retransmission is enabled, because the retransmitted will probably arrive in time.

The presented retransmission schemes were investigated independently, however the calculated TFRC rate depends on the RTT moving average. The used TFRC rate and RTT retransmission thresholds are therefore correlated. In this paper this correlation is investigated.

III. TFRC AND RTT THRESHOLDS INTERDEPENDENCE MODEL

In this paper we examine the relation between the $X_{Threshold}$ and the $RTT_{Threshold}$. In order to efficiently estimate the $RTT_{Threshold}$ from the $X_{Threshold}$, first the relation of RTT and TFRC must be analyzed. The RTT and TFRC values are in inverse relation, as the TFRC equation it defines (1).
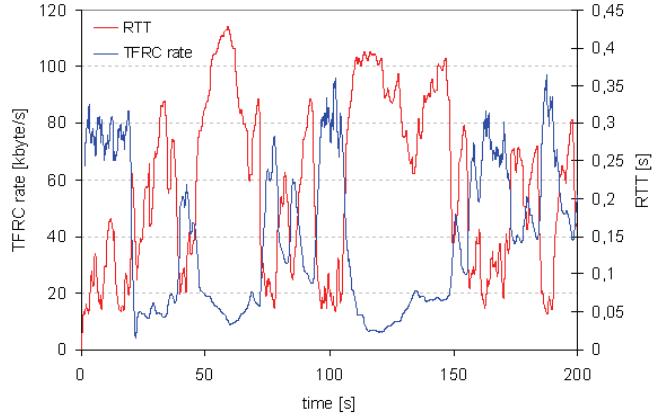

Figure 1.   Measured RTT and TFRC sending rates. The link capacity was 1Mbps, the offered traffic rate was 300kbps, the constant bitrate background traffic was 300kbps, while the link packet loss probability was 0.1%.

The figure illustrates the inverse relation of RTT and TFRC values. It can be observed that in time periods when the calculated TFRC is rate is low, the measure RTT is higher. This behavior is due to the increasing buffer levels of the routers in the network. The TFRC decreases the sending rate when there are too many packets in the network and the possibility of packet drop due to congestion is becoming high.

In our estimation model we utilize the inverse relation of RTT and the offered sending rate of TFRC. Our aim is to determinate $RTT_{Threshold}$ analyzing the number of retransmitted packets. The number of retransmitted packets should be equal using the TFRC-based threshold and the $RTT_{Threshold}$.
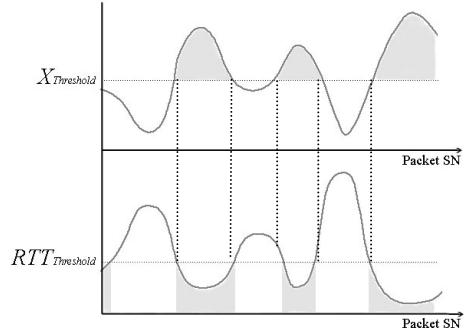

Figure 2.   Retransmissions are enabled for the packets belonging to the grey periods. (SN stands for Sequence Number)

The number of successfully retransmitted packets depends on the time periods when RTT< $RTT_{Threshold}$ or the calculated TFRC sending rate is higher than $X_{Threshold}$. The periods when the retransmission can be performed successfully are signed grey on Figure 2. According to the TFRC sending rate calculation method the sending rate and the measured round-trip-time change inversely. In order to formally define the number of possible retransmissions, the X' and R' indicator functions must be specified. Both X' and R' indicator functions specify whether a packet is allowed to be retransmitted or not.

$$X' = \begin{cases} 1 & \text{if } X > X_{Threshold} \\ 0 & \text{else} \end{cases} \qquad (4)$$

$$R' = \begin{cases} 1 & \text{if } RTT < R_{Threshold} \\ 0 & \text{else} \end{cases} \qquad (5)$$

The sum of X' determinates the maximum number of retransmission, however actually only the lost packets can be retransmitted, so it must be multiplied by the packet loss probability. The estimated value of the number of retransmitted packets can be calculated as follows:

$$P_X = p \cdot \sum_{SN=0}^{SN_{max}} X' \; ; \; P_R = p \cdot \sum_{SN=0}^{SN_{max}} R' \qquad (6)$$

Our aim is to find the matching $RTT_{Threshold}$ to $X_{Threshold}$ attaining $P_X = P_R$. If SN_{max} is the number of transmitted data packets, we can consider $\sum_{SN=0}^{SN_{max}} X' \Big/ SN_{max}$ as the probability that a packet is enabled to be retransmitted. According to the X' and R' definition, we can calculate the ratio of the periods, when the retransmission is possible:

$$p(X > X_{Threshold}) = p(RTT < RTT_{Threshold}) = \frac{\sum_{SN=0}^{SNmax} X'}{SN_{max}} = \frac{\sum_{SN=0}^{SNmax} R'}{SN_{max}} \qquad (7)$$

The possibility that the actual round-trip-time is less than the retransmission threshold is:

$$p(RTT < RTT_{Threshold}) = F(RTT_{Threshold}) \qquad (8)$$

According to the distribution function determination, the given probability (8) is the distribution function of $RTT_{Threshold}$. Similarly for the sending rate:

$$p(X > X_{Threshold}) = 1 - p(X < X_{Threshold}) = 1 - F(X_{Threshold}) \qquad (9)$$

Using equation $p(X > X_{Threshold}) = p(RTT < RTT_{Threshold})$, that determinates the relation of $RTT_{Threshold}$ and $X_{Threshold}$, we can proof that the relation of the distribution functions of $RTT_{Threshold}$ and $X_{Threshold}$ is linear:

$$F(RTT_{Threshold}) = 1 - F(X_{Threshold}) \qquad (10)$$

The TFRC specification [11] determinates the calculated sending rate (X_{calc}) as inverse function of RTT variable; however the real sending rate is determined as follows:

$$X = \max(\min(X_{calc}, \; 2 \cdot X_{recv}), \; s/t_{mbi}) \qquad (11)$$

Variable X_{recv} is the receiving rate of the receiver. The parameter t_{mbi} is the the maximum inter-packet backoff interval and set to 64 seconds, thus the sender sends at least one packet every 64 seconds. We did not take the backoff interval value into account, because we consider that the data delivery or the video streaming is continuous. In situations when the offered sending rate (e.g. video bitrate) is low, compared to the link capacity, the X_{calc} can be significantly higher than $2X_{recv}$. According to (11), the real sending rate is not always the calculated sending rate (X_{calc}), therefore the sending rate is not the inverse function of the round-trip-time.

Considering the TFRC sending rate determination method we propose a linear estimation model to find the matching $RTT_{Threshold}$ to every $X_{Threshold}$. The maximum of the sending rate is determined as $2X_{recv}$, while RTT_{min} and RTT_{max} parameters are assumed as known values. RTT_{min} is the network delay when the network routers' buffers are empty, while RTT_{max} can be assumed as the round-trip-time when the network is in congested state. Based on presented behavior of the TFRC sending rate calculation, we can estimate the corresponding thresholds. The next figure illustrates the proposed linear estimation model:
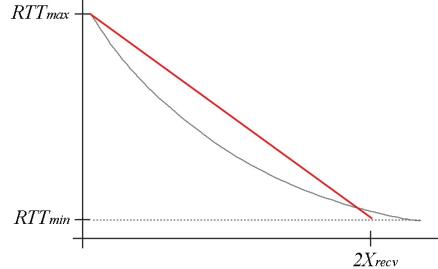

Figure 3. Linear estimation of $RTT_{Threshold}$

The formal determination of the TFRC and RTT thresholds interdependence is done in the following equation:

$$RTT_{Threshold} = \frac{(RTT_{min} - RTT_{max})}{2 \cdot X_{recv}} X_{Threshold} + RTT_{max} \qquad (12)$$

We used linear approximation instead of exponential one, because the infection level of the exponential curve can vary depending on the actual loss rate and RTT, as the TFRC equation (1) defines it. The packet loss rate is on a higher exponent in the equation, thus its effect is more significant. The used RTT_{min} and RTT_{max} parameters are also changing, because always the lowest/highest measured RTT values are considered as RTT_{min} and RTT_{max}. The estimator model must adapt itself to the varying parameters.

IV. Simulation Results

In order to test the performance of the estimation model, described in the previous section, we analyzed some scenarios with ns-2 [13] network simulator. The simulation environment made it possible to adjust the link characteristics and analyze the effects of additional background traffic. In the simulations a *dumbbell* topology with 1Mbps links. The bottleneck link was considered to be the wired or wireless link, in order to introduce packet losses using a simple random drop model with the given loss probability. The loss probability on link S–R was set between 0% and 10%.

Figure 4. The *dumbell* network topology

In the simulations we used the Datagram Congestion Control Protocol (DCCP) [1] with TFRC congestion control algorithm to analyze the performance of the estimation model. Furthermore WWW and CBR (Constant Bit Rate flow) over UDP background traffic were set up. In order to simulate the effects of a heavy loaded network the number of WWW users was between 0 and 25. The WWW traffic was generated based on Pareto distribution with average page sizes of 20kB (shape 1.5), while the frequency of new WWW page requests were following the Exponential distribution with 5 seconds average waiting times. The duration of the simulations was 200s. The purpose of the background traffic was to continuously change the conditions of the bottleneck link causing high variations in the measured RTT.

In the first scenario a well-defined setup was analyzed to introduce the estimation model. The offered bitrate (CBR/DCCP) was 300kbps (37.5kbyte/s) and delivered by
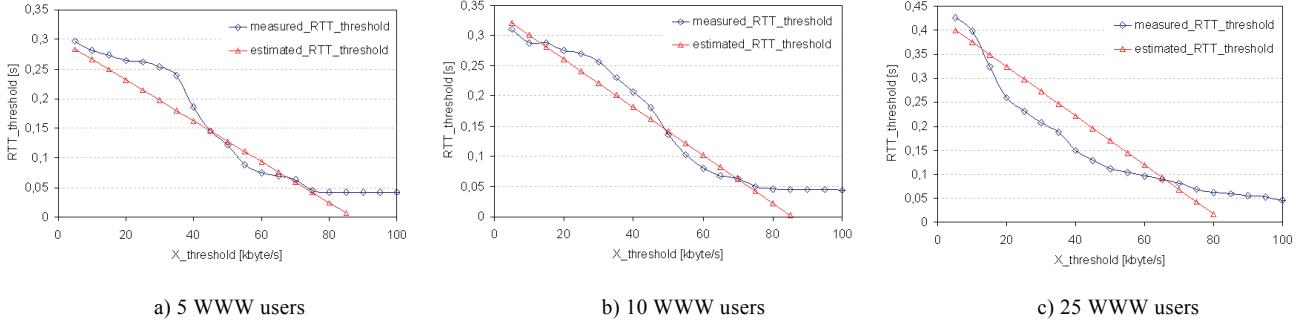
the DCCP protocol, in the background 100kbps CBR/UDP was set up and the number of WWW users was 5, 10 and 25. We have measured the corresponding thresholds, when the number of retransmissions is the same.

The measured results are illustrated in Figure 6. In the figure the blue line indicates the matching $RTT_{Threshold}$ having the same number of retransmissions like

using $X_{Threshold}$. The red line is generated by our estimator model. In order to analyze the quality of the estimation, we have examined the difference of the measured $RTT_{Threshold}$ and the estimated one. The estimation error was calculated as follows:

$$err[\%] = \frac{mRTT_{Threshold} - eRTT_{Threshold}}{mRTT_{Threshold}} \cdot 100\% \qquad (13)$$

Variable $mRTT_{Threshold}$ stands for the measured RTT threshold and $eRTT_{Threshold}$ for the estimated threshold. According to (13) the $err[\%]$ function can be negative too.

We have changed the congestion level in the network by increasing the number of WWW users, without changing other settings.
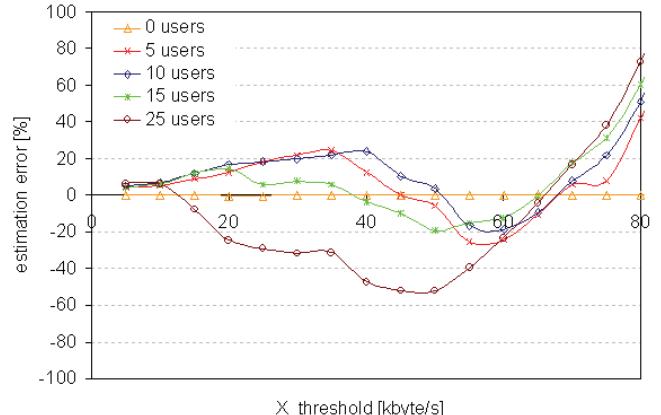

Figure 5. Estimation error in function of the number of WWW users. The offered traffic rate is 300kbps, theCBR/UDP background traffic is 100kbps.

The best threshold estimation was measured when the background WWW traffic was minimal. Increasing the number of WWW users the estimation performance

a) 5 WWW users

b) 10 WWW users

c) 25 WWW users

Figure 6. Linear estimation and measured values of $RTT_{Threshold}$. The number of WWW users is 5 , 10, 25

decreases, while without WWW users the estimation error is near zero. According to the Ns-2 simulations the congestion level plays an important role in the estimation effectiveness. When we increase the offered bitrate (CBR/DCCP) to 500kbps, change the background CBR/UDP traffic and switch off the random WWW user traffic, the obtained results are similar.
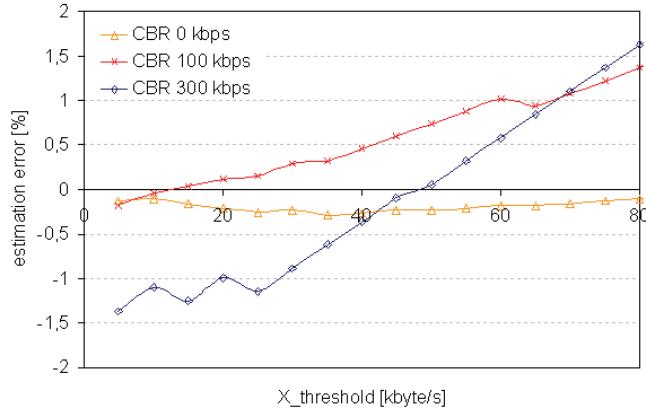

Figure 7. Estimation error in case of different CBR/UDP traffic with 500kbps CBR/DCCP setup

Figure 7. shows, if only the 500kbps data stream is transmitted with the DCCP/TFRC protocol, the $RTT_{Threshold}$ estimation is sufficiently accurate. When we change the sending rate retransmission threshold ($X_{Threshold}$), the estimation error of the delay-based threshold is less than ±0.5%. As the results show our estimation model performs well when the network load is low.

The other way of increasing the network load is to set the data stream (CBR/DCCP) bitrate higher. In next scenario we analyzed the different bitrate data streams delivery over DCCP/TFRC without any additional background traffic. The retransmission threshold ($X_{Threshold}$) maximal value is the highest calculated TFRC sending rate during the simulation. This is the reason for having different maximal $X_{Threshold}$ values in the simulations. The obtained results are presented in the following figure.
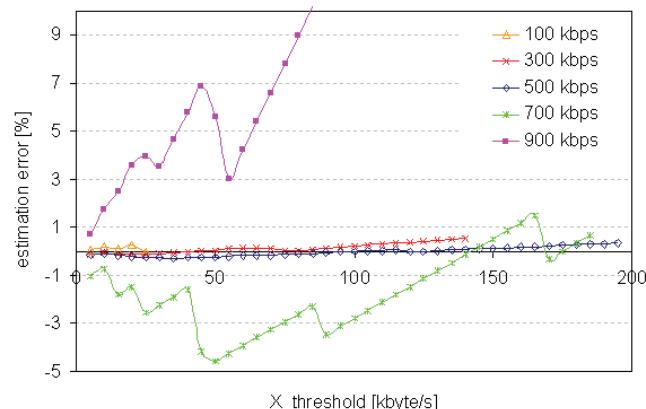

Figure 8. Estimation error in case of different bitrate data streams delivery without background traffic

Similarly to previous simulation results, the estimation effectiveness is also adequate when the network load is not too close to the network capacity. In the analyzed network with 1Mbps bottleneck link, the estimation error was below 1% when the source bitrate was less or equal to 500kbps. However, in case of 700kbps offered bitrate, the estimation error is maximum 5% that is still acceptable.

In order to examine the overall performance of the TFRC and RTT thresholds interdependence model we have calculated the average estimation errors corresponding to different sending rate thresholds. Using average values of the estimation error we can represent different sending rate thresholds ($X_{Threshold}$) as one point in the graph. Using this method we analyzed the accuracy of the estimations by changing the offered bitrate and the number of WWW users simultaneously. The simulation measurements are presented in Figure 9. The evaluated results show that the number of WWW users may drastically decrease the estimation effectiveness, as well the source bitrate. The average estimation errors are close to zero when the background traffic is zero. By setting the offered bitrate of the data stream not higher than 500kbps, the estimation error can be retained below 30%. The estimation of the delay-based threshold from the sending rate threshold is inefficient when the network is overloaded. In case of 900kbps source rate and additional background rate, the estimation error can be even 50%.
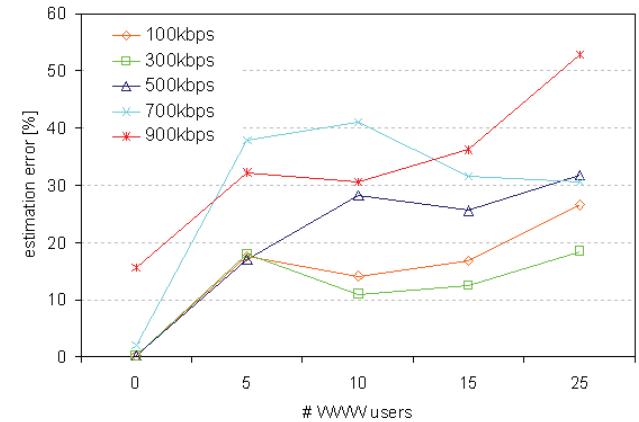

Figure 9. Average estimation error

In the last analyzed scenario the effect of packet losses was examined. The TFRC can not distinguish between wireless and congestion losses and it will decrease the sending rate in both cases, thus the expected behavior is similar to the overloaded network case. However, according to our measurements the effect of packet losses is not so significant than the background traffic increase. In the next figure different packet loss rates are analyzed. It depicts how effectively our estimation model can give a RTT threshold to a sending rate based retransmission threshold.
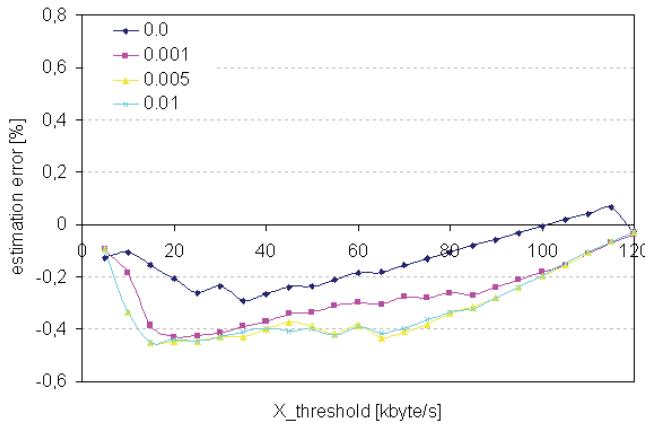
Figure 10. Estimation accurancy in case of packet drops

In this scenario a 500kbps bitrate data stream was used without background traffic, while the link capacity is 1Mbps. As previously presented the performance of the proposed method is high when the network load is low. Simple packet losses can also degrade effectiveness as Figure 10. show. Of course the best matching estimation was done without packet losses.

V. CONCLUSION

Retransmission-based error recovery for real-time multimedia applications is applicable only if efficient support is available to avoid waste retransmissions. Congestion-based (congestion avoidance) and delay-based (late retransmission avoidance) retransmission decisions are good solutions to provide selective error recovery. In both cases the decisions are made using sending rate- and RTT thresholds.

In this paper we analyzed the relation of the two retransmission decision methods and introduced an estimation model to find the matching RTT threshold to the sending rate threshold. The obtained results show that the model can be really effective when the network load is low and in average the estimation error is 20-30%, however the network load significantly affects the performance of the proposed method. When the network load is below the congestion load, the estimation error is only 1-2%.

REFERENCES

[1] Kohler, Handley, Floyd,"Datagram Congestion Control Protocol", Internet Engineering Task Force, RFC 4340, March 2006.

[2] S. Floyd and E. Kohler, "Profile for DCCP Congestion Control ID 2: TCP-like Congestion Control", draft-ietf-dccp-ccid2-10.txt, March 2005.

[3] S. Floyd, E. Kohler, and J. Padhye, "Profile for DCCP Congestion Control ID 3: TFRC Congestion Control", draft-ietf-dccp-ccid3-11.txt, March 2005.

[4] Jim Martin , Arne Nilsson , Injong Rhee, "The incremental deployability of RTT-based congestion avoidance for high speed TCP Internet connections", ACM SIGMETRICS Performance Evaluation Review, v.28 n.1, p.134-144, June 2000

[5] Jim Martin , Arne Nilsson , Injong Rhee, "Delay-based congestion avoidance for TCP", IEEE/ACM Transactions on Networking (TON), v.11 n.3, p.356-369, June 2003

[6] Li Yan, Bin Qiu, Lichang Che, "A Delay-Based End-to-End Congestion Avoidance Scheme for Multimedia Networks", Advances in Multimedia Information Processing - PCM 2004, LNCS 3332, pp. 389–396, 2004.

[7] Raj Jain, "A delay-based approach for congestion avoidance in interconnected heterogeneous computer networks", ACM Computer Communication Review, 19(5):56{71, Oct. 1989.

[8] C. Jin, D. X. Wei, S. H. Low, "The case for delay-based congestion control", in Proceedings of IEEE Computer Communication Workshop (CCW), pp. 99-104. Laguna Beach, CA, Oct 2003.

[9] Árpád Huszák, Sándor Imre: "TFRC-Based Selective Retransmission for Multimedia Applications", 5th @WAS International Conference on Advances in Mobile Computing and Multimedia, MoMM2007, ISBN 978-3-85403-230-4, pp. 53-63., Jakarta, Indonesia, 3-5 December 2007

[10] Árpád Huszák, Sándor Imre: "Source Controlled Semi-reliable Multimedia Streaming Using Selective Retransmission in DCCP/IP Networks", Elsevier – Computer Communications, The International Journal for the Computer and Telecommunications Industry, Volume 31., Issue 11., pp. 2676–2684, ISSN 0140-3664, 15 July 2008

[11] S. Floyd, M. Handley, J. Padhye, J. Widmer, "TCP friendly rate control (TFRC): protocol specification", RFC 3448, January 2003.

[12] Árpád Huszák, Sándor Imre: "Source Controlled and Delay Sensitive Selective Retransmission Scheme for Multimedia Streaming", IEEE Symposium on Computers and Communications, ISCC 2007, ISBN 1-4244-1521-7, pp. 191-196., Aveiro, Portugal, 1-4 July, 2007

[13] Ns-2 – Network Simulator, http:///www.isi.edu/nsnam/ns/index.html