Performance Evaluation of Cross Layer Adaptive Multimedia Transmission: The case of Wired Networks

Christos Bouras, Apostolos Gkamas and Georgios Kioumourtzis

Abstract— **This paper presents the performance evaluation of the wired part of a proposed framework with cross layer adaptation mechanisms, for multimedia transmission over wired and wireless networks. The performance evaluation of the proposed framework is contacted through a number of simulations with the network simulator software (ns2), which has been enhanced in order to support the functionality of the proposed framework. The performance evaluation includes various simulation scenarios in order to investigate the behavior of the proposed framework under different conditions. The proposed framework can support both wired and wireless receivers in one platform in which cross layer information are used to support the wireless users, whereas TCP friendly estimations are used to support the wired users.**

Index Terms— **Cross layer adaptation, Multimedia communication, Wireless networks, Network Simulator 2, RTP protocol**

I. INTRODUCTION – RELATED WORK

MULTIMEDIA data transmission experience a number of constrains that result in low Quality of Service (QoS) that is offered to the end user. These constrains have mainly to do with the nature of multimedia applications which are characterized by three main properties: the demand for high data transmission rate (bandwidth-consuming applications), the sensitiveness to packet delays (latency and jitter) and the tolerance to packet losses (packet-loss tolerant applications), when compared to other kind of applications. As wireless communications and networking fast occupy centre stage in research and development activity in the area of communication networks, the suitability of the layered protocol architecture is coming under close scrutiny from the research community. Although layered architectures have served well for wired networks, they are not suitable for wireless networks. To illustrate this point, researchers usually present what they call a cross-layer design proposal. Thus, there have been a large number of cross-layer design proposals in literature recently. Generally speaking, cross-layer design refers to protocol design that is done by actively exploiting the dependence between protocol layers to obtain performance gains. This is unlike layering, where the protocols at the different layers are designed independently.

The main concept of this paper is the "holistic approach" in which all layers participate to the adaptation process and make its own contributions. We strongly believe that any cross-layer adaptation effort is incomplete when only some of the layers take part in the adaptation process because all layers belong to the same system. Further research will guide us to the most appropriate adaptation scheme, as many factors, which are involved, have to be investigated and evaluated through deeper studies and evaluation.

Up to now, there have been a number of proposals for improving QoS in multimedia applications through cross-layer adaptation. In [1] the need of a cross-layer optimization is examined and an adaptation framework is proposed amongst the application (APP), the Medium Access Control (MAC) and the Physical (PHY) layers. In [2] the issue of cross-layer design in wireless networks is addressed. The focus is on the way that higher layers share knowledge of the PHY and MAC layers conditions in order to provide efficient methods to allocate network resources over the Internet. In [3] a joined APP and MAC adaptation is proposed with the use of MPEG-4 and the latest Fine Granularity Scalability (FGS) extension. Signaling issues between the layers for cross-layer optimization over wireless networks are examined in [4]. In [5] a joined adaptation scheme of the APP, MAC and PHY layers is presented. Finally, reference [6] outlines the need for new cross-layer architecture to address known problems of mobility, packet losses and delay that are observed in wireless networks.

This paper presents the performance evaluation of the wired portion of a framework for cross-layer adaptation for multimedia transmission over wired and wireless networks, based on a "holistic approach". The rest of this paper is organized as follows. Our proposed framework is presented in section 2. Section 3 discusses the implementation issues of the proposed framework. Simulation results are presented in section 4. Conclusions and future work are discussed in

Manuscript received July 10, 2008.

Christos Bouras is with the Research Academic Computer Technology Institute and University of Patras, N. Kazantzaki Str., University of Patras, 26500 Rion, Greece (tel: +30-2610-960375, fax: +30-2610-960358, e-mail: bouras@cti.gr).

Apostolos Gkamas is with the Research Academic Computer Technology Institute, N. Kazantzaki Str., University of Patras, 26500 Rion, Greece (tel: +30-2610-960465, fax: +30-2610-960358, e-mail: gkamas@cti.gr).

Georgios Kioumourtzis is with the University of Patras, Computer Engineering and Informatics Department, 26500 Rion, Patras, Greece (tel: +30-2610-960316,fax:+30-2610-960358,e-mail: gkioumou@ceid.upatras.gr).

section 5.

II. Proposed Framework

In this section we describe the proposed framework for cross-layer adaptation. The proposed framework consists of four entities: The sender which represents the multimedia sender, the proxy which is located at the edge of the wired network, the AP which co-located with the proxy and can be integrated with the proxy and finally the wired and wireless receivers. This proposed framework separates the wired from the wireless part of the network by introducing a new entity named "proxy" between the sender and the receiver. The sender transmits multimedia data to the wired receivers and the proxy using the wired part of the network and the proxy is responsible to transmit the multimedia data received by the sender to the wireless receivers. For the transmission of multimedia data in the wired network multicast is used. Proxy entity is responsible for collecting wireless users requirements and sends them to the sender in order to perform the cross layer adaptation.

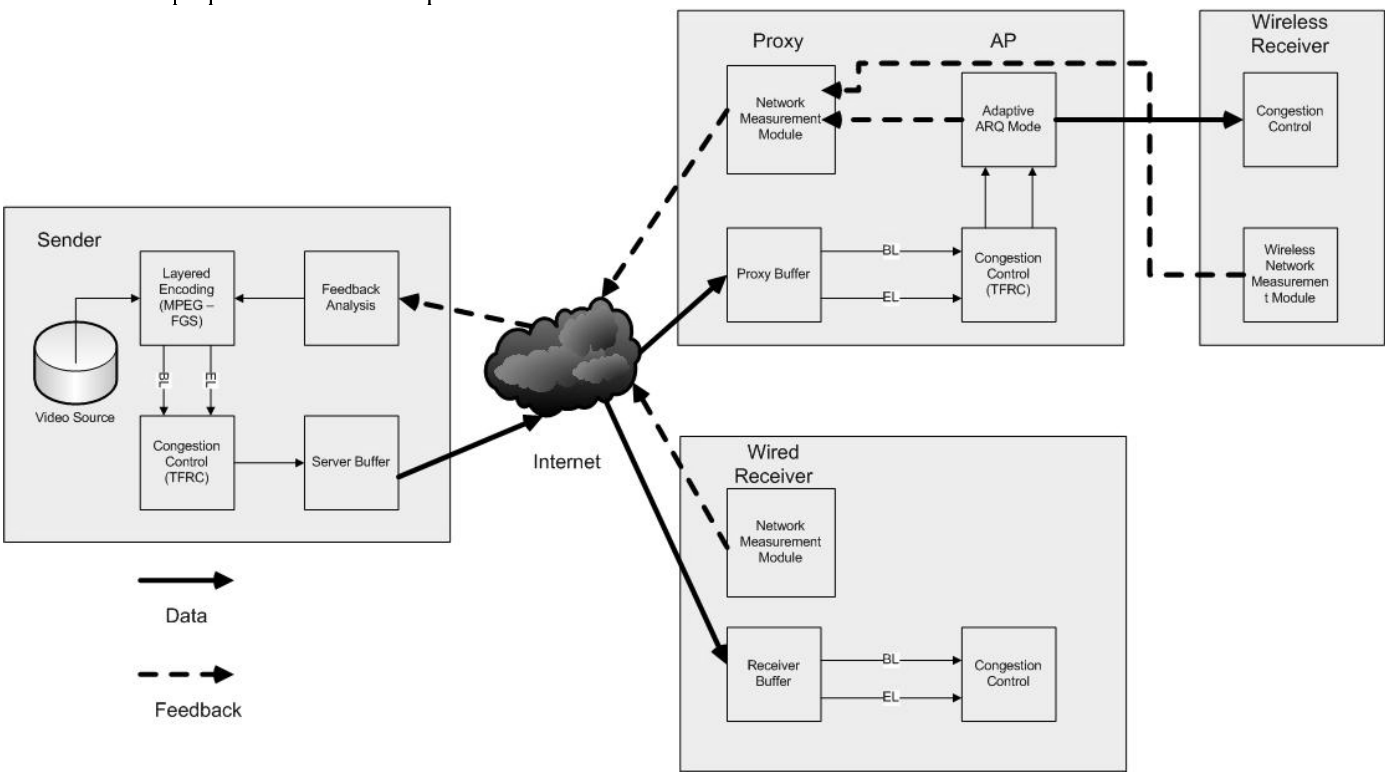

Fig. 1. Block diagram of the proposed framework

Figure 1 shows the block diagram of our proposed framework. The interested reader can find a detailed description of the functionality of the four entities of the proposed framework in [7].

A. Sender Entity

At the sender we distinguish four main functions: Feedback analysis, Layered encoding, Congestion control and taking advance of available (if they are available) network services (DiffServ or IntServ). The proposed framework is using the MPEG-4 protocol with the latest FGS enhancement [8] for video streaming between the sender, the receivers and the proxies for the following reasons: FGS encoding produces only two layers the base layer (BL) in which the bit rate is equal to or lower to the minimum bandwidth and the enhancement layer (EL) that consumes the remaining available bandwidth. The sum of BL and EL bit rates must be equal or less to the available bandwidth.

The sender generates two RTP sessions one for the BL and one for the EL and transmits each RTP session to different multicast groups. With the use of RTCP adaptive feedback mechanism the wired receivers and proxies send their feedback to the sender in the form of RTCP receiver reports. We have added an application specific part (APP) to the RTCP receiver reports, which are sent by the wired receivers, in order to include the wired receivers' estimation about the TCP friendly bandwidth share $r^i_{tcp\_BL}$ for BL and $r^i_{tcp\_EL}$ for EL, in the path between the wired receiver i and the sender. In addition, we have added an application specific part (APP) to the RTCP receiver reports, which are sent by the proxies, in order to include the proxies' j estimation about the available bandwidth r^j_w in the wireless network that are responsible for. In this paper we have modeled only one wireless node to study the effects of the wireless bandwidth capacity over the server transmission rates. The sender stores the last values of $r^i_{tcp\_BL}$ and $r^i_{tcp\_EL}$ from all the wired receivers and proxies and this information is used for the adjustment of the transmission rate of the BL (r_{BL}) and EL (r_{EL}), according to the following procedure:

$$
\begin{aligned}
& r_{BL} \leftarrow \min(r^1_{tcp\_BL}, \ldots, r^i_{tcp\_BL}) \\
& r_{EL} \leftarrow \min(r^1_{tcp\_EL}, \ldots, r^i_{tcp\_EL}) \\
& if\,(r_{BL} < r_{EL}) \\
& \quad diff \leftarrow r_{EL} - r_{BL} \\
& \quad r_{BL} \leftarrow r_{EL} \\
& \quad r_{EL} \leftarrow diff
\end{aligned}
\tag{1}
$$

Sender entity executes the above procedure when receives an RTCP receiver report either by a wired receiver or a proxy that contains an application specific part. The algorithm defines the transmission rate of BL and EL layers. If the estimated transmission rate of EL layer is higher than this of BL layer then the algorithm shifts the rates so that BL enjoys always higher transmission rates. The rest of the available bandwidth is allocated to EL layer. This algorithm implements the sender's congestion control mechanism.

A different approach is to try to adjust the BL transmission rate to a predefined targeted rate, depending on receivers' capacity and QoS criteria. When the channel capacity is higher than the predefined target rate for BL layer, the EL will consume the rest of the available bandwidth. We present this algorithm in equation (2).

$$
\begin{aligned}
& set\ (\text{target\_rate}) \\
& r_{BL} \leftarrow \min(r^1_{tcp\_BL}, \ldots, r^i_{tcp\_BL}) \\
& r_{EL} \leftarrow \min(r^1_{tcp\_EL}, \ldots, r^i_{tcp\_EL}) \\
& channel\_capacity \leftarrow r_{BL} + r_{EL} \\
& if\,(channel\_capacity >= \text{target\_rate}) \\
& \quad r_{BL} \leftarrow \text{target\_rate} \\
& \quad r_{EL} \leftarrow channel\_capacity - r_{BL}
\end{aligned}
\tag{2}
$$

We can see that when the transmission rates of BL and EL exceed the predefined target rate of BL layer, the algorithm adjusts the transmission rates of both layers so that BL is able to transmit at rates up to target rate. EL layer can also be transmitted at rates that fill in the gap between the BL targeted rate and the total channel capacity. Our main objective with this algorithm is to prevent fast oscillations of the BL layer, which may result to an infeasible adaptation rate by the Audio Video (AV) coder. In this way we are able to better adapt and harmonize the BL transmission rate with the AV coder. We will elaborate the behavior of this algorithm through simulation results in Section IV.

B. Wired Receiver Entity

Each wired receiver measures the characteristics of the path, which connects it with the sender and informs the sender with the use of receiver reports. The following parameters are measured:

Packet loss rate (l_i): The wired receiver calculates the packet loss rate of both layers (BL and EL) during the reception of packets with the use of RTP packets sequence numbers.

RTT estimations (t^{e-i}_{RTT}): The wired receiver makes estimation for the RTT between itself and the sender based on one-way delay measurements, with the use of RTP packets timestamps.

The wired receiver emulates the behavior of a TCP agent with the use of the analytical model of TCP and estimates a TCP friendly bandwidth share $r^i{}_{tcp}$ every RTT using the analytical model proposed in [9] for each layer ($r^i_{tcp\_BL}$ for BL and $r^i_{tcp\_EL}$ for EL). If the wired receiver experiences packet losses it will use the following equation to estimate a TCP friendly bandwidth share (in bytes/sec):

$$
r^i{}_{tcp} = \frac{P}{t^{e-i}_{RTT}\sqrt{\dfrac{2l_i}{3}} + 4t^{e-i}_{RTT}\min(1,3\sqrt{\dfrac{3l_i}{8}})l_i(1+32l_i^{\,2})} \tag{3}
$$

Where P is packet size in bytes, l is the packet loss rate and t_{RTT} is the Round Trip Time (RTT) of the TCP connection.

If the wired receiver does not experience packet losses, in order to estimate a TCP friendly bandwidth share r^i_{tcp}, the r^i_{tcp} must not be increased more than a packet / RTT. For this reason the wired receiver calculates the value of r^i_{tcp} with the following equation (in bytes/sec):

$$
r^i_{tcp} = r^i_{tcp} + \frac{1}{t^{e-i}_{RTT}} P \tag{4}
$$

Each wired receiver measures the packet loss rate based on RTP packets sequence numbers in both layers (BL and EL). In order to prevent a single spurious packet loss having an excessive effect on the packet loss estimation, wired receivers smooth the values of packet loss rate using the a filter (which has been presented and evaluated in [10]), which computes the weighted average of the m most recent loss rate values and provides a good estimation of the packet loss rate.

When a wired receiver i receives a RTP packet from a sender layer (BL or EL), it uses the algorithm presented in [7] in order to estimate the RTT between the sender and the wired receiver.

The wired receiver i feedbacks the sender with the values of $r^i_{tcp\_BL}$ and $r^i_{tcp\_EL}$, which stand for the wired receiver i TCP-friendly bandwidth estimations for BL and EL layers, respectively. The sum of these two streams is the total bandwidth estimation of wired receiver i.

Every time the wired receiver sends a receiver report to the sender, it includes the average value of $r^i_{tcp\_BL}$ and $r^i_{tcp\_EL}$ since the last receiver report.

C. *Proxy and Wireless Receiver entities*

Proxy and wireless receiver entities are designed "jointly" to clearly demonstrate the interactions between them. Proxy receives the multimedia data (voice-video) from the sender and stores BL and EL multimedia data into two different files. At this point the proxy has the ability either to transmit to wireless receiver only the BL, or to transmit both the BL and EL depending on wireless receiver link bandwidth estimation. In our proposed framework, we need the available bandwidth of the wireless link between the proxy and the receiver. Our proposed algorithm for wireless link bandwidth estimation is based on successful transmissions of MAC layer frames and is presented in [7]. Lastly, we need to mention that proxy entity acts also as a single wired receiver and performs all the actions described in the wired receiver section.

III. Implementation Issues

In this section we describe the extensions made to the RTP module in the network simulator ns2. We also present the benchmark that was used for the performance evaluation of our proposed framework.

A. *Modifications to the RTP code in ns2*

The ns2 code for the implementation of the RTP/RTCP [11] protocols is very generic and offers only the methods for the creation of the sessions between the sender and receivers. The RTCP sender and receiver reports do not provide any functionality except for the basic API for further development. In this work we extended the ns2 code to include the following:

- A *TCP friendly control-like mechanism* based on the analytical TCP model described in equation (3). This control mechanism has been integrated inside the existing RTP source code in *ns2* and makes use of the RTCP sender and receiver reports for the external signaling between the sender and wired receivers.
- A *coordination policy module,* expressed in equation (1), for the two MPEG-4 FGS layers (BL and EL). These two layers are transmitted by multicast streams with the same characteristics. Our module performs the co-coordinating functions between them. It ensures that the two streams do not antagonize, so that the transmission rate of BL is always higher than that of EL. We integrate also the second approach for the co-ordination of the BL and EL layers, which is expressed in equation (2).
- *Extensions of the RTP and RTCP packet headers* to include the fields for the external signaling between the sender and wired receivers.
- *Additional functionality* for the generation of the RTCP sender and receiver reports. With these extensions both sender and receivers are able to disseminate the necessary feedback information in their RTCP reports.

IV. Performance Evaluation in Wired Network

A. *Simulation environment and Network Topology set up*

The topology which was used for the evaluation of the proposed framework is a Local Area Network (LAN), which consists of one multimedia sender, six heterogeneous wired receivers and one wireless node. The heterogeneity of the receivers lays in the variation of the link capacity, which connects the receivers with the LAN. We have intentionally set up a "bottleneck" between routers 2 and 3 in order to create two different sets of wired receivers. The first set of receivers (Nodes 1, 2, 3 "fast receivers") is able to receive at higher bit rates than the second set (Nodes 4, 5, 6 "slow receivers"). The wireless node was added in order to observe the impact of the wireless medium uncertainties over the video transmission rates. We use the legacy PHY and MAC IEEE 802.11 protocols in the wireless node with rates up to 2Mb/s. In this paper we focus in wired networks and the simulation of networks with a larger number of wireless receives and different IEEE 802.11x protocols are left for future work. In the current scenario we will investigate how our modified RTP protocol reacts under the presence of a congested link in a wired network.

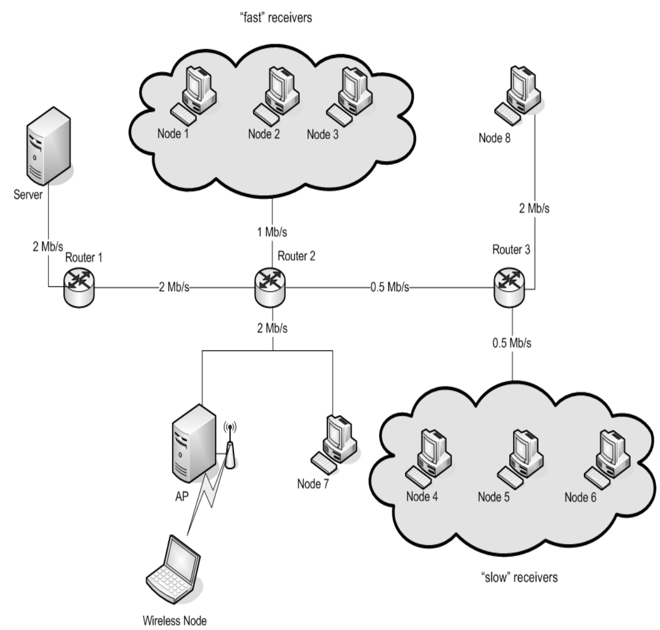

Fig. 2. Simulated network topology

The server transmits two different video streams in order to simulate the transmission of the BL and EL layers of a MPEG-4 FGS video streaming application. The initial rate for the BL was set to 256 Kb/s and for the EL to 64 Kb/s. All receivers join the two multicast streams and receive both the BL and EL MPEG-4 FGS. We run six different simulation sets to investigate:

- The transmission control policy that we pose between the BL and EL MPEG-4 FGS layers.
- The behavior of the proposed framework towards the TCP and UDP traffic.
- The effects of the wireless bandwidth estimation on the transmission rate of the multimedia server, and finally

• A comparison between our "light" TCP friendly algorithm and the TFRC code in ns2.

It has to be mentioned that the simulation and the comparison of our proposed framework with multi-stream (more than two streams) solutions is out of scope of this paper. Figure 2 depicts the network topology of the simulated scenarios.

B. First simulation: transmission of BL and EL in the absence of any coordination policy module

In this simulation the server transmits two multicast streams (BL and EL layers) at rates of 256Kb/s and 128 Kb/s respectively. We run the simulation for 200 seconds. In this initial set up we do not apply any coordination policy between the BL and the EL layers, except for the TCP friendly congestion control mechanism. The two streams operate independently and adjust its transmission rates based on TCP friendly bandwidth estimation criteria. In the chart presenting simulation results (Figure 3) we sum up the receiving rates of both layers (BL and EL) from two representing nodes of the two different groups (Node 1,"fast receiver" and Node 4, "slow receiver") and also the wireless Node.

One first observation is that the wireless node enjoys high receiving rates that are similar to those of Node 1 as expected due to the simulation topology. In addition, we expected these results as in our scenario we have included only one wireless node to transmit mainly the RTCP reports to the server. Our future work will focus mainly in the wireless part of our proposed network with a higher number of wireless receivers.

A second observation is the oscillations of the receiving rates. This is a direct result of the congestion control module and the lack of any co-ordination between the BL and the EL layers. The two layers operate independently and fail to co-operate. The high bit rates of one layer results to the elimination of the other layer. We expected these results as we do not employ any classification model in our network and RTP packets from both layers have the same "weight". A second reason for the cause of the observed oscillations is that the congestion control algorithm takes into account the estimated rates of the "slowest" receiver in order to adjust the server transmission rates.

The receivers translate the packet losses and the increased RTT as an indication of network congestion and recalculate the TCP friendly bandwidth share in accordance with the TCP analytical model which is used and presented in equation (3). When the server receives the RTCP receiver reports, it adopts its transmission rate accordingly. It has to be mentioned that we do not employ any "fairness" policy among the receivers for the RTP traffic transmission. Therefore, one single "slow" receiver defines the server's transmission rate. As a result, receivers with high receiving capacity enjoy only a portion of this capacity. However, our whole concept is to provide an acceptable transmission rate and quality to the set of receivers as a whole by including even those receivers with limited receiving capabilities. An alternative approach is to divide the EL into a set of sub layers and transmit each layer at different rates to different multicast groups, while keeping the transmission rate of BL constant.

From the above simulation results we conclude that having BL and EL layers being transmitted independently it will cause high oscillations. This is not the wished behavior of the designers of the MPEG-4 FGS, as BL layer transmission rates should be equal or less than the available receiving capacity of the set of the receivers. On the other hand, EL should provide a higher quality video stream to those receivers that are able to receive at higher bit rates. In the next simulation we will see how a coordination policy algorithm can increase the level of cooperation between the two layers and provide better results.

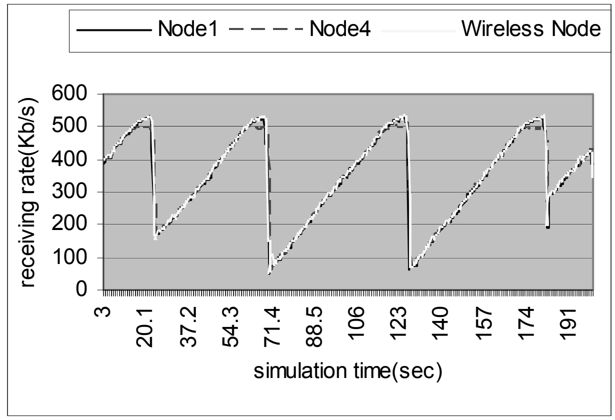

Fig. 3. First simulation results

C. Second simulation: transmission of BL and EL with a coordination policy module

In this simulation we employ the coordination policy module described in equation (1). The topology and the network attributes are the same with the previous simulation, so that we can obtain comparable results.

In this simulation BL always enjoys higher bit rates than EL, which now occupies the rest of the available bandwidth. EL is not allowed to exceed the transmission rates of BL layer in any case. In this way we have increased the level of co-operation between the two layers and ensured that in most cases the video streaming data is delivered to all receivers at a predefined rate. We observe from simulation results (Figure 4) that oscillations have been reduced when compared to previous simulation. There are certain areas (from 100 to 149 simulation seconds) in which EL has lower sending rates than BL. In those areas the two layers compete with each other. However, the existing oscillations may result to an infeasible adaptation rate by the Audio Video (AV) coder. Our conclusion is that when having multiple competing streams one should employ upper and lower thresholds to the basic stream (in our case this is the BL). In the next simulation we will apply the algorithm described in equation (2) in an effort to smooth these oscillations and provide more stable transmission rates for BL layer.

D. Third simulation: transmission of BL and EL with a predefined targeted rate for the BL layer

In this simulation we define a targeted transmission rate at 256 Kb/s for BL layer. This rate is used as the low threshold for our video transmission. However, it may increase the RTT time and as a result the network congestion in some links in the network.

From the simulation results we observe (Figure 5) better receiving rates for the three representing nodes. In most cases receiving rates are between 200 and 400 Kb/s. The lowest receiving rate drops down to 100 Kb/s, which is still acceptable. What is more, the packet loss ratio is very low giving us the feeling of a very good solution when one tries to increase the overall throughput by keeping low the packet loss ratio. It has to be mentioned that our TCP friendly bandwidth share algorithm is based on the RTCP reports sent by receivers. As we explained in the previous section we have extended the functionality of the RTP code in ns2 but we kept all other attributes of the protocol untouched. The computation of the RTCP transmission interval takes the session participants and the transmission rates into account in order to maintain scalability. Therefore, the RTCP transmission intervals of the sender and receiver reports are higher than the RTT time. The TCP analytical model, which is used, can benefit from the shot intervals of acknowledgments (ACK); one ACK is transmitted almost in every RTT. The impact in our modified RTP is that the server is not always up-to-date, concerning the current network conditions. We sacrifice the timely response of the sender in order to minimize bandwidth consumption of the RTCP sender and receiver reports. If we had set shorter RTCP intervals we would have consumed a higher portion of the available bandwidth to be used for the transmission of the two layers. The benefit would have resulted to more accurate computations of sender's transmission rates.

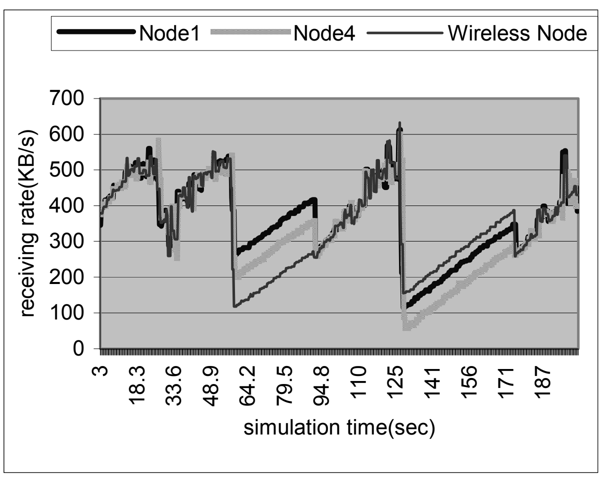

Fig. 5. Third simulation results

Another observation is that receivers do not have quick reactions when recovering from slow transmission rates. The reason behind it is the implementation of the "smooth loss rating" algorithm, which is presented in [7]. Under this scheme, receivers have "smoother" reactions because they use the smooth loss values to calculate the TCP friendly bandwidth share. This behavior is desirable for multimedia transmission comparing with the TCP-like oscillations of the transmission rates. Multimedia encoders however, perform better under smooth changes of the transmission rate.

It is also important to observe that "fast" receivers enjoy higher transmission rates than "slow" receivers and this is a result of the coordination function between BL and EL layers. This control allows slow receivers to receive at least BL whereas fast receivers receive both layers that provide higher quality.

E. Fourth simulation: transmission with background TCP traffic

In this simulation we add TCP background traffic across the network. Node 7 starts transmitting TCP traffic to Node 8 via the congested path from router 2 to router 3. In the simulation results we excluded the receiving rates of the wireless node to have a better readable chart. The wireless node has similar performance with the set of "fast" receivers.

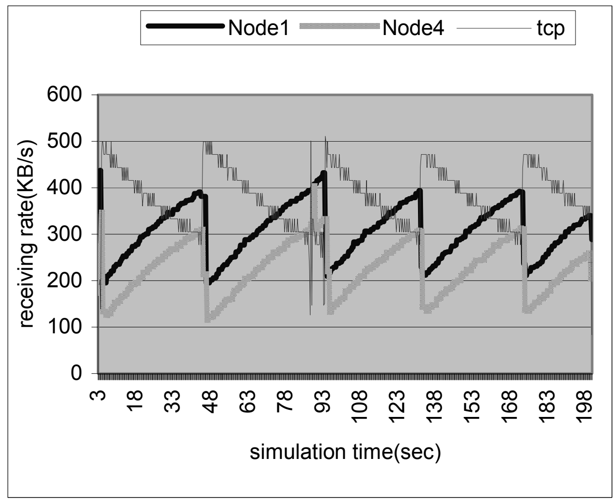

Fig. 6. Fourth simulation results

As we observe in the simulation results (Figure 6), the server in the event of TCP traffic backs off and reduces its transmission rates in such way that allows the TCP traffic passing through the network. The above simulation results confirm the TCP friendly behavior of our modified RTP protocol. Our model is more than TCP friendly that it ought to be. As we have already explained we try to integrate TCP friendly behavior in the RTP implementation in a "multi-session" environment based on RTCP receiver reports. The result is that TCP traffic responds faster than our protocol in a congested network.

However, we confirm that in the event of competing traffic the server transmits the MPEG-4 FGS layers to the wired receivers at rates that provide acceptable receiving rates of the video stream in terms of QoS. We also observe that Node 1, which belongs to the set of "fast receivers", has higher receiving rates than Node 4 ("slow receiver"). The packet loss ratio (not shown again in the charts in the shake of the economy of space) is still low. Therefore, it is accurate to say that our RTP/RTCP implementation provides at least an acceptable solution for multicast video broadcasting, in terms of TCP friendly behavior providing also high transmission rates with low packet losses.

F. Fifth simulation: transmission with background UDP traffic

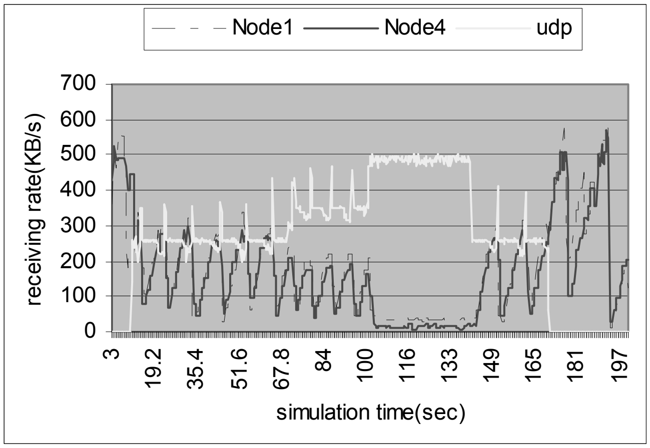

Fig. 7. Fifth simulation results

In this simulation we add UDP background traffic to study how our proposed RTP modification behaves in the light of competing UDP traffic. This simulation scenario is more challenging for the RTP video stream transmission, as the UDP traffic does not employ any congestion control mechanism. In the beginning of the simulation the UDP traffic is transmitted across the network from Node 7 to Node 8, at a rate of 256 Kb/s. As we observe in the simulation results (Figure 7), the server reduces its transmission rate due to the congested path from route 2 to route 3. However, the receiving rates of Node 1 and Node 4 are between 100 and 200 Kb/s. Therefore, we need to create more challenging conditions. At the 70th simulation second we increase the transmission rate of the UDP traffic up to 350 Kb/s. The server backs-off as the packet loss rate and the RTT time have been increased. However, receivers are still able to receive video transmission this time at rates lower than 100 Kb/s. At the 100th simulation second we further increased the UDP traffic up to 512 Kb/s and the impact now on the streaming video transmission is severe. UDP traffic occupies almost all the available bandwidth and causes the sender to reduce its transmission rate to very low values as receivers experience high number of packet losses. At the 140th simulation second, we decrease the UDP down to 256 Kb/s and we observe that the receivers increase gradually its receiving rate. The video streaming occupies again almost all the available bandwidth. We conclude that our RTP/RTCP implementation can cope with UDP traffic as long as the UDP traffic is transmitted at similar rates to those of the RTP rates. When UDP traffic is transmitted at very high rates our proposed framework will experience very low performance.

G. Sixth simulation: comparison with TFRC implementation in ns2.

In the last simulation we compare the TFRC implementation in ns2 against our modified RTP/RTCP implementation with the integrated TCP friendly congestion control. The TFRC code in ns2 has been used for simulation throughout a number of researchers and provides an acceptable implementation of the TFRC specifications.

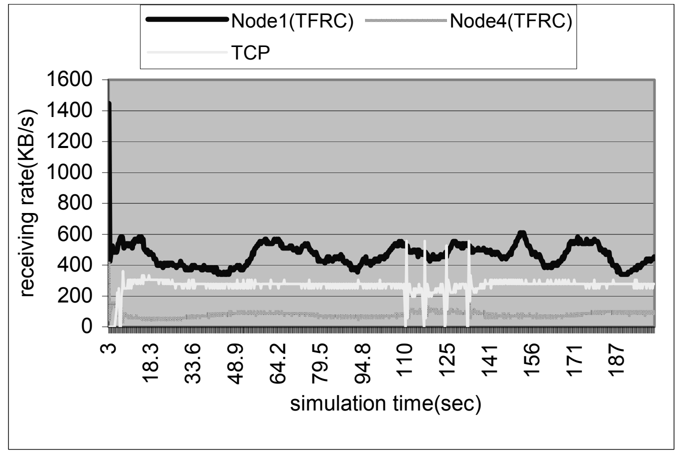

Fig. 8. Sixth simulation results

The simulation scenario has exactly the same network attributes as our previous simulations to achieve a fair comparison. In this case, RTP traffic is transmitted to the same set of receivers and the congestion control is left to TFRC protocol. We transmit also the same TCP traffic across the network from Node 7 to Node 8.

We observe from the simulation results (Figure 8) that the ns2 TFRC implementation has "smoother" oscillations than our implementation and this is a desired attribute especially for video transmission. TFRC presents indeed TCP-friendly behavior except for some cases in which TCP is severely affected and drops to zero transmission rates. In our implementation the TCP traffic has always higher transmission rates.

A second observation is that although Node 1 ("fast receiver") enjoys high receiving rates, Node 4 ("slow receiver") has very low receiving rates. None can blame TFRC for this performance as each receiving node adjusts its TCP friendly bandwidth share estimation individually. As a result, Node 4 estimates very low transmission rates as a result of the highly congested network. In our implementation we have added the "coordination module" that takes into account all information received by receivers and adjusts the transmission rates of the BL and EL layers accordingly.

Our final conclusion is that our RTP/RTCP implementation introduces very good characteristics when we have multiple video streams transmitted under congested network conditions. More extensions have to be added to smooth the observed oscillations. However, the code and the implementation complexity are very low when compared to TFRC module in ns2.

V. Conclusions / Future work

In this paper we evaluate the performance of the wired portion of a framework for cross-layer adaptation for multimedia transmission over wired and wireless networks. We extended the functionality of RTP protocol to enhance it with TCP friendly characteristics. We investigated the behavior of the proposed framework through a number of simulations with a group of heterogeneous wired receivers. The performance evaluation included various simulation scenarios in order to investigate the behavior of the proposed framework under different conditions. The simulations conducted with the ns2 simulation software, which has been extended to support the functionality of the RTP protocol with TCP friendly congestion control. Simulation results endorse our design goals for the transmission of video streaming data with the MPEG-4 FGS application protocol. Our future work includes the overall evaluation of the proposed framework to a heterogeneous group of wired and wireless receivers with focus on the wireless domain.

REFERENCES

[1] M. van der Schaar, D. Sai Shankar N, "Cross-layer wireless multimedia transmission: Challenges, principles and new paradigms, IEEE wireless Communications, August 2005

[2] Shakkottai, S. Rappaport, T.S. Karlsson, P.C. "Cross-layer design for wireless Networks", IEEE Communication magazine, Volume: 41, Issue: 10, Oct 2003.

[3] M. van der Schaar, S. Krishnamachari, S. Choi "Adaptive Cross-Layer Protection Strategies for Robust Scalable Video Transmission Over 802.11 WLANs", IEEE Journal on selected areas in communications, vol 21, no.10 December 2003.

[4] Qi Wang; Abu-Rgheff, M.A., "Cross-Layer Signalling for Next-Generation. Wireless Systems", IEEE WCNC 2003, vol. 2, Mar. 2003, pp. 1084–89.

[5] Jie Chen, Tiejun Lv and Haitao Zheng. "Cross-layer design for QoS wireless communications", Proceedings of the 2004 International Symposium on Circuits and Systems, 2004. ISCAS '04, Volume: 2, On page(s): II- 217-20 Vol.2, May 2004.

[6] Carneiro, G. Ruela, J. Ricardo, M, "Cross-layer design in 4G wireless terminals", IEEE Wireless Communications, Volume: 11, Issue: 2, page(s): 7- 13, Apr 2004.

[7] C. Bouras, A. Gkamas, G. Kioumourtzis, "A Framework for Cross Layer Adaptation for Multimedia Transmission over Wired and Wireless Networks", The 2007 International Conference on Internet Computing (ICOMP'07), Las Vegas, Nevada, USA, 25 - 28 June 2007.

[8] W. Li, "Overview of Fine Granularity Scalability in MPEG-4 Video Standard," IEEE Trans on Circuits and Systems for Video Technology, Vol. 11, No. 3, March, 2001, pp.301--317.

[9] J. Pandhye, J. Kurose, D. Towsley, R. Koodli, "A model based TCP-friendly rate control protocol", Proc. International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV), Basking Ridge, NJ, June 1999.

[10] L. Vicisiano, L. Rizzo, J. Crowcroft, "TCP - like congestion control for layered multicast data transfer", in IEEE INFOCOM, March 1998, pp. 996 - 1003.

[11] RFC 3550, RTP: A Transport Protocol for Real-Time Applications, H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, July 2003

Christos Bouras obtained his Diploma and PhD from the Department Of Computer Engineering and Informatics of Patras University (Greece). He is currently an Associate Professor in the above department. Also he is a scientific advisor of Research Unit 6 in Research Academic Computer Technology Institute (CTI), Patras, Greece. His research interests include Analysis of Performance of Networking and Computer Systems, Computer Networks and Protocols, Telematics and New Services, QoS and Pricing for Networks and Services, e-Learning Networked Virtual Environments and WWW Issues. He has extended professional experience in Design and Analysis of Networks, Protocols, Telematics and New Services. He has published 250 papers in various well-known refereed conferences and journals. He is a co-author of 8 books in Greek. He has been a PC member and referee in various international journals and conferences. He has participated in R&D projects such as RACE, ESPRIT, TELEMATICS, EDUCATIONAL MULTIMEDIA, ISPO, EMPLOYMENT, ADAPT, STRIDE, EUROFORM, IST, GROWTH and others. Also he is member of experts in the Greek Research and Technology Network (GRNET), Advisory Committee Member to the World Wide Web Consortium (W3C), IEEE - CS Technical Committee on Learning Technologies, IEEE ComSoc Radio Communications Committee, IASTED Technical Committee on Education WG6.4 Internet Applications Engineering of IFIP, ACM, IEEE, EDEN, AACE, New York Academy of Sciences and Technical Chamber of Greece.

Apostolos Gkamas obtained his Diploma, Master Degree and Ph.D from the Computer Engineering and Informatics Department of Patras University (Greece). He is currently an R&D Computer Engineer at the Research Unit 6 of the Research Academic Computer Technology Institute, Patras, Greece. He is also visiting lecturer in the Telecommunication Science and Technology Department of University of Peloponnesus. His research interests include Computer Networks, Telematics, Distributed Systems, Multimedia and Hypermedia. More particular he is engaged in transmission of multimedia data over networks and multicast congestion control. He has published more than 40 papers in international Journals and well-known refereed conferences. He is also co-author of three books (one with subject Multimedia and Computer Networks one with subject Special Network Issues and one with subject IPv6). He has participated in various R&D project (in both EU and national) such as IST, FP6, eLearning, PENED, EPEAEK, Information Society.

Georgios Kioumourtzis is a PhD candidate in the Computer Engineering and Informatics Department at the University of Patras, Greece. He has received his B.S from the Hellenic Military Academy and graduated also from the School of Telecommunications for Signal Officers. In 2005, he received the Master of Science in System's Engineering and the Master of Science in Computer Science from the Naval Postgraduate School California, USA. His thesis work was related to Mobile AdHoc Wireless Networks (MANETs). His current research interests include Multimedia transmission over heterogeneous networks, transmission protocols, cross-layer optimization in wireless networks and IEEE 802.11x technology