

A Network-Driven Architecture for the Multicast Delivery of Layered Video and A Comparative Study[†]

Panagiotis Stathopoulos, Voula Zoi, Dimitrios Loukatos, Lambros Sarakis,
Nicolas Mitrou

National Technical University of Athens, ECE Department, Zografou, Greece
{psthath,vzoi,dlouka,ls6}@telecom.ntua.gr, mitrou@softlab.ntua.gr

Abstract: Multicast multimedia distribution, despite the benefits it offers, features a number of unresolved issues, such as network and users heterogeneity. Layered stream organization has been considered as suitable to cope with these issues. In this paper we, initially, study proposed mechanisms for the multicast delivery of layered video streams, highlighting their principles and functionality. Most of them are suited for Best Effort networks, resulting in increased application complexity or sub-optimal performance. Based on the assumption that Quality of Service (*QoS*) will be probably deployed to the future Internet, we propose a simple, network-based architecture for the multicast delivery of layered video streams. The Differentiated Services (*DiffServ*) framework and content-aware packet marking are exploited in this architecture. We discuss, how the application defines priority layers within the video stream, while *DiffServ* copes with layer-aware packet dropping. Each video stream is delivered to a single multicast group and no application-level complexity is required. During congestion in nodes of the multicast tree, packets corresponding to lower priority layers are dropped first, thus protecting the most valuable video stream entities. Protocol extensions for coping with user heterogeneity and the problem of congestion from undelivered packets are also proposed. Finally, a qualitative comparison of all the aforementioned schemes is attempted.

Key words: Multicast, layered video, network-driven, content-aware packet marking, *DiffServ*.

[†] This work was partially funded by the IST-2001-33093 CREDO project.

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: [10.1007/978-0-387-35616-7_23](https://doi.org/10.1007/978-0-387-35616-7_23)

W. Cellary et al. (eds.), *Internet Technologies, Applications and Societal Impact*
© IFIP International Federation for Information Processing 2002

1. INTRODUCTION

Forthcoming media distribution will be, to a great extent, multicast to a variety of access networks and terminal equipments. Despite the benefits, multicast deployment is limited due to a number of problems such as: address manageability, routing and congestion control across heterogeneous network branches, amongst others. Congestion control [1] has been an early Internet problem, which has led to the incorporation of congestion avoidance and control mechanisms to the TCP protocol. On the other hand, although the UDP protocol does not inherently employ congestion avoidance mechanisms, unicast UDP streams can achieve congestion responsive behaviour utilizing the RTP/RTCP protocol stack, in conjunction with rate adaptation mechanisms [2]. However, in the case of multicast streams the same mechanisms are not trivially applicable, due to the multiplicity of the receivers and the heterogeneity of network resources across the different multicast branches.

Therefore, the issue of congestion control in a multicast environment has raised a significant research interest and several mechanisms, best suited for Best Effort networks have been proposed. Some of them are built on the application level, others shift some of the complexity towards the network, requiring network enhancements, while some recently proposed totally rely on custom network mechanisms. However, it is quite probable that the future Internet will offer services beyond the present simple BE service. The Differentiated Services (*DiffServ*) framework [3], as one of the more promising and feasible means for providing Quality of Service (*QoS*) in IP networks, can be exploited to enable multicast layered video delivery, without the need for complex application-level functions.

In this paper a survey of the most considerable paradigms for multicast video delivery is conducted. The most important principles of their operation are described. Then, a simple, open loop network-based architecture exploiting the DiffServ framework and content-aware packet marking is proposed. Network protocols extensions, for coping with the problem of congestion from undelivered packets is also proposed. A comparative evaluation of all these mechanisms is presented, providing insight on the benefits and possible constraints of each of the proposed solutions.

The rest of the paper is organized as follows: In Section 2 the most considerable mechanisms for multicast video delivery are described. In Section 3 the proposed DiffServ motivated architecture is presented. In Section 4 a qualitative comparison and evaluation of these mechanisms is attempted. Finally, in Section 5 concluding remarks and directions for further research are provided.

2. MECHANISMS FOR THE MULTICAST DELIVERY OF VIDEO STREAMS

Mechanisms for the multicast delivery of video streams can be, classified as *receiver-driven*, *source-driven*, *network-driven* or *hybrid*, according to the party involved in the congestion control procedure. In the next subsections the most considerable examples of each category are described.

2.1 Receiver-driven Mechanisms

The first mechanism of this category, proposed in [4], is the Receiver-driven Layered Multicast protocol (*RLM*). A layered encoded video stream is assumed, with each layer transmitted on a different multicast group. Receivers subscribe to a number of layers, depending on the level of congestion. Rate adaptation decisions are taken by each receiver, by leaving multicast groups (which corresponds to dropping layers) when congestion is noticed, or joining multicast groups (which corresponds to adding layers), when the network capacity is sufficient. Congestion is estimated based on the experienced loss rate. On the other hand there is no direct way for a receiver to be aware of the available resources. The solution of periodic join attempts, combined with a learning algorithm for controlling their frequency is adopted.

A number of enhancements over the RLM have been proposed, in order to overcome certain constrains of the initial protocol (see section 4). In [5] TCP friendliness and improved fairness are incorporated, by emulating TCP behaviour. In order to achieve an effective bandwidth usage, receivers behind the same bottleneck are synchronized, by joining a layer only at synchronization points (SP). SPs in higher layers are exponentially less frequent than in lower layers. In this way, a receiver that has subscribed to a small number of layers is likely to catch up with receivers with a higher subscription level, thus improving the overall fairness of the protocol, in comparison with RLM.

The Fair Layered Increase/Decrease with Dynamic Layering (*FLID-DL*) proposed in [6] enhances the RLC by using Dynamic Layering. The rate of each layer decreases over time and this is achieved exploiting a Digital Fountain encoder [7] at the source. In the Digital Fountain encoding scheme the delivery of specific packets is not required, instead a number of arbitrary packets can be independently decoded. Dynamic Layering reduces the leave latencies associated with dropping a layer, since the receiver has to periodically join additional layers to maintain a non-decreasing rate. FLID retains the sender-initiated synchronization points used to coordinate

receivers introduced by RLC, but uses probabilistic increase signals instead of injecting periodic bursts.

An alternative approach proposed to refrain from RLM join experiments is to calculate the TCP-friendly bandwidth at the receivers, and subscribe to the corresponding layers. Receivers simply adjust their subscription level to the rate given by the equation that models the throughput of an equivalent TCP session (The Layered Transmission Scheme (*LTS*), proposed in [8] and the TCP-Friendly Transport Protocol (*TFRP*), proposed in [9]).

2.2 Hybrid Source/Receiver Mechanisms

Source driven mechanisms try to accommodate all receivers with a single rate stream. The source determines this rate, by aggregating receiver feedbacks values. There is a trade off, in choosing the single rate that optimally satisfies the majority of the receivers. Moreover, feedback aggregation and processing do not scale well, especially as the number of receivers increases.

Due to the insufficiency of source driven mechanisms to handle multicast video delivery, hybrid receiver/source mechanisms have been proposed. In the Multicast Loss-Delay Based Adaptation Algorithm (MLDA) [10] the major enhancement against the other receiver-driven mechanisms, is that the sender continuously adjusts the bandwidth distribution of the layers to best support receivers' requirements. Independently, the receivers adjust their subscription level to the appropriate rate. Similar approaches, resulting in an enhanced Signal to Noise Ratio have been proposed in [11], featuring feedback merging at intermediate network nodes.

2.3 Network-driven mechanisms

Recent publications have also acknowledged the value of network-driven approaches. In [12] a network-based congestion control mechanism for layered multimedia multicast delivery is proposed (Network-driven Layered Multicast: *NLM*). The Time to live (TTL) threshold in each link, together with the TTL and Type of Service (*ToS*) bits in the IP header of each packet are used. The ToS value characterizes traffic belonging to instances of this protocol, while each layer is assigned a different TTL value; high priority layers are assigned the highest TTL value. The main concept is that the intermediate routers within the multicast tree dynamically adjust the interface's TTL value according to the length of the output queue, which is used as an estimate of the congestion level, therefore changing the number of layers transmitted. Each router collaborates with neighbouring routers in

order to distribute the estimated level of congestion at the multicast tree. In this way, no traffic control is required by the receivers, while the network responds quickly to congestion. The authors compare this mechanism with the RLM through simulations and find that it achieves better fairness and service stability.

2.4 Hybrid Receiver/Network Mechanisms

Hybrid receiver/network mechanisms combine application-level mechanisms and protocols with network-based functions in order to improve service stability and fairness against the plain RLM, as well as responsiveness to short-term congestion. Such a hybrid approach, based on the RLM concept and network level mechanisms has been proposed in [13]. Receiver-driven Layered Multicast with Priorities (*RLMP*) attempts to combine the advantages of the RLM protocol and priority dropping. Two-drop priorities are assumed, with congested routers dropping packets of groups with low priority first. As in RLM, receivers decide which groups to subscribe to, but in this case they also choose what priority to give to each layer. If they mark all layers as high priority, then during congestion packet drops will be experienced by all layers, including the base layers, therefore resulting in severe quality degradation. In this way, incentives are given to receivers to mark as high priority only those layers that fit the available resources, while the low priority layers absorb network congestion.

3. A DIFFSERV-MOTIVATED ARCHITECTURE FOR LAYERED MULTICAST DELIVERY

Network mechanisms, may achieve rapid response to congestion and higher bandwidth utilization without complex application level functions. The emergence of DiffServ enables traffic control, within a complete and feasible QoS framework. Multicast delivery of video can benefit from such a framework. In this section, a DiffServ motivated architecture, the Differentiated Services Layered Multicast, (*DSL*M), is proposed for the multicast delivery of layered video. While the DSLM core architecture does not require any protocol extensions, these can be applicable in order to better cope with the congestion from undelivered packets issue. A mechanism based on such extensions is defined in subsection 3.3.

3.1 The Core DSLM Architecture

The main idea of DSLM is that once the application flow provides indications of how to scale in cases of losses, in order for the remaining data to be meaningful, it is the task of the network to manage packet drops according to their importance. Within this architecture, each video stream is delivered on a single multicast group over a DiffServ domain. The different drop precedence of the Assured Forwarding (AF) [14] Per Hop Behavior (PHB) are exploited in order to introduce multiple layers of importance within the same multicast stream. These layers, which are based on the coding properties, are defined by the source and aim to protect the most important entities of the stream. A single multicast tree is maintained and congestion at the nodes of each tree is managed locally at each node by dropping packets corresponding to entities of lower priority. In this manner, congestion is absorbed by the lowest priority layers, while it is dealt with at the node it occurs, as soon as it occurs.

As discussed in [15], priority-dropping schemes should provide incentives to applications to limit their rate and avoid classifying all packets to the highest priority. The DiffServ framework inherently provides such incentives by introducing the Service Level Agreements (SLAs) and per flow policing at the ingress boundary of the DiffServ domain. The relative rate allocated to each multicast group depends on the SLA, while fairness of the multicast streams against each other and against TCP applications is achieved by properly configuring the DiffServ network and according to the SLA parameters [16].

Within this architecture there is no need for maintaining multiple multicast groups per stream, for synchronisation among group members or for awareness of other multicast groups behaviour. Moreover, layered encoding of the video stream is not necessary; instead temporal dependencies within the stream can be exploited in order to define priority layers, as described in subsection 3.2. The response time to congestion is at the time scale of the congestion event. Therefore, this architecture can successfully cope with short-term congestion, which is produced by variable bit rate video streams exhibiting significant burstiness in small time scales. In this manner, a fine grain rate adaptation, together with high exploitation of available bandwidth is achieved.

An instance of the mechanism's operation is illustrated in Figure 1. Each packet is assigned a different priority (AF11, AF12, AF13) by the application, while it is policed to conform to the SLA by the ingress policer P. In every congested router's (R) interface the packets of the lowest priority layers are dropped (D) according to the congestion level of each node. Thus, the end host (H) receives packets of high priorities. In the example shown in

Figure 1 an AF13 packet (packet 4) is dropped at R1’s interface to R2, due to competitive traffic, since it is marked with the lowest priority. Between R1 and H1 an AF12 packet 2 is dropped, due to a higher level of traffic congestion, since layer AF13 is exhausted. AF11 class packets are however still protected. The same pattern applies to all other nodes.

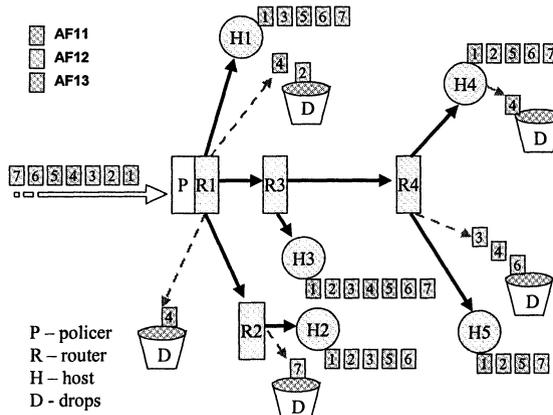


Figure 1. The priority dropping mechanism

3.2 Packet marking policies

One key characteristic of the proposed architecture is that it is flexible to accommodate various packet marking policies, according to the application’s needs, determined by the encoding properties of the stream. Different priorities to entities within a stream, according to loss and delay sensitivities, can be exploited instead of strict layered organization, or complex coding procedures. In [17] alternative ways for assigning priorities to entities within an MPEG video stream are presented. In MPEG-4 video streams, an object-based prioritisation policy could be adopted [18]. DSLM is also applicable for audio streams, exploiting unicast techniques for audio streams packet marking, presented amongst others in [19], [20].

3.3 Extensions for handling congestion from undelivered packets

Although simple, the DSLM core architecture cannot resolve the congestion from undelivered packets issue [1][15]. In greater detail, congestion from undelivered packets may arise when intermediate routers forward packets that will be later dropped, thus depleting resources from other flows. This can occur due to devices heterogeneity (e.g. in Figure 1. packet 4 is dropped on H4 since it cannot decode the corresponding layer), or due to network resources heterogeneity on downstream routers (e.g. in Figure 1, R3 forwards to R4 packets 4 which is dropped on the downstream router R4). In some cases, and especially in large-scale networks, this may result in wasting a significant amount of resources. In order to resolve this, downstream network congestion information should be propagated upstream to the source routers of the multicast tree, forcing them to drop as early as possible packets that would be anyway dropped later. In this subsection a set of functions, enhancing the DSLM core towards this, are defined.

In greater detail, a Congestion Information (CI) monitoring mechanism, a protocol for propagating such information the, Congestion Information Propagation protocol, (CIP) and additional traffic control functions are described. The routers' queue length could provide an indication of the overall congestion level. Such an approach is proposed in NLM. However, a DiffServ-oriented approach suitable for DSLM, is to monitor congestion level, in every PHB drop precedence, DSLM packets are classified to. Congestion, on a given drop precedence, can be indicated by its packet drop probability. In order to avoid bursts of drops affect the stability of the system, an exponentially weighted moving average is included:

$$DP_{AVG(AFij)} = W_{DR} * DP_{CUR(AFij)} + (1-W_{DR}) * DP_{AVG(AFij)}$$

$DP_{AVG(AFij)}$ is the average packet drop probability experienced for the drop precedence j of the AF_i PHB, $DP_{CUR(AFij)}$ is the calculated drop probability on the current time window and W_{DR} is the weight. Since this mechanism is used mainly to discover network heterogeneity and not short-term congestion, the frequency for calculating a new drop probability average is expected to be in the order of 10 to 30 seconds. Congestion Information (CI) propagated to upstream routers, is the Drop Probability Vector (DPV),

$$DPV = (DP_{AVG(AF1)}, DP_{AVG(AF2)}, DP_{AVG(AF13)})$$

assuming that DSLM packets are classified to the AF1 class. While this information is propagated for each multicast group, the DPV calculation is performed on a per AF drop precedence and not on a per-flow basis. The CI propagation protocol (*CIP*) could be built as extension to standard routing protocols, as proposed in RLMP or as special ICMP messages as proposed in NLM. In this paper we will not go further into details on the CIP protocol, however due to the infrequency of the estimation the overhead is expected to be low. The DPV is aggregated on routers so that each router is choosing the minimum drop probability for each multicast group and drop precedence for the output ports this group is replicated.

The forwarding state on multicast routers is in the form of (*S*, *G*, *Ifacelist*), where *S* is the source, *G* is the multicast group and *Ifacelist* is the the list of interfaces the group’s packets are replicated at. In this enhanced DSLM, due to the CIP protocol, the *Ifacelist* includes, apart from the router’s interface number, the aggregated DPV of the downstream routers. This corresponds to a minor memory overhead (3 additional bytes for each DSLM multicast group). The packet classification and forwarding procedure changes, since prior to classifying a DSLM packet to a PHB according to the ToS byte value, a packet drop or ToS remarking action is performed based on the collected downstream CI, effectively the corresponding DPV value. According to this classification a packet is either dropped or remarked to a higher drop probability. In table 1 some indicative values, elaborating this mechanism, are given.

	Example 1		Example 2	
ToS	DPV $DP_{AVG(AF1j)}$ $j=1,2,3$ (%)	Action (remarking or drop)	DPV $DP_{AVG(AF1j)}$ $j=1,2,3$ (%)	Action (remarking or drop)
AF11	4	AF11	0	AF11
AF12	35	DROPPED	5	AF12
AF13	60	DROPPED	40	DROPPED

Table1 Packet dropping based on downstream router congestion information

In this table two examples are illustrated. In the first, AF13 packets are facing heavy congestion, while AF12 packets are facing moderate congestion, on downstream routers. Since most AF13 packets will finally be dropped, thus making the corresponding layer useless by most video decoders, this layer is dropped from all upstream routers, thus making resources available to other flows. In the moderate congestion example 2, only the AF13 layer is dropped on intermediate routers. Exact thresholds or

policies for remarking or dropping are not given, since this paper focuses on the overall architecture and not on specific quantitative values. While in this approach a number of per flow traffic control is employed in core DiffServ routers, this is compatible to the multicast model for DiffServ currently supported, which stores per multicast group state.

4. EVALUATION AND QUALITATIVE COMPARISON

In this section, the aforementioned mechanisms, which are suitable for the multicast delivery of layered video, are evaluated. Although a strict quantitative comparison is not performed, we initiate from already available experimental data, as well as from the high-level mechanisms specifications. This comparison provides an insight, about each mechanism performance, characteristics, and suitability under different conditions.

In principle, receiver-driven mechanisms are suited for BE network environments, exploiting mechanisms exclusively based on the application layer. Such schemes scale well to large receiver sets, in terms of bandwidth utilization, in comparison to single rate approaches. However, the initial RLM suffered also from a number of constraints. A trade off exists between the number of layers, the extra complexity introduced and the bandwidth utilization achieved. A small, number of layers leads to a coarse rate adaptation and low bandwidth utilization, while a large number of layers to extra complexity in multicast groups management. Other constraints concern the degree of responsiveness to congestion and the service stability, impacted by the latency of the control mechanisms and by transitions among layers, correspondingly. Moreover, it does not provide a TCP-friendly behaviour while it may result in unfair distribution of bandwidth among concurrent RLM sessions, with the earlier started sessions being favoured. Failed join experiments may create additional congestion, during transition periods. Finally, in order for layered schemes to be efficient, it is imperative that receivers behind the same bottleneck synchronize their join and leave decisions. Several protocols have been developed that enhance the original concept of RLM, resolving a subset these constraints.

In RLC TCP friendliness and fairness among sessions of the same protocol are resolved. However, the saw-tooth like behaviour may not well suit the needs of multimedia streaming applications. On the other hand, the FLID-DL protocol is a considerable improvement over RLC and RLM, in the issue of long leave latencies, while being more flexible on the rate distribution of the layers. However, it also results in increased overhead for

the underlying multicast routing protocol, as join and leave decisions occur much more frequently, while special video encoding is needed. The LTS and the TFRP, while avoiding join experiments and being easy to implement, require measurements of the round-trip time in a scalable manner. Moreover, to prevent rate oscillations, it is necessary to accurately measure and smooth loss and round-trip time values through filtering.

The MLDA hybrid source/receiver approach overcome the trade off between the number of layers and the rate adaptation flexibility, reduces join/leave messages traffic while improving the responsiveness to congestion and the service stability by dynamically optimising the rate of each layer. A major disadvantage of MLDA is the complexity of the protocol and the added complexity of the source application that has to distribute the data onto the dynamic layers.

The hybrid receiver/network RLMP approach considers complementary network mechanisms for enhancing the RLM protocol with the advantages of priority dropping. However, the proposed framework for priority dropping is custom and limited to the scope of this mechanism. Extensions to multicast routing protocols, defining the priority of each group are required, while additional per flow state needs to be preserved in routers.

NLM, is based on network only support, with no need for any application-level control. It can offer high bandwidth utilization and responsiveness to congestion, since the applications do not need to collect information about the network's congestion level. On the other hand, NLM uses the TTL decrement function as a congestion control mechanism, which is not its original purpose. The TTL initial value and decrement algorithm needs careful tuning, which is custom for every IP sub-network, for avoiding multicast scope problems. Moreover, fairness is heavily determined by the initial TTL values and its decrement policy definition. Additionally, a protocol for the distribution of congestion information among routers is required.

In table 2 the most important properties of these mechanisms are summarized. In greater detail, mechanisms are characterized by the party (Receiver, Sender, Network) involved (Rcv, Src, Net correspondingly) with RLM performance taken as reference. Y/N correspond to Yes or No, M corresponds to a modest performance, while the NCD indication corresponds to Network Configuration Dependent characteristics. Mechanisms that significantly outperform RLM in the fields of, fairness, TCP friendliness, responsiveness to congestion, are marked as >>RLM, while protocols that are modestly improving RLM as >RLM. Protocols with similar performance to RLM are marked as ≈RLM. The intensity and frequency of the group

management functions required are also compared to the ones of RLM. The fewer they are, the more favourably they are evaluated.

The FLID-DL and the MLDA mechanisms are arising, as the most promising among the ones that do not need network support. However, they both require specialized encoding schemes and are of increased complexity. Among the mechanisms requiring some network enhancements, the RLMP features a significant improvement over RLM. However, it is more probable that instead of a two priority dropping function, not placed inside a IP QoS framework, DiffServ islands will be deployed. In this sense the core DSLM architecture, while it does not resolve the congestion-from-undelivered-packets problem, for a balanced DiffServ island it can provide delivery of multicast video in a scalable and simple manner, achieving the highest possible bandwidth utilization and congestion responsiveness, while keeping low applications complexity. For heterogeneous and extended environments, DSLM can be combined with the extensions described in section network congestion information monitoring and propagation mechanisms described in subsection 3.3, providing solutions to the congestion-from-undelivered-packets problem, with similar complexity to RLMP, in terms of router processing and to NLM in terms of congestion monitoring, but without abusing the TTL decrement mechanism, while offering a simple solution in the form of the core DSLM.

Table 2: A summary of the characteristics of mechanisms suitable for the multicast delivery of video streams

	Inflated from:	Network Support	Intra-protocol fairness	TCP friendliness	Responsiveness to Congestion	Long Leave Latencies	Group Management	Codec Requirements	Comments
RLM	Rcv	N	M	N	M	Y	Y	Layered Encoding	Trade off between number of layers and rate adaptation flexibility
RLC	Rcv	N	> RLM	Y	≈ RLM	≈ RLM	≈ RLM	Layered Encoding	Non-flexible layer rate dimensioning
FLID-DL	Rcv	N	≈ RLC	Y	> RLM	< RLM	> RML	Digital Fountain & Dynamic Layering	Highly specialized encoding is required
LTS/TFRP	Rcv	N	> RLM	Y	≈ RLM	≈ RLM	< RLM	Layered Encoding	Scalable measurement off loss and delay required
MLDA	Rev & Src	N	> RLM	Y	> RLM	< RLM	< RLM	Dynamic Layering	Protocol and application complexity
RLMP	Rev & Net	Y	> RLM	NCD	> RLM	< RLM	≈ RLM	Layered Encoding	Custom priority framework required. Enhancements to multicast routing protocols
NLM	Net	Y	>> RLM	Y	>> RLM	N	N	Layered Encoding	Enhancement to routing protocol, appropriate TTL values selection

	Initiated from:	Network Support	Intra-protocol fairness	TCP friendliness	Responsiveness to Congestion	Long Leave Latencies	Group Management	Codec Requirements	Comments
DSL	Net	Y	>> RLM	NCD	>> RLM	N	N	Layered Encoding is not necessary	DiffServ deployment. Congestion from undelivered packets unresolved
DSL+	Net	Y	>> RLM	NCD	>> RLM	N	N	Layered Encoding is not necessary	DiffServ deployment. Enhancements to multicast routing protocol

5. CONCLUSIONS

In this paper, a thorough survey of the most considerable mechanisms for the multicast delivery of layered video was conducted. Most of them were proposed to best function within a BE network, or used some custom network enhancements. However, several limitations of those mechanisms have not been fully addressed yet. Based on the assumption that the future Internet will offer services beyond the BE, with the DiffServ framework being one of the most promising, we proposed the DSLM architecture for the multicast delivery of layered video. DSLM, by exploiting the DiffServ framework, does not assume application level mechanisms, as the network handles packets drops. DSLM can comprise a building block, offering multicast delivery of layered video, in the framework of a properly configured DiffServ domain. DSLM core architecture achieves high congestion responsiveness, fairness and bandwidth utilization, while keeping the application and network simple and not requiring special codecs. Moreover, combining per PHB drop precedence congestion monitoring, a congestion propagation protocol and a drop policy triggered by collected network congestion information, the congestion from undelivered packets problem can be resolved. The implementation complexity of this network enhancements were similar to the proposed enhancements in RLMP and NLM. Work in progress includes the conduction of extensive simulations

and experiments in order to compare quantitatively the DSLM architecture to other candidates, verify the overall architecture and fine-tune network parameters. Future work will include, combination with solutions based on active networking and focused towards devices heterogeneity [21] and incorporation of the DSLM architecture to a stateless multicast DiffServ core as described in [22].

REFERENCES

- [1] S. Floyd, "Promoting the use of end-to-end congestion control in the internet," *IEEE/ACM Transactions on Networking*, August 1999.
- [2] J. Widmer, R. Denda, M. Mauve "A survey on TCP-friendly congestion control," *IEEE NETWORK*, vol 15, 2001.
- [3] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, "An architecture for differentiated services," RFC 2475, December 1998.
- [4] S. McCanne, V. Jacobson, M. Vetterli "Receiver-driven layered multicast," *Proceedings of ACM SIGCOMM*, August 1996.
- [5] L. Vicisano, L. Rizzo, J. Crowcroft "TCP-like congestion control for layered multicast data transfer," *Proceedings of INFOCOM*, April 1998.
- [6] J. Byers, M. Frumin, G. Horn, M. Luby, M. Mitzenmacher, A. Roetter, W. Shaver, "FLID-DL: Congestion Control for Layered Multicast," in *Proceedings of NGC 2000*, November 2000.
- [7] J. Byers, M. Luby, M. Mitzenmacher, and A. Rege, "A digital fountain approach to reliable distribution of bulk data," *Proceedings of ACM Sigcomm '98*, Sept. 1998.
- [8] T. Turletti, S. Parisi, and J. Bolot, "Experiments with a layered transmission scheme over the internet," *Tech. Rep.*, Technical report RR-3296, INRIA, France, Nov. 1997.
- [9] W. Tan and A. Zakhor, "Error control for video multicast using hierarchical fec," *Proc. International Conference on Image Processing*, Oct. 1999.
- [10] D. Sisalem and A. Wolisz, "Mlda: A tcp-friendly congestion control framework for heterogeneous multicast environments," *Eighth International Workshop on Quality of Service (IWQoS)*, June 2000.
- [11] X. Hénocq, F. Le Léannec, C. Guillemot, "Joint source and channel rate control in multicast layered video transmission," *SPIE Visual Communication and Image Processing conference*, Perth, June 2000.
- [12] K. Kang, D. Lee, HY Youn, K. Chon "NLM: network-based layered multicast for traffic control of heterogeneous network," *Computer Communications*, March 15 2001.
- [13] R. Gopalakrishnan, J. Griffioen, G. Hjalmtysson, C.J. Sreenan, S.Wen., "A simple loss differentiation approach to layered multicast," *Proceedings of IEEE International Conference on Computer Communications (INFOCOM)*, pp. 461-469, March 2000.
- [14] J. Heinanen, F. Baker, W. Weiss, J. Wroclawski, "Assured Forwarding PHB Group," RFC 2597, June 1999.
- [15] S. Bajaj, L. Breslau, S. Shenker, "Uniform versus priority dropping for layered video," *Proceedings of SIGCOMM 1998*.
- [16] P. Pieba, N.Sedig, B. Nandy, "The dynamics of TCP and UDP interactions in IP QoS Differentiated Services networks," *Proceeding of the 3rd Canadian Conference on Broadband Research*, November 1999

- [17] W. Tan and A. Zakhor, "Packet classification schemes for streaming MPEG video over delay and loss differentiated networks," Proceedings of Packet Video Workshop, Kyongju, Korea, April 2001.
- [18] A. Cellatoglu, S. Fabri, S.T. Worrall, A.M. Kondo, "Use of Prioritised Object-Oriented Video Coding for the Provision of Multiparty Video Communications in Error-Prone Environments," IEEE VTC-Fall, Amsterdam, 1999.
- [19] H. Sanneck, N. Tuong Long Le, A. Wolisz, "Intra-flow Loss Recovery and Control for VoIP in Proceedings of ACM Multimedia 2001," Ottawa, September, 2001.
- [20] F. Wang, P. Mohapatra, "Using Differentiated Services to support Internet Telephony," Computer Communications 24 (2001) 1846-1854
- [21] H. Akamine, N. Wakamiya, M. Murata, H. Miyahara, "An approach for Heterogeneous Video Multicast Using Active Networking," Proceedings of Second International Working Conference on Active Networks (IWAN2000), October 2000.
- [22] A. Striegel, G. Manimaran, "A Scalable Approach to DiffServ Multicasting," in Proc. Of International Conference on Communications (ICC'2001), Helsinki, Finland, June 2001.