

MULTIGROUP COMMUNICATION

Using active networks technology

Agnieszka Chodorek¹ and Robert R. Chodorek²

¹*Department of Telecommunications and Photonics, The Kielce University of Technology, al. Tysiąclecia Państwa Polskiego 7, Kielce, Poland;* ²*Department of Telecommunications, The AGH University of Science and Technology, al. Mickiewicza 30, Kraków, Poland*

Abstract: Common multicast tree shared by all layers/streams belonging to one session, is essential to provide multicast-based congestion avoidance. It enables both synchronization of layered data (in the case of layered multicast) or streams (in the case of multicast stream replication) and stable congestion control. Although many authors have been addressed the problem of multicast transmission, the methodology of building common delivery trees still remains an unresolved issue. In the paper, a new solution of that problem is proposed – a multigroup communication based on active network technology.

Key words: multicast communication, congestion control, active networks

1. INTRODUCTION

There are two main architectures of multicast-based congestion avoidance, able to preserve real-time characteristics of multimedia transmission: receiver-driven layered multicast and stream replication multicast (Kim and Ammar, 2001). In the both cases, the multimedia stream is divided into several complementary layers (layered multicast) or independent streams (stream replication) with different QoS requirements. All layers/streams are synchronised and simultaneously transmitted through the network as separate multicast groups. Receivers can individually subscribe or unsubscribe to the appropriate multicast groups to achieve the best quality signal that the network can deliver.

The great advantage of multicast-based congestion avoidance is that it is a good solution of network heterogeneity problem. However, such a transmission assumes, that all multicast groups (layers or streams) will

follow the same multicast tree even if they are sent separately (Matrawy et al., 2002). Above assumption is especially necessary in the case of layered multicast, where identical propagation parameters for each layer allow to avoid loss of synchronism of transmitted layers.

Moreover, common multicast tree, shared by all layers/streams belonging to one session, is needed to provide stable congestion control. Receiver-driven congestion avoidance forms a close loop control from point of congestion to the receiver and to the point of congestion again. The receiver acts as a controller, which motivates the control device (network node at the point of congestion) to perform rate adaptation. Stable control process will be possible only if group membership stay in one-to-one relationship with effective transmission rate measured at the point of congestion.

Unfortunately, common multicast tree cannot be guaranteed in, connectionless in nature, IP network. Although many authors have been addressed the problem of layered multicast transmission management, also using active network technology (Yamamoto and Leduc, 2000), the methodology of building common delivery trees still remains an unresolved issue. In the paper, a one of possible solution of that problem is proposed – a multigroup communication based on active network technology. Instead of the, mentioned above, previous propositions of active networks based layered multicast, the proposed management scheme allows full utilisation of existing multicast infrastructure.

The rest of the paper is organized as follows. Section 2 proposes multigroup communication management. Section 3 describes examples of multigroup management while Section 4 addresses the evaluation of proposed management scheme. Section 5 concludes the paper.

2. MULTIGROUP COMMUNICATION USING ACTIVE NETWORKS TECHNOLOGY

2.1 A concept of multigroup communication

W define multigroup as *a set of multicast addresses, belonging to one multicast sessions, which follow the same multicast tree, while multicast forwarding is adapted individually to particular group membership.* Multigroup comprises of a set of multicast groups, tied together by common delivery tree. It was assumed, that multicast delivery tree will be constructed on the basis of multicast distribution of the lowest-quality stream (base layer or base stream). This assumption arose directly from the concept of layered multicast, where receiver have to connect to the multicast group, which transmits the base stream and then connect to one or more supplementary

groups (in order of relevance of carried substreams). In result, all receivers are always connected to the multicast group of the base layer and the base layer's delivery tree is, in fact, the base delivery tree, which always connects possible session participants. In the case of multicast stream replication, the base (lowest-quality) stream also is the only stream to which each receiver must connect, at least once (at the beginning of transmission).

The proposed multigroup management, able to construct multigroup from a separate layers, is implemented using active network (AN) technology. Code, which defines the behaviour of the multigroup, is moving along delivery tree. Instead of typical AN-based solutions, the code isn't distributed from the sender to receiver(s), but it is distributed hop by hop, from receiver to the sender (exactly: the router closest to the source) – as the other multicast management messages are moved.

For multigroup management purposes, the session description transmitted by Session Description Protocol (SDP) must be extended by code, which defines the behaviour of the multigroup. In particular, the code conveys information about multicast group behaviour within the multigroup and packet marking policy. Receiver obtains the code together with other session data transmitted by SDP. During the connection's establishment, a copy of a code is installed at network nodes within a delivery tree. Thus, each network node obtains definition of a multigroup behaviour (in particular, a method of joining and leaving multigroup as well as a single layer).

2.2 Joining and leaving multigroup

A receiver connects to the multigroup sending the `join_multigroup` message. The `join_multigroup` message is sent to the default Designated Router (DR), as the IGMP (or MLD) reports are. The DR router is usually the nearest router supporting IP multicast. If DR router belongs to the shared tree (there is a receiver in link-local network connected to the base layer), DR router will activate data delivery on interface, from which it receives the `join_multigroup` message. Otherwise, the router will have to connect to the delivery tree.

A router (or DR router) which want to connect to common delivery tree install the code and sends the `join_multigroup` message to the NH router. NH router is the next hop router in delivery tree, toward the root. The address of the NH router can be obtained from the routing protocol. Delivery tree is stored in local cache for fast lookup purposes. If, from any reason, routing protocol changes delivery tree, cache will be automatically updated.

Code is forwarded along delivery path to the next hop toward the root. NH router checks the uniqueness of the code, and (if needed) install the code and then sends code to the successive hop on delivery path. The sequence is

repeated until the root or the first router served the base layer is achieved. In result, each router in the path obtains code. To rebuild distribution tree standard multicast routing protocols are used. Its worth remarking, that (instead of IP multicast) `join_multigroup` message is used for communication both between receiver and DR and between routers.

Receiver leaves multigroup by sending `leave_multigroup` message to the DR router, which checks the global number of receivers connected to the layer `l` in router's directly attached networks. This number is stored into the global variable `rcv_all[l]`. If the `rcv_all[l]` is equal to zero, the router will send the `leave_multigroup` message to the NH router and will activate prune mechanism of routing protocol. In result, the router leaves multicast delivery tree, then terminates execution of the code and removes it.

2.3 Joining and leaving layers

If the receiver connects to the upper layer, sends the `join_layer` message. The message propagates along the delivery tree, until the first router served the upper layer is achieved. The `join_layer` message activates the transmission of upper layer (activates proper interfaces along the delivery tree). The upper layer transmission follows the same delivery path from the sender to the receiver as the base layer transmission. A simplified algorithm for joining layer is as follows:

```

receive join_layer(layer) message on interface
if (rcv_all[layer] == 0) {
    lookup route cache for NH router
    send join_layer(layer) to NH router
    update forwarding table }
if (rcv[layer, interface] == 0) {
    enable forwarding for interface }
increment rcv[layer, interface]
increment rcv_all[layer]

```

The above pseudo-code defines two counters. The `rcv[layer, interface]` describes the number of receivers at given interface, counted for given layer. The `rcv_all[layer]` counter describes global number of receivers connected to the layer `layer` in router's directly attached networks.

Receiver, which decides to leave layer `layer`, sends the `leave_layer` message to its DR router, which increments both `rcv[layer, interface]`, and `rcv_all[layer]`. If the number of receivers at the given interface reaches 0, the router won't propagate data via this interface. If the global number of receivers reaches 0, the router will send the `leave_layer` message to the upper (the next) router. When a layer is leaved, the remaining datagrams of the layer are useless and should be removing from the queue rather than be

carried to the receiver. The last operation carried out during the leaving layer is flush procedure, which discards these datagrams. A simplified algorithm for leaving layer is as follows:

```

receive leave_layer(layer) message on interface
decrement rcv[layer, interface]
if (rcv[layer, interface] == 0) {
  disable forwarding for interface }
decrement rcv_all[layer]
if (rcv_all[layer] == 0) {
  lookup route cache for NH router
  send leave_layer(layer) message to NH router
  update forwarding table
  flush(layer)}

```

Receiver joins and leaves, successively, layer by layer, in order of their relevance. In the case of emergency (e.g. if the receiver will be switch off without prior `leave_layer` message), layers will be leaving using the time-out signalization from IGMP (or MLD) protocol.

2.4 Maintenance of multicast connections

DR routers provides typical, defined by IGMP (or MLD) standard, maintenance procedures to determine if there is any group member on their directly attached networks. Such a group checking is performed using standard mechanisms based on IGMP (or MLD) Query message.

This maintenance procedure actualize information about hosts belonging to each multicast group and allows verifying status of particular group membership. Differences between real and counted status of group membership may be caused by e.g. connection failures or other emergency situations, while receivers leave groups without sending typical `leave_multigroup` or `leave_layer(layer)` messages.

3. EXAMPLES AND SCENARIOS

3.1 Joining the multigroup

Assume that receiver R1 decide to join multigroup, which serve the MPEG-4 layered video transmission. Initially no one receiver is connected to the multigroup. In this example, message with code is distributed along delivery tree from the DR router to the root. The process of joining multigroup is performed in 7 steps (Fig. 1):

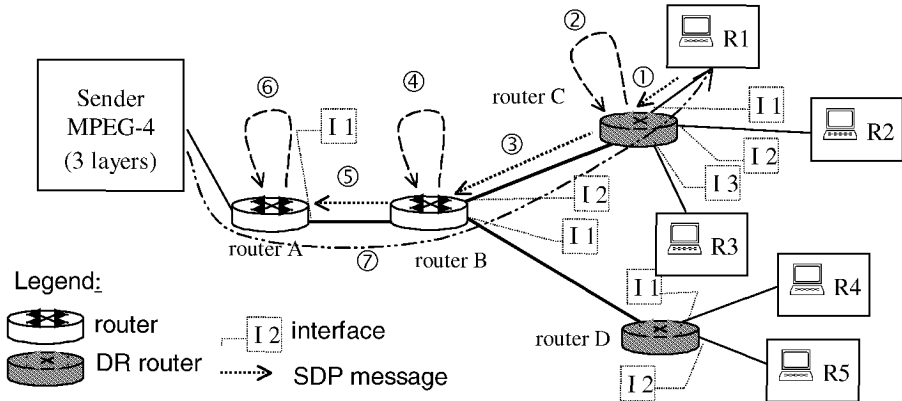


Figure 1. Receiver R1 joins the multigroup

1. Receiver R1 sends the `join_multigroup` message to the router C (a DR router of the receiver's link-local network).
 2. Router C gets the `join_multigroup` message via the interface I1. Check the uniqueness of the code. The code hasn't been yet installed, so the router installs it and sets parameters. In particular, the global number of receivers connected to the base stream `rcv_all[1]` and the number of receivers at given interface `rcv[1,1]`, are set to one. Router creates delivery tree using multicast routing protocol (updates forwarding table).
 3. Router C lookup for the address of the NH router in local cache and sends the `join_multigroup` message to the NH router (router B).
 4. Router B gets the `join_multigroup` message via the interface I2 and repeats the processing of the obtained code (see 2).
 5. Router B lookup for the address of the NH router in local cache and sends the `join_multigroup` message to the NH router (router A).
 6. Router A gets the `join_multigroup` message via the interface I1.
 7. Router A activates interface I1. Router B transmits the multicast packets which convey the base layer to router C via the interface I2. Base layer is transmitted from the router C via the interface I1 to the receiver R1.
- Router A is a root of delivery tree and receives multimedia data strictly from the sender.

3.2 Joining upper layers (streams)

Assume that receivers R1 and R5 decide to join the upper (namely, 2nd) layer. Initially all receivers are connected to the multigroup, but only R2 and R3 are connected to the layer 2. In the first example, message with code is transmitted only from the receiver to the DR router. The process of joining the upper layer by receiver R1 is performed in 3 steps (Fig. 2, scenario A):

1. Receiver R1 sends the `join_layer(2)` message to the DR router (C).
2. Router C gets the `join_layer(2)` message via the interface I 1. Check the status of counter `rcv_all[layer]`. Receivers R2 and R3 are connected to layer 2, so `rcv_all[2] == 2`. Thus, router C activates forwarding of multicast packets to interface I 1 and increments variables `rcv[2,1]` and `rcv_all[2]`. In result, `rcv[2,1]=1` and `rcv_all[2]=3`.
3. Layer 2 is transmitted from the router C (via I 1) to the receiver R1.

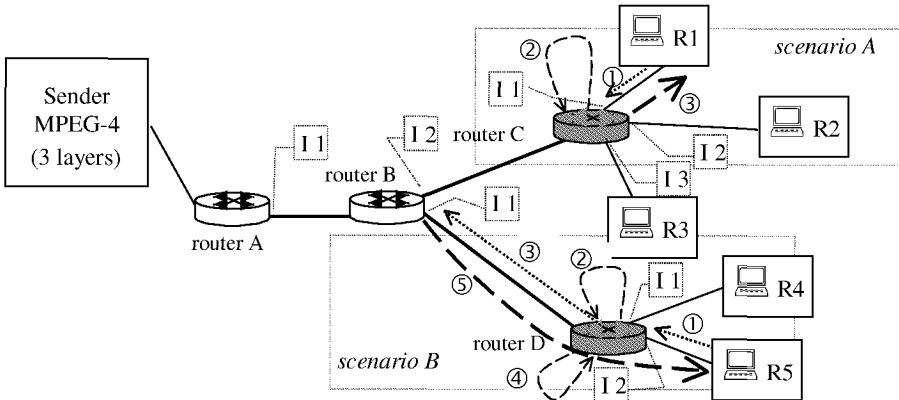


Figure 2. Receivers R1 and R5 joins the layer 2

The second example shows transmission of the `join_layer(layer)` message from the receiver to the router B. The process of joining the upper layer by receiver R5 is performed in 5 steps (Fig. 2, scenario B):

1. Receiver R5 sends the `join_layer(2)` message to the DR router (D).
2. Router D gets the `join_layer(2)` message via I 2, checks the status of counter `rcv_all[layer]`. No one receiver is connected to layer 2 in router's directly attached networks (`rcv_all[2] == 0`). Thus, router C activates forwarding of multicast packets to interface I 2 and increments variables `rcv[2,1]` and `rcv_all[2]`. In result `rcv[2,1] = 1` and `rcv_all[2]=1`.
3. Router D sends the `join_layer(2)` message to the NH router (B).
4. Router B gets the `join_layer(2)` message via I 1. Because `rcv_all[2] == 3`, router B activates interface I 1 and increments variables `rcv[2,1]` and `rcv_all[2]`. In result `rcv[2,1] = 1` and `rcv_all[2] = 4`.
5. Layer 2 is transmitted from the router B through the router D to R1.

3.3 Leaving upper layers (streams)

Assume that receivers R1, R2 and R3, are connected to the layer 3 and receiver R5 to layer 2. Due to possibility of congestion, receiver R5 have to leave layer 2 to reduce effective transmission rate in the access network.

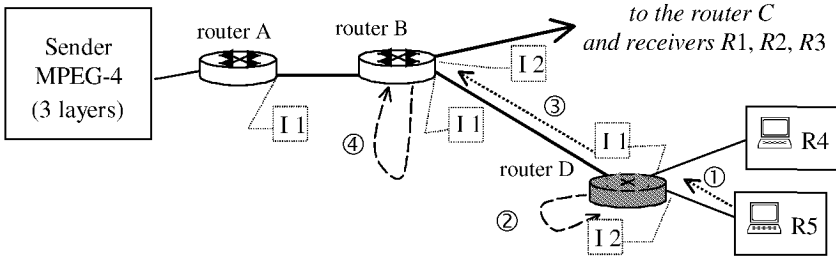


Figure 3. Receiver R5 leaves the layer 2

In this example, `leave_layer(layer)` message is transmitted from the receiver R5 to the first router in tree, which counter $rcv_all[layer] > 1$. The process of leaving layer 2 by receiver R5 is performed in 4 steps (Fig. 3):

1. Receiver R5 sends the `leave_layer(2)` message to the DR router (D).
2. Router D gets the `leave_layer(2)` message via I 2. Router decrements $rcv[2,2]$. Because $rcv[2,2]$ is now equal to 0 (no one receiver is connected via I 2), router D stops forwarding via I 2 and decrements $rcv_all[2]$.
3. Because $rcv_all[2] = 0$, router D sends `leave_layer(2)` to NH router.
4. NH router (B) gets the `leave_layer(2)` message via I 1, decrements $rcv[2,1]$. Because $rcv[2,1] = 0$, router B stops forwarding of multicast packet via I 1 and decrements $rcv_all[2]$. Counter $rcv_all[2] = 3$, so router B forwards multicast packets and don't propagate `leave_layer(2)` to NH router (router A).

4. RESULTS

Described multigroup management was successfully implemented in Berkeley's *ns-2* network simulator environment. As the source of elastic traffic (ST), FTP over TCP (SACK version) was used. TCP packets have a size of 1000 bytes. As a video application, the ECN-capable layered multicast was used – see (Chodorek, 2003) for details. Video traffic sources (SV) were modeled as three-layer VBR streams, generated from 13 publicly available video traces, encoded in high, medium and low picture quality. Properties of traces can be found in (Fitzek and Reisslein, 2001). The video streams use ECN-capable RTP protocol (Chodorek, 2002) and RTP packets have maximum 188 bytes.

The proposed transmission scheme has been simulated in a 3 different topologies (Fig. 4), to expose the performance issues as well as scalability. Senders are connected to the router at 100 Mbps and 1 ms delay. Receivers are connected to the router through 1 ms delay link. Routers are connected

with a link at 5 ms delay. All routers are ECN-capable (RED queue marks packets instead of dropping) and utilize 3-level RED queue.

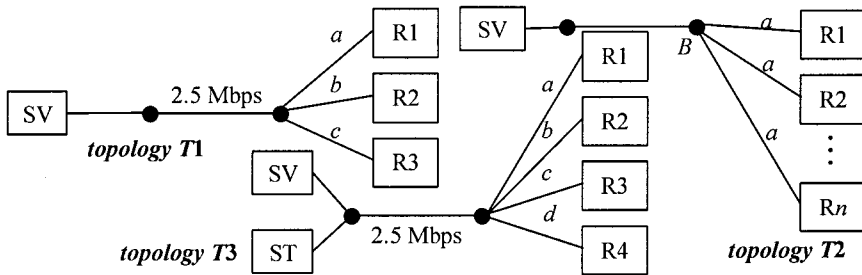


Figure 4. Simulation topologies. Parameters of T1: $a = 0.4$ Mbps, $b = 0.8$ Mbps, $c = 2$ Mbps. Parameters of T2: $a = 100$ Mbps. Parameters of T3: $a = b = c = 2$ Mbps, $d = 0.2$ Mbps.

Topology *T1* was used for adaptability tests. Receiver R3, connected via non-congested link, always was able to receive full video information (layer 1 to 3). Receiver R2 was connected via lightly congested link and always was able to receive layer 2. If the realistic video source was characterized by low-detail, slowly dynamic content, R2 could connect to layer 3. Receiver R1 was connected through heavy congested link. R1 was able to receive layer 1 and, if the video trace was small enough, could connect to layer 2.

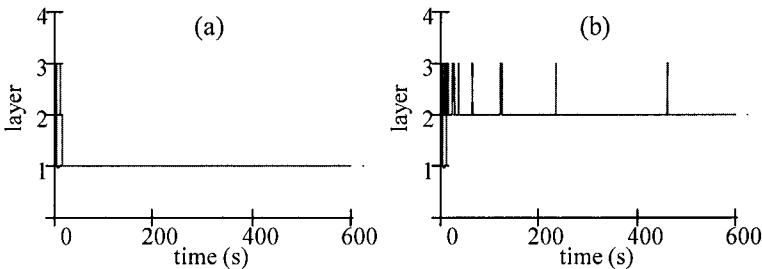


Figure 5. Adaptability (R1, *starwars* video): a) without multigroup, b) with multigroup.

Experiments shows, that adaptability of the receiver-driven layered multicast is the same or better when the multigroup communication is used, because of faster reaction on the variable network conditions (here: VBR traffic). In Figure 5 an example of such a situation is depicted. The slow reaction of the system without multigroup (layer 3 was leaved about 2 times slower without the multigroup) has resulted in longer congestion during the first seconds of transmission, what lead to the saturation of RED's average queue size, so the receiver gets ECNs even if congestion is over. Last, but not least, reason of better behavior of the system implementing multigroup

communication is flush procedure, which discards packets belonging to unsubscribed layer from an output queue, what increases the bandwidth utilization and decreases delay of the rest of packets.

Table 1. Packet loss ratio. Because the ECN-capable transmission was used, the losses are caused only by buffers overflows.

video source	with multigroup	without multigroup	video source	with multigroup	without multigroup
bean	0	0	lambs	38.6	42.9
cam1	0	0	ski	0	1.1
cam2	0	0	startrek	0	0
cam3	0	2.6	starwars	0	0.9
dieh	0	3.5	troop	0	3.1
dino	5.3	5.2	vclips	0	1.6
form1	0	4.4	-	-	-

Adaptability can be measured as the function of packet loss – if the system is fine-tuned to the network conditions, packet loss ratio will be small. As shown in Table 1, packet loss ratio equals 0 (the best adaptation) in the case of 11 movies of 13 while the multigroup was used. Only 4 movies of 13 have achieved the best adaptability without the multigroup. If the network is not well-dimensioned (losses larger than 5% – *dino* and *lambs* video sources), the multigroup improves adaptability (*lambs*), although the other situation, where packet loss ratio was a little larger for proposed scheme also was observed (*dino*).

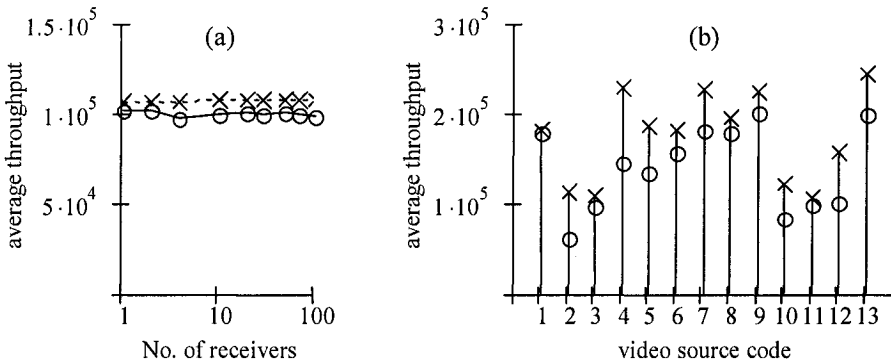


Figure 6. Session-size scalability of a system with (x's) and without multigroup (o's): a) average throughput as a function of number of receivers (*starwars* video source), b) average throughput observed for population of 30 receivers. Video source codes: 1 – *bean*, 2 – *cam1*, 3 – *cam2*, 4 – *cam3*, 5 – *dieh*, 6 – *dino*, 7 – *form1*, 8 – *lambs*, 9 – *ski*, 10 – *startrek*, 11 – *starwars*, 12 – *troop*, 13 – *vclips*.

Session-size scalability was investigated using topology *T2*. Bandwidth *B* was large enough to assure that receivers can connect to layer 2. Results of experiments are depicted in Figure 6. Although both systems (with and without multigroup communication) achieve very good scalability (Fig. 6a), achievable throughput is usually larger for system with multigroup communication (Fig. 6b). This is caused both by faster reaction of the system and discarding IP datagrams belonging to unused layers.

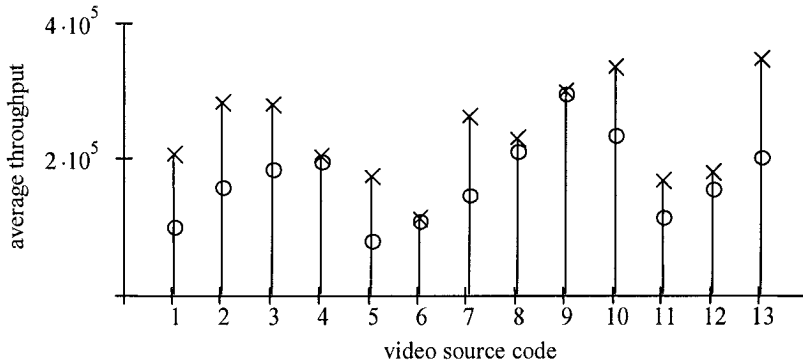


Figure 7. RTP competing with TCP flow. Video source codes: as in Fig. 6.

Other investigations were carried out using topology *T3*, where *R1...R3* receives VBR stream and *R4* receives TCP packets. During the experiment, ECN-capable layered multicast share link with ECN-capable TCP flow. Results shows, that TCP flow always (with and without multigroup, for all tested video sequences) achieves throughput close to the nominal (0.2 Mb/s), what confirms effects described in (Chodorek, 2003). Thus multigroup communication doesn't influence on TCP-friendliness of layered multicast. However, multigroup communication increases the average throughput of competing video from 1% (*dino*) to more than 110% (*bean*) and in more than 60% of movies (8 of 13) this growth was significant (Fig. 7).

Some elements of the management scheme, essential from the transmission point of view, were tested in the Linux active routers. The remarks from field trials are as follows:

- the multigroup communication together with IGMPv2 gives faster layer switching than usage of IGMP alone – receiver leaves layer about 2 seconds faster than without multigroup management, while duration of layer joining remains unchanged,
- in the case of dense mode multicast routing protocols, which doesn't require rebuilding of delivery tree during layer switching, multigroup communication doesn't give significant advantage,

- in the case of sparse-mode multicast routing protocols, where the layer switching rebuilds delivery tree, the usage of multigroup communication results in faster layer switching.

Above results do not cover additional mechanisms, which accelerate join/leave operations, as for example flush procedure.

5. CONCLUSIONS

Multigroup comprises of a set of multicast groups, belonging to one session, tied together by common delivery tree. It can be understood as a set of multicast addresses, which follow the same multicast tree, while multicast forwarding is adapted individually to particular group membership. It's worth remarking that the multigroup communication do not require continuous IP multicast address space.

Multigroup communications simplifies the management of multiple groups belonging to one multicast sessions. Experimental results show, that this simplification improves utilization of network resources and improves adaptability of receiver driven layered multicast.

ACKNOWLEDGEMENTS

This research was supported by State Committee for Scientific Research (KBN) under grant No. 4 T11D 015 24 (years 2003-2005)

REFERENCES

- Chodorek, R. R., 2002, A simulation of ECN-capable multicast multimedia delivery in ns-2 environment, *Proc. of 14th European Simulation, ESS'2002*, pp. 233-237.
- Chodorek, R. R., 2003, ECN-capable multicast multimedia delivery, *Springer LNCS 2720*, pp. 62-72.
- Fitzek, F.H.P., and Reisslein, M., 2001, MPEG-4 and H.263 video traces for network performance evaluation, *IEEE Network 15(6)*: 40-54.
- Kim, T., and Ammar, M.H., 2001, A comparison of layering and stream replication video multicast schemes, *Proc. of NOSSDAV 2001*.
- Matrawy, A., Lambadaris, I., and Huang, Ch., 2002, Comparison of the use of different ECN techniques for IP multicast congestion control, *Proc. ECUMN'02*, pp. 74-81.
- Yamamoto, L., and Leduc, G., 2000, An active layered multicast adaptation protocol, *Springer LNCS 1942*, pp. 180-194.