社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

# アクティブネットワークにおける輻輳適応型動画像マルチキャスト

アカミネ　エクトル† 　若宮　直紀†† 　宮原　秀夫††

† 大阪大学 大学院基礎工学研究科　〒 560–8531 大阪府豊中市待兼山町 1–3
†† 大阪大学 大学院情報科学研究科　〒 560–8531 大阪府豊中市待兼山町 1–3
E-mail: †akamine@nal.ics.es.osaka-u.ac.jp, ††{wakamiya,miyahara}@ist.osaka-u.ac.jp

**あらまし**　本稿では，受信状況が様々に異なる複数のクライアントに効率よく実時間で動画像ストリームをマルチキャスト配信するための，アクティブネットワーク技術を利用した輻輳適応型動画像マルチキャスト技術を提案している．階層構造をなすネットワークにおいて，ローカルドメインに配置されたアクティブノードがドメイン内の動画像マルチキャストを管理するローカルサーバとなり，クライアントのリクエストを収集し，ネットワークの負荷状態にあわせて品質調整により動画像レート制御を行い，適切に設定されたマルチキャストグループへ配信する．本稿では，クライアントの動画像ストリーム受信状況に応じて動画像マルチキャストグループを動的に再構成するメカニズムを提案し，シミュレーションによりそれぞれのクライアントにあった動画像が実時間配信できることを示している．
**キーワード**　アクティブネットワーク，動画像マルチキャスト，TCP-friendly，動画像品質調整

# Congestion-Adaptive Video Multicast in an Active Network

Hector AKAMINE†, Naoki WAKAMIYA††, and Hideo MIYAHARA††

† Graduate School of Engineering Science, Osaka University　Toyonaka, Osaka 560–8531, Japan
†† Graduate School of Information Science and Technology, Osaka University　Toyonaka, Osaka 560–8531, Japan
E-mail: †akamine@nal.ics.es.osaka-u.ac.jp, ††{wakamiya,miyahara}@ist.osaka-u.ac.jp

**Abstract**　We present a service model for video multicast in an active network that considers heterogeneity of the session clients. We consider a network with a hierarchical structure, where active nodes located at the edges become local servers, taking charge of the multicast transmission to the clients in the local domain. Active nodes adjust the rate of the incoming stream, and transmit the filtered streams to the local clients. Clients are organized in several multicast groups, according to their reception conditions. We show through preliminary simulation experiments the effectiveness of the approach.
**Key words**　active network, video multicast, TCP-friendly, video filtering

## 1. Introduction

Video multicast services must consider the constraints imposed by a heterogeneous environment. Bandwidth of the network links, network congestion, and processing capabilities of the clients' hosts affect the session members differently, and therefore it becomes necessary to provide several streams of different quality even in the same session. We can use diverse approaches to provide different quality streams. The most cited method is layered encoded video, which has the advantages of being completely receiver-driven and that does not require extra support from the network [1]. Nevertheless, layered encoding adds overhead both in the size of the resulting streams and in the complexity required at the decoding, and can offer only a limited number of quality levels. In an active network, some of the nodes, called active nodes, have enhanced packet processing capabilities. Active nodes can be programmed to improve the service offered by the network to the end hosts. An application of active network technology is filtering [2], in which active nodes modify the rate of multimedia streams to adapt them to the network conditions.

In this paper, we consider the use of filters in an active network to provide a heterogeneous multicast video service without the limitations of layered encoding. Active nodes, located at the edge or local domains, act as local servers, receiving requests from the domain clients, and grouping them according to their reception conditions. The origin server sends the video stream to the local servers which, if required, generate different rate streams according to the formed groups. Rate is adjusted using TCP-friendly rate control techniques. To provide such a service, local servers and clients communicate with each other and organize groups to provide clients with video

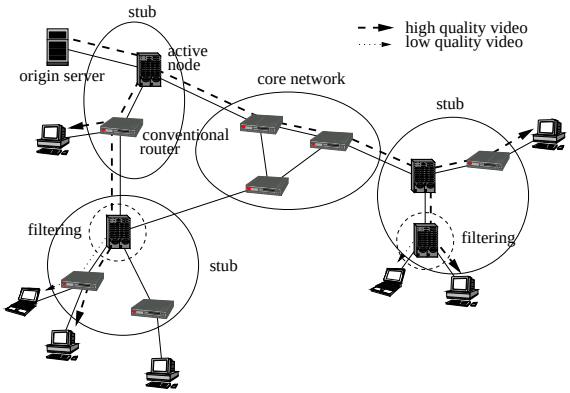


Figure 1 Service model

streams of satisfactory quality in an efficient way. In Section 2. we first show the general idea of the service considering a network divided in several domains. In Section 3. we explain the role of the active nodes inside each domain. Some simulation results are shown in Section 4.. We point out some conclusions and future work in Section 5..

## 2. Service Scheme

In the simplest service scheme, the video server distributes the video streams to all the requesting clients, regardless of their location and the network topology in a "flat" tree. This approach has scalability problems, resulting in a poorly offered service. If we take into consideration that large network topologies have a hierarchical structure [3], active network technology can help to distribute the load of the video server to improve the server offered to the session members. In Figure 1, the network is composed of several domains. We place active nodes at the edge networks, which assist the origin server distributing video streams for the clients in their respective domains. We call them *local servers*, as they work as servers for the clients in the domain. There can be several local servers in a stub domain. In this case, they cooperate to provide clients a better service as explained in subsection 3. 2.

## 3. Service at the Local Domain

Local servers process requests from clients that wish to join the video session. Session members are gathered in several multicast groups according to their reception conditions. Local servers take streams from the origin server and provide each multicast group with a filtered stream whose rate is appropriate for the group members, as shown in Figure 2. The service lies between unicast and standard multicast. In unicast, a different stream is sent to each client, requiring excessive network resources. In standard multicast, the server sends all the clients the same stream, saving resources, but leaving clients unsatisfied if their reception conditions are diverse. In the proposed service, clients with similar reception conditions are grouped together, and a stream with an appropiate rate is sent to each multicast group.

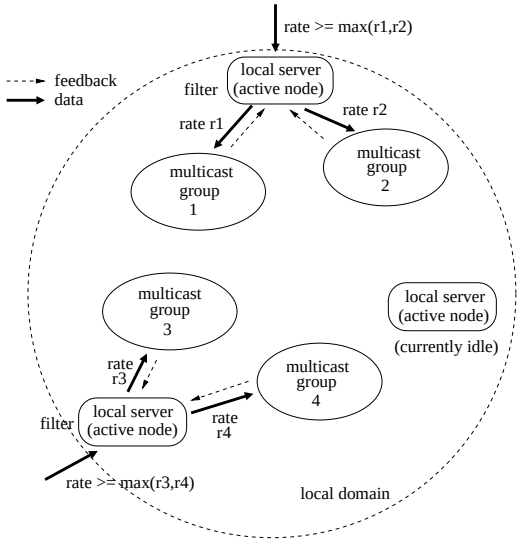


Figure 2 Service scheme inside a domain

The service dynamically groups clients with similar reception conditions, allots them to the appropriate local server, and controls the rate of the streams sent to each multicast group. Rate control of the transmitted streams is required to adapt to the state of the network, reduce packet loss due to congestion, and assure that network resources are used fairly. We consider to use TCP-friendly rate control. In the case of multicast, the TCP-friendly rate is set to the "worst receiver" of the group. For this reason it is necessary to complement the rate control with a mechanism to dynamically group clients with similar reception conditions.

### 3. 1 TCP-Friendly Rate Control

In a best-effort network, the rate of data streams must be controlled to reduce network congestion and achieve fair use of the network resources. A great proportion of the Internet traffic is sent using TCP, which limits the rate of unicast flows to reduce the occurrence of congestion. In contrast, multicast and other applications that send data using UDP, do not have a built-in rate control mechanism, and can clog the network since they maintain the same sending rate even if the network is congested. To solve this problem, several mechanisms have been proposed to regulate the rate of those data streams, which to try to achieve the same throughput as a TCP connection under the same conditions, hence the name TCP-friendly rate control. The most popular of these approaches is TFRC [4], which is an equation-based congestion control mechanism based on measurements of packet loss and RTT between the sender and receiver.

TCP-Friendly Multicast Congestion Control (TFMCC) [5] is an equation-based congestion control mechanism that constitutes the multicast extension to TFRC. In TFMCC each receiver measures the loss event rate and the RTT to the sender and calculates its TCP-friendly rate. The rate values are passed to the sender in a scalable way, and the sender sets the rate of the multicast data stream to the lowest reported rate.

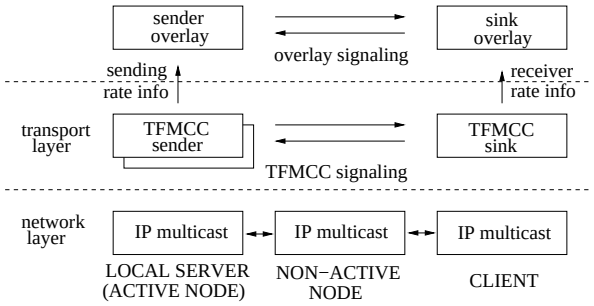


Figure 3 Overlay to TFMCC to handle group membership

### 3.2 Management of Group Membership

TFMCC determines the rate of a multicast group according to the receiver with the worst reception conditions. This is not desirable in the case of a heterogeneous multicast group, where receivers have diverse reception conditions, because group members with better reception conditions must bear to receive the same low-quality video stream. To provide users with video streams having more adequate quality, we need a mechanism to allocate receivers in several groups according to their reception conditions. If there is more than one server in the subnetwork, we also need to choose the most appropriate server for each particular client.

We consider an overlay to TFMCC as shown in Figure 3. Being a transport protocol, TFMCC relies on IP multicast for routing and group join/leave mechanisms. The TFMCC protocol runs between the local server and the receivers of each group. Each TFMCC receiver calculates its own TCP-friendly rate, then sends it to the sender to define the rate of the group. TFMCC uses a feedback suppression mechanism to avoid feedback implosion. The overlay layer runs between a local servers and its receivers. At the receiver side, it sends the calculated TCP-friendly rates to the local server. At the local server side, it uses the received feedback to define group membership. In contrast with TFMCC that basically needs information of the worst receiver, we need rate information of all the members of the group, and then it is not possible to use feedback suppression. Nevertheless, because the control interval is longer, feedback implosion can be controlled because it is sent less frequently.

Membership of the multicast groups belonging to the same server is changed using two events we call *group split* and *group merge*. When a group contains receivers with different reception conditions, it is divided in two groups (split) to reduce the heterogeneity. Analogously, the members of two groups with similar rates are combined to form a single group (merge) to reduce the used network resources.

When multiple local servers are available, we can consider to *move* receivers from one server to another in which they would have better reception. The approach to move receivers is covered briefly and not evaluated, leaving it for future work.

#### 3.2.1 Session Join/Leave

We consider a simple approach for clients to join or leave a session. A client is assumed to know the location of the local servers in the domain and their distance to it, using any convenient metric such as the number of hops or the RTT. The client sends a join message to the closest local server. Then, the server notifies the client with a multicast group address to join. The specified multicast group provides the lowest quality/rate video stream offered by the server, since the reception condition of the new client is unknown. Finally, the client uses standard IP multicast procedures to join the group and the required TFMCC join procedure.

A client leaving a session sends a leave message to its local server, or it can also be assumed to leave after a timeout interval without sending feedback information.

#### 3.2.2 Group split/merge

Clients apply a low-pass filter to the TCP-friendly rate values obtained from the TFMCC layer, and send the filtered values as feedback information to their local server. Filtering is required to absorb the influence of instantaneous and drastic rate variations.

A multicast group is *split* in two if there is too much variation in the reported TCP-friendly rate values. For this purpose, we use the *variation coefficient* $C_v$, $C_v = \frac{\sigma}{\bar{r}}$, where $\sigma$ is the standard deviation and $\bar{r}$ is the average of the reported rates of the clients of the group. If $C_v$ exceeds a *split threshold* $a$, we proceed to divide the group in two.

To define the membership of each group, we order the reported rate values in ascending order, and define the "cut" or the border between two groups. We start setting the cut between the first, lowest rate receiver, and the second. We calculate the $C_v$ for both partitions and calculate the average $\bar{C_v}$. We vary the cut and find the partition that makes $\bar{C_v}$ minimum from all the possible combinations. With the use of $C_v$ we reduce the amount of state information that would be required if we needed to track each client individually. The drawback of using the TCP-friendly rate as the unique parameter, is that it is not possible to detect characteristics in the topology, such as the existence of different bottleneck links, that can help to determine a better group partition.

Two multicast groups are *merged* if their rates are close. We calculate the variation coefficient $C_v$ of the sending rates for each pair of multicast groups sourced by the local server. We compare the lowest of the $C_v$ values obtained with a *merge threshold* $b$, merging the groups if $C_v < b$.

The above criteria for splitting and merging is a very simple approach, but has the drawback of limiting changes in group membership. It is impossible for a client to move from the current group to another one directly.

We do not merge a group that is split in the same control instance. We limit the number of merge and split actions to one each per control interval. Introducing or eliminating too many groups at once could affect the stability of the network, putting it in a state where the measured values and criteria are no longer valid. We set a limit in the number of groups sourced by a local server, to reflect the limitation of available processing resources at the active node.

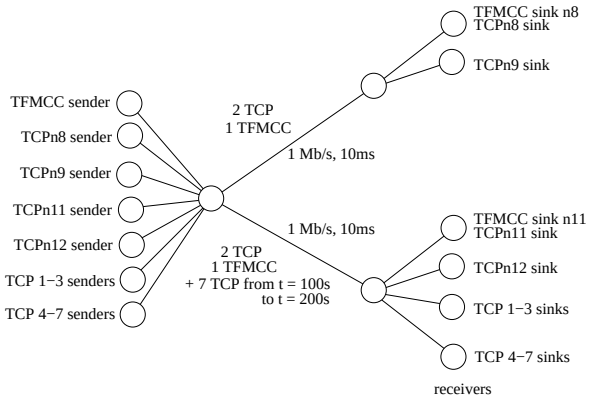


Figure 4 Topology with 2 bottleneck links

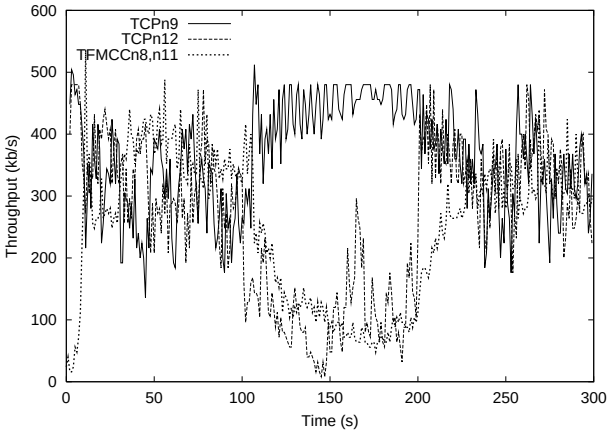


Figure 5 Throughput without splitting

**3.2.3** Moving a group to another server

A local server can decide to pass its lowest rate group to another server in the same domain to improve the reception quality of its members. The group to pass must be homogeneous, with its calculated $C_v$ below the splitting threshold. When a local server decides to move a group, a *probing* phase starts, in which the group receives data from both to the original and the newly elected server. The group is definitely moved to the new server only if the throughput of the stream from the new server becomes greater than the original stream. The evaluation of this mechanism is left for future work.

## 4. Evaluation of Splitting/Merging

We show some simulation results of the splitting/merging mechanism using *ns-2*. Fig. 4 shows a topology with two bottleneck links of 1 Mb/s and a delay of 10ms (access links are set to 100 Mb/s and 10ms) . We set 1 TFMCC and 4 TCP connections for 300s (2 TCP flows traversing each bottleneck link). From t = 100s to t = 200s we set 7 extra TCP flows in the lower bottleneck link.

Fig. 5 shows the throughput of TFMCC when no splitting is done. The sender sets the rate according to the worst receiver and the same rate is sent to all receivers in the group, even though receiver TFMCCn8 has a higher fair rate than TFMCCn11.

In Fig. 6 group splitting/merging is performed. We set the control interval to 5 s, the splitting threshold $a$ to $0.35$ and the merging

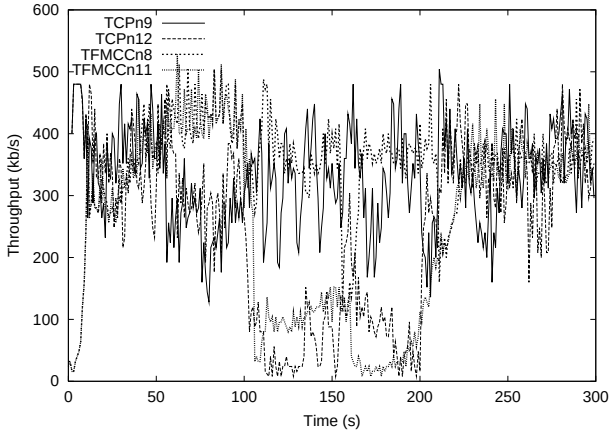


Figure 6 Throughput with group splitting

threshold $b$ to $0.4$. The graph shows a split at t = 105s, a merge at t = 155s, a split at t = 160s, and a merge at t = 210s. The undesired merge at t = 155s is caused by the variation in the TFMCC sender rates, but in this case it is corrected by a split at t = 160s.

## 5. Conclusions and Future Work

We presented a service scheme for heterogeneous video multicast, in which active nodes located at the edges assist the video server taking care of the session clients located at their own local domain. We also proposed a mechanism to control group membership inside each local domain, considering TCP-friendly rates as the criteria to measure heterogeneity. Preliminary simulation results show that the approach can be effective, but further evaluation is required to set the control parameters adequately, and relate them with the network characteristics. We also need to complete the mechanism to move session clients between servers when several servers are present. Finally, it is also necessary to examine the required communication between local servers in different domains and with the origin server.

**Acknowledgements**

This work was partly supported by Special Coordination Funds for Promoting Science and Technology of the Ministry of Education, Culture, Sports, Science and Technology of Japan.

**References**

[1] Lorenzo Vicisano, Jon Crowcroft, and Luigi Rizzo, "TCP-like congestion control for layered multicast data transfer," in *Proc. INFOCOM '98*, March-April 1998.

[2] Ralph Keller, Sumi Choi, Marccel Dasen, Dan Decasper, George Fankhauser, and Bernhard Plattner, "An active router architecture for multicast video distribution," in *Proc. INFOCOM 2000*, March 2000.

[3] K. Calvert, M. Doar, and E. Zegura, "Modeling internet topology," *IEEE Communications Magazine*, vol. 35, no. 6, June 1997.

[4] Sally Floyd, Mark Handley, Jitendra Padhye, and Jörg Widmer, "Equation-based congestion control for unicast applications," in *Proc. SIGCOMM 2000*, August 2000, pp. 43–56.

[5] Jörg Widmer and Mark Handley, "Extending equation-based congestion control to multicast applications," in *Proc. SIGCOMM '01*, August 2001.