Control Evaluation in a LVoD System Based on a Peer-to-Peer Multicast Scheme*

Rodrigo Godoi, Xiaoyuan Y. Xu, Porfidio Hernández and Emilio Luque
Computer Architecture and Operating Systems Department.
Universitat Autònoma de Barcelona, UAB.
Edifici Q, Barcelona 08193, Spain.
{rodrigo, xiao}@aomail.uab.es, {porfidio.hernandez, emilio.luque}@uab.es

Abstract

Providing Quality of Service (QoS) in video on demand systems (VoD) is a challenging problem. In this paper, we analyse the fault tolerance on a P2P multicast delivery scheme, called *P*atch *C*ollaboration *M*anager / *M*ulticast *C*hannel *D*istributed *B*ranching (*PCM/MCDB*) [13]. This scheme decentralizes the delivery process between clients and scales the VoD server performance. PCM/MCDB synchronizes a group of clients in order to create local network channels to replace on-going multicast channels from the VoD server. Using the P2P paradigm supposes facing the challenge of how often peers connect and disconnect from the system. To address this problem, a centralized mechanism is able to replace the failed client. We evaluate the failure management process of the centralized scheme in terms of the overhead injected into the network and analyse the applicability of a distributed approach to managing the process. Analytical models are developed for centralized and distributed approaches. Their behaviour are compared in order to evaluate whether the distributed scheme can improve the fault management process, in terms of reducing server load and generating better scalability.

Keywords: VoD, Multicast, P2P, Fault-tolerance.

Resumen

Proporcionar Calidad de Servicio (QoS) en sistemas de Vídeo bajo Demanda (VoD) es un problema desafiador. En este artículo, analizamos la tolerancia a fallos en un esquema de envío de informaciones, basado en comunicaciones *multicast* y colaboraciones P2P, denominado PCM/MCDB [13]. El esquema descentraliza el proceso de envío de información entre los clientes y escala las prestaciones del servidor de VoD. PCM/MCDB sincroniza un grupo de clientes con objeto de crear canales de redes locales para reemplazar canales *multicast* en curso del servidor. La aplicación del paradigma P2P supone cómo afrontar el problema de la conexión y desconexión de clientes del sistema. Para resolver este problema, un mecanismo centralizado es capaz de reemplazar el cliente fallido. En el trabajo evaluamos el proceso de gestión de fallos del esquema centralizado en términos del flujo de informaciones insertado en la red y analizamos la aplicabilidad de un esquema distribuido para el proceso de gestión. Modelos analíticos son desarrollados para las aproximaciones centralizada y distribuida. Sus comportamientos son comparados con objeto de evaluar si un esquema distribuido puede mejorar el proceso de gestión de fallos desde el punto de vista de reducir la carga del servidor y proporcionar mejor escalabilidad.

Palabras claves: VoD, Multicast, P2P, Tolerancia a fallos.

* This work was supported by the MEyC under contract TIN 2004-03388.

1. INTRODUCTION

Recent advances provide multicast scheme application on real networks. This allows clients to share delivery channels and decrease the server and network resource requirements. The patching multicast policy [01] [02], for example, dynamically assigns clients to join on-going multicast channels and patches the missing portion of video with a unicast channel. The disadvantage of a multicast scheme, compared with unicast, is the complexity of implementing interactive operations, because there is not a dedicated channel per client.

Most recently, the peer-to-peer (P2P) paradigm has been proposed to decentralize the delivery process to all clients, achieving system scalability beyond the physical limitations of VoD servers. In [03] [04], the authors propose the Chaining delivery policy to link clients in a delivery chain. Even though P2P policies achieve high resource requirement reduction in the server, the schemes' applicability in a true-VoD system has been questioned due to client failure problems.

The P2Cast P2P delivery scheme [09] creates a delivery multicast tree and is able to combine the Patching and Chaining policies. P2VoD [07] introduce the concept of generation, which groups a set of clients in the information-propagation process. P2VoD and P2Cast present P2P VoD systems with fault-tolerance mechanisms based on the recursive reconstruction of the delivery tree. Neither of them evaluates the cost involved in the failure management process, which is very important, since the system presents restrictions in order to maintain the QoS.

In [05] [06], we proposed a P2P delivery policy that was able to synchronize a set of clients to collaborate with the server. However, [05] [06] does not provide a client-failure recovery mechanism. In [08] the authors present a study of the P2P paradigm applied to file sharing, where they show the significant amount of heterogeneity in this kind of system. The observations lead to a set of problems that we must also take in account for VoD systems.

In this paper we propose a failure management process, based on a P2P multicast delivery scheme, named *PCM/MCDB*. This scheme allows clients to collaborate with the server in a distributed way to send video information using multicast channels. The scheme is designed as two separate P2P policies. The first policy (PCM) creates multicast channels from the server to send video information and indicates collaborative clients for patching the missing portion of video. In the second policy (MCDB), we introduce the idea of a multicast channel branching where a group of clients is synchronized to generate local network multicast channels (branches).

Unlike traditional P2P schemes, client failures in the new delivery scheme do not immediately affect the QoS. We developed an advanced client failure detection mechanism in which each client in a collaboration group monitors neighbouring clients. The failure mechanism is able to detect a client failure before delivery disruptions occur. Once a client failure is detected, a centralized failure recovery policy is triggered in the server to replace the failed client with another, providing continuous video playback without glitches.

In our study, the proposed centralized scheme is evaluated using an analytical model developed according to server, network and client parameters. A distributed scheme is also analysed in order to improve the performance of the VoD system in terms of server resource requirements and scalability. Our analysis is based on the overhead introduced into the network by the failure management process.

The remainder of this paper is structured as follows. Section 2 presents the key ideas behind our delivery scheme. The Failure Management Process, using centralized and distributed approaches, is analyzed in Section 3. In Section 4, the developed analytical models are presented. Performance evaluation is shown in Section 5. In Section 6, we indicate the main conclusions of our results and future studies.

2. P2P MULTICAST DELIVERY SCHEME

In this section we introduce the environment considered and present the delivery policies adopted. The control analysis is developed taking multicast islands into account. This means networks have routers with IP Multicast capability. Thus, the models developed are only applied in this environment in a first approach. The entire VoD system is composed of distinct multicast islands and the communication between the islands is through border routers using unicast channels.

2.1. Overview of P2P VoD Architecture

In a video service, video information is sent by the server through the network to clients. There are 3 main components that implement the VoD architecture: server, clients and network. The server design defines the data organization strategy, data retrieving process from the storage system and the data delivery process to the network interface. All these modules have to be designed in order to satisfy the soft real-time requirements of the video delivery process.
The client receives, decodes and displays the video information. Throughout the process, the client design includes buffers that temporarily cache received information for 3 purposes:
1) A portion of this buffer is used to achieve smooth playback. We call this portion the jitter buffer. The size of this portion is invariable and is mainly dependant on the video format and variations in network bandwidth. 2) The client caches video information from the delivery channels (delivery buffer). The size of the delivery buffer changes according to the delivery policy. 3) All the client buffer that is not used for the previous 2 purposes will be utilized in the client collaboration. We call this portion of buffer “collaborative buffer” and it is able to cache video information for sending to another client.
Clients are connected to the VoD server through the network. In our design, we assume that the network is segmented and each client is able to maintain independent communication with other clients. We also assume that the local client is able to deliver video information to the local network using the multicast technique.
Video information is assumed to be encoded with a Constant Bit-Rate (CBR). The video information is delivered in network packets and the packet size is invariable. We call a network block a video block. We enumerate the blocks of a video from 1 to L, L being the size of a video in video blocks.

2.2. PCM and MCDB Policies

The delivery scheme decides how the video information is sent to clients. Our delivery scheme is designed based on two policies (Figure 1): (a) *Patch Collaboration Manager* (PCM) and (b) *Multicast Channel Distributed Branching* (MCDB).
The objective of PCM is to create multicast channels to service groups of clients, and allows clients to collaborate with the server to deliver portions of video in the admission process. With PC**M,** clients receive video information from both a multicast and unicast channel. The multicast channels are created by the server, whereas the unicast channels could be created either by the server or the clients. Multicast channels deliver every block of a video while unicast channels only send a portion of a video. We call the multicast channel a Complete Stream and the unicast channel a Patch Stream (Figure 1 a).

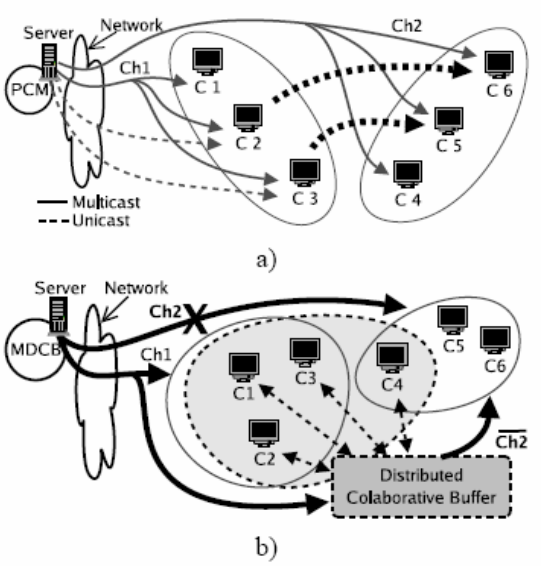

Figure 1. Delivery Scheme: a) PCM Collaboration. b) MCDB Collaboration.

The objective of MCDB, however, is to eliminate multicast channels so as to reduce server load. The policy replaces an on-going multicast channel with a local multicast channel. A group of collaborative clients is synchronized to form a Distributed Collaborative Buffer. Clients of this group use their buffers to cache video blocks from another multicasting channel. The cached blocks are delivered by the collaborative clients in order to generate the local multicast channel. When a multicast channel is replaced by one generated with collaborators clients, we call the new channel a branch channel (Figure 1 b).

3. NODE FAILURE MANAGEMENT PROCESS

In VoD systems, failures can be caused by a network failure, a client machine crash or even VCR operations. Furthermore, in P2P based systems, peers come and leave freely, so a client departure can be faced as a failure for the system. In such a situation, the client stops sending video information that can degrade the QoS. To address the problem of a failed client, we use a Failure Management Process based on three components: failure detection, recovery and maintenance of the system's information coherence.

3.1. Failure Detection

Client failure detection supposes that a collaborator suddenly leaves the system. The MCDB associates each client with 2 neighbouring clients in accordance with the client position in the distributed circular buffer. For instance, in Figure 2, client C2 has clients C1 and C3 as neighbouring clients. Each client periodically receives 2 synchronization messages from its neighbours. The messages notify the state of the neighbours and, if one of the neighbours has failed, the client sends a control message to the element responsible for starting the recovery process. This detection mechanism is able to detect a failure in advance because only the client, in the collaborative group, that is sending the video information affects the quality of the branch-channel. In Figure 2, a failure of client C2 does not affect the quality of branch-channel until client C3 finishes delivering block 13 and 14. This approach to failure detection is distributed, once every node receives messages from other nodes.

3.2. Failure Recovery

In PCM/MCDB policy, the recovery process is centralized in the server and is triggered when a client failure is detected. The centralized approach supposes a simpler design and can represent an adequate and efficient solution for a range of multicast applications, since the server is responsible for all the processes and their steps. However, a centralized architecture has obvious implications, such as server load or the fact that a single controller represent a single point to manage all nodes' failure operations, and if this crashes, the whole fault tolerance scheme is lost. To address these problems we evaluate the centralized scheme and propose a distributed one and analyse its performance.
The recovery process defines different recovery actions according to the state of a failed collaborative client. In the MCDB P2P delivery process, a collaborative client can be in 4 states: 1) one client in the group caches the video information from other multicast channel. 2) a client could be waiting to start the delivery process. 3) One client is delivering video information to the branch-channel. 4) Clients could be waiting to start the caching process. In Figure 2 a), C1 is caching, C2 is waiting to start delivering, C3 is delivering and C4 is waiting to start caching.

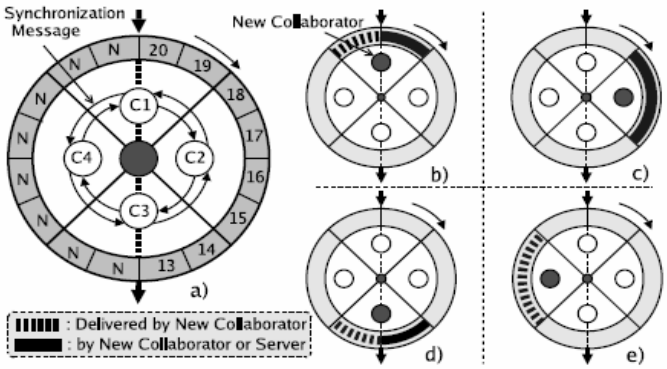

Figure 2. Client Failure Recovery: a) Advance Failure Detection. b) Recovery Process of State 1. c) Recovery Process of State 2. d) Recovery Process of State 3. e) Recovery Process of State 4.

State 1 Caching: The failure recovery policy tries to find a new collaborative client to replace the failed client. The new collaborator continues with the cache process of the failed client. The video blocks that are already cached by the failed client will be delivered by the new collaborator, if the collaborator has these video blocks. For example, in Figure 2 b), C1 fails and the new collaborator will cache blocks 21 and 22. Block 19 and 20 will be sent by the server or by the new collaborator.
State 2 Waiting for delivering: The recovery action for this state is quite similar to the case for state 1. However, the new collaborator or server has to send all the video blocks that are cached by the failed client (Figure 2 c).
State 3 Delivering: In this case, the recovery process finds a collaborator to replace the failed client, just **as** in state 1. The new collaborator does not need to have the cached information of the failed client, but has to have the same collaborative buffer capacity. In case the new collaborator does not have video information, the server assumes the delivery process of the failed client temporarily. For example, if C3 in Figure 2 d) fails, and the recovery process is unable to find a new collaborator that has blocks 13 and 14, the server will continue sending blocks 13 and 14.
State 4 Waiting for Caching: The failed client has no useful information in its buffer, so the recovery process only needs to find a new collaborator to replace it (Figure 2 e).
A client failure in states 1, 2 and 4 does not immediately affect the branch channel's delivery process. Thus, the recovery process could take place with a certain delay without affecting the

delivery process. However, if the failed client is in state 3, the delivery process will be immediately disrupted. Since the client failure detection mechanism needs a period of time to trigger the recovery policy, clients of the disrupted channel will not receive any information before the recovery process. In order to avoid glitches, we delay the playback for a short period to create a cushion buffer. This cushion buffer provides video information until the end of the recovery process.
In PCM, the collaborative client is always sending video blocks, so the collaborator only has a state like the 3 in MCDB.
Beyond the margin that the cushion buffer gives the system, in the case of the MCDB policy, we have a more flexible situation in 75% of cases, because the clients' disruption does not immediately affect the transmission (states 1, 2 and 4). However, in order to keep the system working with QoS, we assume the critical cases where an efficient solution is required, which occurs when a node is sending video blocks (state 3 and PCM policy).
In a general sense the recovery stage depends on the system's failure frequency (f_e), the message changing to trigger the recovery process and the message changing between routers in order to maintain or rearrange the distribution tree.

3.3. Maintenance of the System Information

This stage of the Failure Management Process is not connected directly to the other two, in the sense of not being part of the logical sequence, but it is as important, because it has a great influence on the recovery process.
The maintenance of information coherence means that the system must keep the state of the nodes with certain precision, whether they are available or not and all the data needed to perform a set of possible collaborators.
Out of date information, originated by a bad maintenance process, can lead to a wrong collaborator selection, which involves an answer refused by the indicated peer. Thus, a new search for collaboration must be triggered and the time to solve a node failure will increase.
A centralized scheme to maintain node information up to date supposes the server receiving information messages from all active clients on a receive frequency (f_{CI}). In the other hand, a similar process is needed by the routers for the multicast distribution trees, that is, each one sending and receiving messages, in a distributed way, to maintain or update the multicast tree.

4. COSTS OF FAILURE MANAGEMENT PROCESS

To improve the performance of the Failure Management Process, we propose the analysis of two different approaches for the process: a centralized and a distributed one. In order to evaluate both approaches, we analysed the volume of control messages injected into the network to achieve all the three phases, Detection, Recovery and Maintenance. This *overhead* metric represents the cost of each scheme. In this section, we formalize the overhead for each part of the failure management process. For convenience, the parameters used in the analysis are defined in table 1.
The PCM/MCDB already assumes the Detection phase in a distributed approach, so, we define its cost, in function of the Overhead. It is determined for the send frequency of *heart beat* messages and for the number of messages needed in this process, which must be changed for all nodes in every existing group. The cost of detection is therefore given by:

$$C_{\det ection} = f_{HB} \cdot \beta \cdot \sum_{i=1}^{G} N_{C\_g(i)} \qquad (1)$$

The Recovery and Maintenance phases are proposed in a centralized way for PCM/MCDB. Therefore, in the following sections, we analyse these centralized approaches and their associated costs and present the decentralized one, with its respective costs.

$C_{overhead}$	Number of messages injected into the network.
$C_{detection}$	Number of messages injected into the network in the detection phase.
$C_{recovery}$	Number of messages injected into the network in the recovery phase.
C_{maint}	Number of messages injected into the network in the maintenance phase.
N_C	Number of active clients in the system.
H	Number of clients that trigger a recovery process.
G	Total Number of multicast groups.
$HOPS_{g(i)}$	Number of hops for each multicast group G.
p_s	Probability to find a collaborator.
f_{HB}	*Heart beat* messages frequency.
f_e	Faults occurrence frequency.
f_{CI}	Client communication messages frequency.
f_{TI}	Router communication messages frequency.
β	Number of messages required for the detection protocol.
σ	Number of messages required between clients for the recovery protocol.
γ	Number of messages required between routers for the recovery protocol.
ω	Number of messages required between clients for the maintenance protocol.
α	Number of messages required between routers for the maintenance protocol.

Table 1. Parameters used in the analysis

4.1. Recovery

The Recovery process depends on client communication to trigger the process. Messages are sent to start the process and an answer is received, so the new collaborator's connection can be performed. On the other hand, a communication between routers that implements the IP Multicast is also necessary to arrange the distribution tree. This process is inherently distributed, since it is a question of routers, but communication between clients can be taken as a centralized or distributed approach.
Centralized: The centralized scheme supposes all faults, which trigger a Recovery process, make clients contact a central server. This server is responsible for performing a search based on clients' information. The search should select the most suitable collaborator to substitute the failed one. After selecting an adequate candidate, the server contacts the nodes implied in the Recovery and the new collaborator, to perform the link. So, the Overhead cost, considering router communication is given by:

$$C_{recovery} = f_{e.} \cdot \left[\sigma \cdot H + \sum_{i=1}^{G} \gamma \cdot Hops_{g(i)} \right] \quad (2)$$

Distributed: The distributed scheme assumes the triggered Recovery process is managed autonomously by its own nodes. We define a Manager Node per Multicast group, which is responsible for keeping information about the group members.
The Manager Node is a client that has the responsibility for managing a Multicast group, because it has full member information. The selection of this node is performed based on its history in the system and its capabilities, such as buffer size, process capacity and available bandwidth. A hierarchy is established in the Multicast group, in order to enable attribution of the function of Manager for another node in a set of nodes, in case the Manager fails.

On a first attempt, the Manager Node receives the recovery query and searches a substitute node in its clients' group information list. This phase verifies the existence of a candidate with the necessary characteristics to substitute the failed peer. If there is a node capable in the group, the linking process is performed; otherwise, the Manager Node contacts the Manager Node of another group, and asks for a qualified candidate to replace the failed peer. This process is repeated for all groups, until a new collaborator is found, always respecting a threshold time in order to maintain the QoS. After selecting an adequate candidate, the contact between the nodes that query for Recovery and the new collaborator is established, to perform the join. Thus, the Overhead cost for this case, considering router communication, is given by:

$$C_{rec.} = f_{e.} \cdot \left(\frac{H \cdot (1 + \sigma \cdot p_s)}{p_s} + \sum_{i=1}^{G} \gamma \cdot Hops_{g(i)} \right) \qquad (3)$$

4.2. Maintenance

The Maintenance means that the system must keep node information (content, buffer size, bandwidth, etc.). The exactness of this information determines how successful the recovery process will be.

To provide the set of possible collaborators with up-to-date information, messages are exchanged between the clients or between the clients and the server, depending on the scheme adopted, centralized or distributed. We evaluate these two approaches below. The Maintenance process also needs communication between the routers that implement the IP Multicast in order to maintain or rearrange the Multicast groups. This process is inherently distributed, given that it is a question of routers.

Centralized: The centralized scheme consists of clients sending periodic messages to a central server to inform about their characteristics. The server analyses the information and creates lists with a set of possible collaborators. Therefore, the Overhead cost, considering router communication, is given by:

$$C_{ma\,\mathrm{int}} = f_{CI} \cdot \omega \cdot N_C + f_{TI} \sum_{i=1}^{G} \alpha \cdot Hops_{g(i)} \qquad (4)$$

Distributed: The distributed scheme supposes that clients inside a Multicast group exchange messages periodically to inform about their characteristics. In this case, there is no central point that contains all the node information. All peers in a Multicast group send messages to the Manager Node, who analyses the information and creates lists with a set of possible collaborators. The process is the same for all groups. Thus, the Overhead cost for this case, considering router communication, is given by:

$$C_{ma\,\mathrm{int}} = f_{CI} \cdot \left(\sum_{i=1}^{G} \omega \cdot N_{C\_g(i)} + G^2 \right) + f_{TI} \cdot \sum_{i=1}^{G} \alpha \cdot Hops_{g(i)} \qquad (5)$$

4.3. Unicast Cost Model

In order to evaluate the influence of the transmission scheme, we developed a model that represents the cost on the three process' phases, considering the unicast transmission scheme.

We adopted the centralized failure management process as background. Like the IP Multicast case, we define an analytic model to represent the cost of a centralized failure management process in a unicast environment. So, the expression for the cost is given by:

$$C_{overhead} = (f_{HB} \cdot N_C \cdot \beta) + (f_{e.} \cdot \sigma \cdot H) + [(f_{CI} \cdot \omega \cdot N_C) + (f_{TI} \cdot \alpha \cdot L_u)] \quad (6)$$

The parameters considered for the model are the same that was used to modelling the failure management process in the multicast environment. The difference between the multicast and unicast models is on the recovery and maintenance phases. On the unicast model, there are no groups; therefore there is no need to restructure the distribution tree when a failure occurs. On the maintenance phase, it's not necessary keep groups' state, but routes are up to date dynamically, based on system's characteristics. The unicast scheme creates a point-to-point communication channel, in which the information flows. In this way, the routers must periodically change messages and process the calculations of the routing algorithms. In our work, we calculate a mean route (L_u) based on the adopted topology, showed in figure 3 and given by:

$$L_u = \frac{\sum_{i=0}^{M-1} (M-i) \cdot 2^{(M-i)}}{\sum_{i=0}^{M-1} 2^{(M-i)}} \quad (7)$$

5. PERFORMANCE EVALUATION

The developed models are evaluated adopting a binary tree router topology. This topology is shown in Figure 3 and has seven levels ($M(i) = [1, 7]$). Values are attributed to the parameters on the centralized and the distributed schemes.

Each tree level has $2^{M(i)}$ routers, therefore the total number of routers in this topology is 254. We consider that each hop has an associate network that is limited to connect a maximum of 120 clients. The number of possible active clients in the system is:

$$N_C = \left(\sum_{i=0}^{M(7)-1} 2^{(M(7)-i)} \right) \cdot 120 = 30.480 \quad clients \quad (8)$$

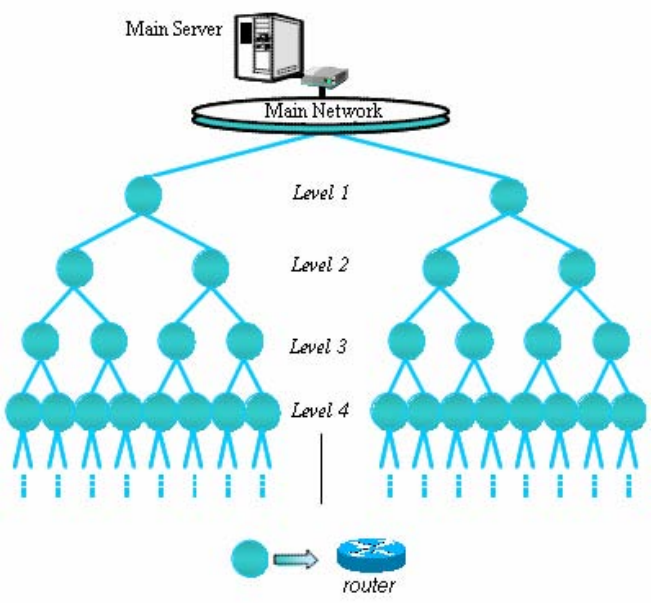

Figure 3. Binary Tree Topology

A single recovery request is considered (H = 1) and the probability of finding a collaborator (ps) was varied between 5%-95%. We verified that this parameter has little influence on the total messages amount, because the number of messages necessary in this communication is much lower than the number involved in the other phases of the process.
We are considering the PIM-SM protocol for the multicast implementation in the routers, because recently it is one of the most used in IP multicast. The number of messages in the detection, recovery and maintenance protocols were assumed as one (γ, ω, α) and two (β, σ). The frequency of sending heart beat messages (f_{HB}) is 1 every 5 seconds. We consider that the status messages (f_{CI} and f_{TI}) are sent with a frequency of 1 every 15 seconds [10] [11] [12].
We observe the behaviour of the centralized and the distributed approaches for three different parameters: the number of multicast groups, the quantity of clients in the system and the frequency of errors occurrence. The evaluation is made in the sense to measure the cost increment that the distributed scheme presents when it's compared to the centralized cost. This incremental cost is represented like a percentage, how defines the following expression:

$$\Delta = \frac{(C_{dist.} - C_{cent.})}{C_{cent.}} \cdot 100 \qquad (9)$$

The number of multicast groups is varied between 20 and 200 groups. The others models' parameters are fixed, such as to all next evaluations done. A multicast group can contain clients that are visualizing the same video and that arrived in the time interval necessary to join in a multicast channel. A multicast group also can be P2P collaboration groups, which share resources between clients. In figure 4 we can observe that the amount of messages grows with the number of the multicast groups in the system. The difference Δ, increases as the number of groups grows. This is caused because the distributed scheme for failure management, considers the communication of the Managers Nodes. In the maintenance phase there is a term G^2 that represents the communication between groups. The increasing Δ is caused by the decentralization policy adopted, that supposes groups communications, nevertheless this doesn't means that a distributed scheme is not scalable, how we analyse in the following evaluations.
In order to evaluate the scalability of the system when the quantity of clients grows, we vary this parameter and observe the behaviour of the message cost. In figure 5 is possible to verify that Δ decreases as the number of clients grows in the system. This diminution occurs because the communication between groups has less importance when compared with the amount of messages originated for the growing quantity of clients. It means, for a low number of clients dispersed in multicast groups the communication between Managers Nodes has major importance, nevertheless when the groups are more dense, the messages generated for the clients assumes more importance, so the difference Δ between the centralized and the distributed scheme decreases.
In LVoD systems that uses the P2P paradigm, the clients connect and disconnect with a certain frequency. The figure 6 shows the influence of the increment in the failure frequency. Each failure triggers a recovery process, so the cost increases. The difference Δ among the centralized and distributed schemes grows with the failure frequency. Although, in systems with characteristics like LVoD system, members join to view a specific content, which in general is not short, so the lifetime of a client in the system can be considered around 300 seconds, in the worst case [11]. In the case of 1 failure every 5 minutes the Δ is 5.7%, what doesn't represents a big overhead.
In order to evaluate the influence of the transmission schem, multicast and unicast were compared. The figure 7 represents the behaviour of the cost, in a centralized scheme of failure management, considering multicast and unicast transmissions. This analysis can show the influence of the communication between routers. The multicast diffusion requires routers' communications in order to maintain and construct distribution trees, in the other hand routers using unicast only needs

change messages periodically in order to up date the routing table. The increment in the number of clients in the system leads to a diminution in the Δ among multicast and unicast. This behaviour occurs because when the number of users grows, the messages interchanged by the routers have less importance if compared with the amount of messages generated by the clients.
These results shows that a distributed control scheme causes an increment in the network load, which can be considered acceptable in some cases, since the server is free of fault control, saving system resources that can be applied to any other function and the system has no single point for managing failures.

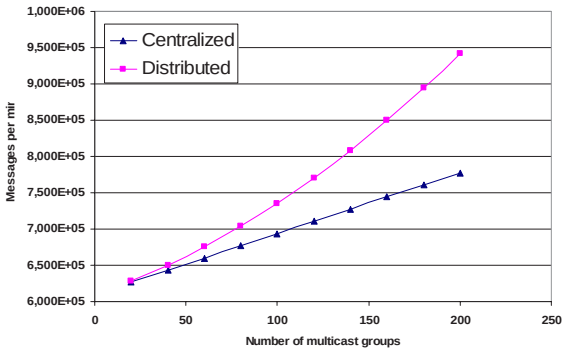

Figure 4. Influence of the number of multicast groups

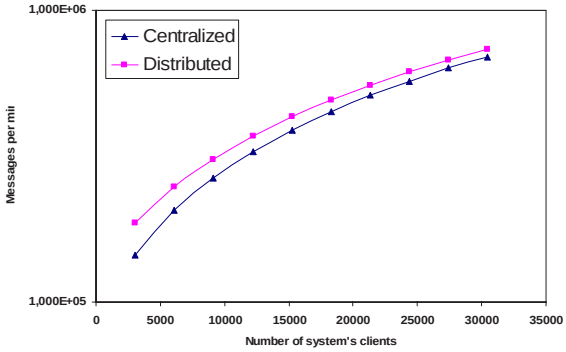

Figure 5. Influence of the number of clients

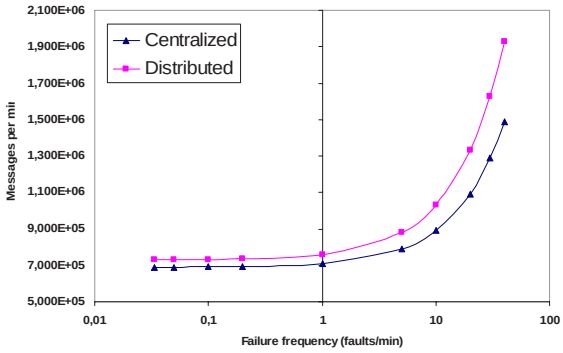

Figure 6. Influence of the failure frequency

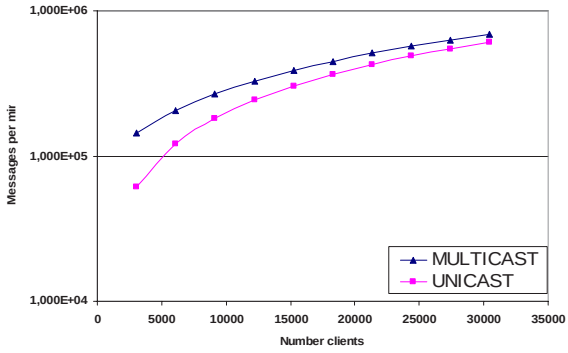

Figure 7. Multicast vs. Unicast

6. CONCLUSIONS

We analysed the cost of the failure management process, based on the volume of control messages, in the PCM/MCDB P2P multicast scheme. Analytic models were developed to represent the behaviour of centralized and distributed schemes, and to represent multicast and unicast communications. A system topology was defined in order to evaluate both schemes.
For large systems with many hops and clients, the distributed approach presents an inherent increase in messages number. Nevertheless, this increment could be assumed since the distributed approach saves system resource, frees the server of control load and creates multiple points to manage failures. Its applicability depends of tuning some parameters, like number of multicast groups, number of hops in the groups, or even defining others communications' protocols for a distributed failure management process.
Thus, a distributed control scheme can present many advantages, including a more feasible system, more scalability and resource saving, in exchange for some increase in the network traffic.

We have started several future research projects. First, our objective is to implement the analytic models in a simulator and compare the results. More research will be needed in order to evaluate the control schemes, and find the most suitable. Finally, we are studying the application of a control structure, composed for policies and maybe for dedicated elements to provide LVoD systems.

REFERENCES

[1] Cai, Y., Tavanapong, W., Hua, K. A. Enhancing patching performance through double patching. Proceeding of 9th Intl Conf. On distributed Multimedia Systems, 2003

[2] Hua, K. A., Cai, Y., Sheu, S. Patching: A multicast technique for true video-on-demand services. ACM Multimedia Conf, 1998.

[3] Hua, K. A., Sheu, S., Wang, J. Z. Earthworm: A network memory management technique for large-scale distributed multimedia applications. Proceedings of the INFOCOM '97, 1997.

[4] Jin, S., Bestavros, A. Cache-and-relay streaming media delivery for asynchronous clients. Proceeding of NGC'02, 2002.

[5] Yang, X. Y., Hernández, P., Cores, F., Ripoll A., Suppi, R., Luque, E. Distributed P2P Merging Policy to Decentralize the Multicasting Delivery. Proceeding of 31st EuroMicro Conference, 2005.

[6] Yang, X. Y., Hernández, P., Cores, F., Ripoll A., Suppi, R., Luque, E. Dynamic distributed collaborative merging policy to optimize the multicasting delivery scheme. Euro-Par, 2005.

[7] Do, T., Hua, K., Tantaoui, M. P2vod: providing fault tolerant video-on-demand streaming in peer-to-peer environment. Communications. IEEE International Conference, 2004.

[8] Saroiu, S., Gummadi, P. K., Gribble, S. D. A Measurement Study of Peer-to-Peer File Sharing Systems. Proceedings of Multimedia Computing and Networking, 2002.

[9] Guo, Y., Suh, K., Kurose, J., Towsley D. P2Cast: P2P Patching Scheme for VoD Service. Computer Science Technical Report 02-34, 2002.

[10] Wang, X., Yu, C., Schulzrinne, H., Stirpe, P., Wu, W. IP Multicast fault recovery in PIM over OSPF. Proceedings of ACM SIGMETRICS, 2000.

[11] Silverston, T., Fourmaux, O. Measuring P2P IPTV Systems. ACM NOSSDAV, 2007.

[12] Tarik C., S. Gjessing and O. Kure. Tree Recovery in PIM Sparse Mode. In Telecommunication Systems 19:3,4, 443–460, 2002.

[13] Yang, X. Un Sistema de Vídeo-bajo-Demanda a gran escala basado en la Arquitectura P2P con Comunicaciones por Multidifusión. PhD thesis, Universitat Autònoma de Barcelona, 2006.