An Implementation of an Overlay Network Architecture Scheme for Streaming Media Distribution

Ch. Z. Patrikakis, Y. Despotopoulos, A. M. Rompotis, N. Minogiannis, A. L. Lambiris, A. D. Salis
Telecommunications Laboratory of the National Technical University of Athens
Heroon Politechniou 9, Zographou, Greece 15773, tel: +30 210 7721513, fax: +30 210 7722534
{bpatr,ydes,arobot,minogian,biril,tsalis}@telecom.ntua.gr

Abstract

In this paper we introduce the implementation of a streaming video distribution scheme based on client relay modules. The purpose is the formation and maintenance of an overlay network architecture responsible for the dispensation of streaming traffic to end-clients. This architecture has been based on the use of modular system components that can accommodate the integration of existing commercial solutions for media reproduction such as video players (used in the implementation as black box components). The result is a system design capable to manage and sustain a media distribution scheme based on an overlay network infrastructure. The presented implementation has been developed in the context of the EU IST OLYMPIC project and is part of a large network architecture for supporting personalised multimedia distribution for covering major athletic events [1].

1. Introduction

For over a decade, researchers have spent considerable effort on the design of protocols to support broadcasting for efficient many-to-many communication. IP Multicasting [2] was introduced at late 80's as an extension to the IP dominant protocol in order to meet the increasing bandwidth requirements of specific applications such as real time/video on demand services. To overcome fundamental problems related to IP multicasting "global" deployment, research has turned to other solutions based on application layer data forwarding and group communication services [3]. As a consequence overlay architectures have been proposed for supporting data distribution and in some cases used for serving streaming applications.

The rest of the paper is organized in three parts as follows: First, the reason and motivation for deploying overlay network architectures for supporting media distribution is provided. Next, the integration and combination of such solutions for providing an end to end distribution scheme is presented.

Following these introductory sections, the paper continues with the presentation of the architectural model, presenting the framework on which the implementation was based. The next section is providing the details of the architecture both at functional and implementation level.
Finally, the last two sections present the trials performed so far, together with the results, conclusions and future work that will be integrated in the proposed architecture.

2. Necessity and motivation of overlay solutions

The IP Multicast service was proposed as an extension to the Internet architecture to support efficient multi-point packet delivery at the network level. With IP Multicast, a single packet transmitted from a specific source is delivered to an arbitrary number of receivers by replicating the packet within the network at fan-out points (routers) along a distribution tree rooted at the traffic's source. Although IP Multicast uses the resources of the network quite efficiently, its deployment has been slowed by issues related to scalable inter-domain routing protocols, charging models, robust congestion control schemes [4]. Therefore, the existing multicast model targets mainly in supporting the communication needs of large groups and is usually limited within areas covered by the same provider [5][6].

Reckoning the aforementioned, it is clear that there is a necessity to propose other solutions that will succeed in transmitting the same data to multiple users overcoming all these problems. Because of the problems encountered during the deployment of a network-level multicast service, many recent research proposals have agreed on an *application-layer* multicast either unicast service and have described solutions for such a service and its applications [7][8][9][13].

Proceedings of the 29th EUROMICRO Conference "New Waves in System Architecture" (EUROMICRO'03)
1089-6503/03 $17.00 © 2003 IEEE

The alternative solution proposed is based on moving the data replication and distribution schemes to the network periphery, using application based multicast models, supported by unicast transmission towards the peripheral distribution points [10][12]. This methodology results to a virtual layer built above the network infrastructure and each of its edges corresponds to a unicast path between two end systems in the underlying internet.

The general notion is that applications are self-organized into a logical overlay network, and transfer data along the edges of the overlay network using unicast transport services. The overlay network is built as a graph with properties so that spanning trees can be easily embedded without the need for a routing protocol, e.g., as a hypercube [8]. Application-layer multicast has a number of appealing features:

- There is no requirement for multicast at the network layer infrastructure or allocation of a global group identifier (such as the IP multicast address).
- Since packets are flowed over the virtual layer via unicast, flow control, congestion control, and reliable delivery services available for unicast transmission can be exploited.
- Adaptability: the overlay network can be dynamically optimized.
- Robustness: increased control and adaptable nature make the overlay network more robust.
- Customization: the design and construction criteria of overlay network can be based on the requirements of the application.

However, application layer multicast has some significant drawbacks. Since data is forwarded between end-systems, end-to-end latencies can be large. In addition, if multiple edges of the overlay architecture are mapped to the same network link, multiple copies of the same data may be transmitted over this link, resulting in an inefficient use of bandwidth [8][9]. Thus, important performance metrics for overlay network topologies deploying application-layer multicast should be applied in order to reduce end-to-end latencies and to optimize bandwidth allocation.

The majority of these proposed solutions typically involve members of a multicast group to organize into an essentially random application-level mesh topology over which a traditional multicast routing algorithm such as DVMRP (Distance Vector Multicast Routing Protocol) is used to construct distribution trees (exceptions are TBCP, YOID algorithms). Routing algorithms require every node to periodically announce its estimated distance from every possible destination to its local neighbours; hence every node maintains state for every other node in the topology. Further, in the case of a change in the topology, every node must be informed about this change and update its routing table if required [10].

Overlay multicast solutions make the deployment of broadcast functionality easier as they implement their functionality entirely at IP hosts and require no modifications at the core network routing technology. With this wide range of broadcast-capable solutions, we have ended up with the scenario where a number of diverse and in some cases non interoperable protocols co-exist in the Internet. No single protocol has been deployed globally. In fact intention is not the adherence of a specific overlay protocol over the Internet. This is due to a variety of reasons including technical shortcomings in the protocols and their implementations, the fact that some protocols are geared towards specific applications and a range of business model concerns. Actually, the Internet landscape is likely to be fragmented into potential overlapping clouds of broadcasting/multicasting connectivity with no interoperation across these clouds.

3. Integrating and combining overlay solutions for end to end media distribution

An important advantage of deploying an overlay architecture (through an Application Layer Multicast mechanism) for distributing streaming media is its built-in adaptation in Content Delivery Networks. The concept behind such a network model is to push the content to the edge of the networks and deliver the content with massive intelligence and manageability. Since abundant network bandwidth is usually available in the network periphery, we can then relay the streaming distribution to the edge of the network. The purpose is to serve all the clients attached under the same local network or domain on the edge, rather than leaving the content distribution fermentation at the backbone.

A CDN is a representative overlay network to the Internet which has been built specifically for the high-performance delivery of common web objects, static data and rich multimedia content. CDN's functionality is based on layer-4 switching forming an overlay network architecture capable of providing dedicated services such as streaming delivery [19] . The aforementioned overlay solutions can be integrated in CDN implementations as long as they remain transparent to the end-users and the network infrastructure. The interoperability issue between overlays and CDNs (representative overlay mechanism) should be confronted when a Provider decides to combine these techniques but as long as the overlay network and the CDN operate independently, probably a simple solution of central overlay node administration and management can be adopted.

The most attractive feature of deploying overlay networks in serving streaming media applications is the fact that these architectures do not require any modifications at the network layer infrastructure shifting the burden of transported traffic to higher levels. Illatively, application layer architectures, acting complementary to Content Delivery Networks, can be deployed at the edges providing a further expansion to the overall content distribution network [13]. According to the adage "keep simple the core and move the complexity to the edge" the content delivery procedure takes place in access points without infecting the

Proceedings of the 29th EUROMICRO Conference "New Waves in System Architecture" (EUROMICRO'03)
1089-6503/03 $17.00 © 2003 IEEE

backbone network and thus relieving it from the increased streaming traffic. In the next section we provide the description of the overlay architecture mentioning as well its adaptability to various networking conditions and its interoperability with commercial CDN implementations.

4. The overlay network distribution model

The implemented platform provides an overlay solution for streaming media distribution by using a relay scheme based on peripheral reflectors or simple clients. Such schemes have been proposed in the past based on the use of several proposed architectures for overlay network architecture creation and management [7][8][12]. However, the presented implementation does not rely only on the existence of dedicated relay nodes, but makes use of the media receiving clients as possible relay points.

The whole implementation has focussed on presenting a solution that can be integrated seamlessly with existing opensource and commercial solutions for both media server and media clients. In this context, the media source is considered a black box, having as unique requirement the compliance to the standards of RTP protocol and distributed media formats. The same applies for the media client, in which several commercial solutions have been tested, in combination with new software located on the client host in order to provide full client relay capabilities. The result is the provision of an overlay distribution scheme that can be used over every existing scheme (including other overlay solutions such as Content Delivery Networks) and in which the participating nodes may decide whether or not to make use of the client relay functionality, as it is presented in Figure 1.

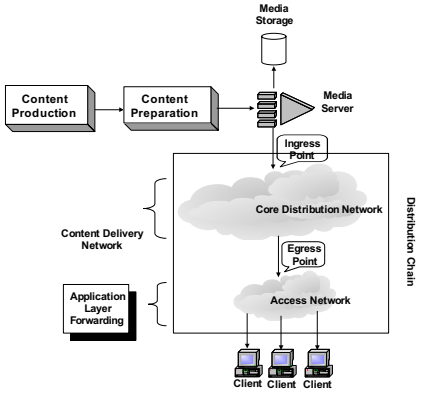

Figure 1 : Media production and distribution

This provides a flexible architecture that can be offered in two different flavours: As a complete end to end scheme in which a central Media Server is distributing information to clients forming an independent overlay distribution architecture, or as an overlay distribution scheme suitable for distribution of media over access networks, in combination with other overlay solutions such as CDNs. In this case, the Media Server (or even several Media Servers) is considered the connecting point between the two complementary overlay implementations (egress point for the CDN). The next figure presents the aforementioned interoperation between CDN and application layer forwarding mechanism. CDN is applied in the core while application layer forwarding solution is applicable to the access.

This leads to the media distribution scheme presented in Figure 2. In this figure, the rectangles represent a Media Server (MS) or a media Relay Server (RS), while the circles represent media clients (C).

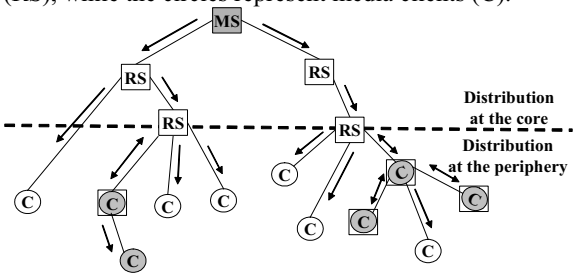

Figure 2 : Media distribution scheme

Note that we have three types of clients:

1. Plain media clients that operate in the normal way, receiving the media stream from a designated MS (or RS). These clients are represented by white circles.
2. Media clients that though they make use of the client relay scheme for receiving the media do not participate as relay nodes for further distribution to other clients. These are represented by grey circles.
3. Media clients that not only make use of the client relay scheme for receiving the media, but actively participate as relay nodes for other clients. These are represented by grey circles surrounded by a rectangle.

The implementation allows a client to choose not only if it wishes to participate in the process of media relaying at peripheral level, but also the level of involvement in this architecture: passive, in the case when it only uses the architecture for receiving media, or active when it participates in the architecture as a relay node

5. Functional description and implementation details

Before proceeding in the description of the participating nodes, we present the philosophy behind the implemented architecture. The idea is based on the use of client hosts that deploy both a media player (for local playback of the received stream) and a media server (for relaying the media to other clients) as depicted in Figure 3.

Based on this implementation, each client instead of using a direct connection to a media server, as it would in normal data distribution architecture, receives the distributed data through the use of a media server located on the same host, which in turn may be used to transmit this data to other clients. This distribution scheme is depicted in the data distribution plane of Figure 3, in which two types of clients are represented: a client operating in the normal way, connected directly

Proceedings of the 29th EUROMICRO Conference "New Waves in System Architecture" (EUROMICRO'03)
1089-6503/03 $17.00 © 2003 IEEE

to the server, and two clients using the relay mechanism.

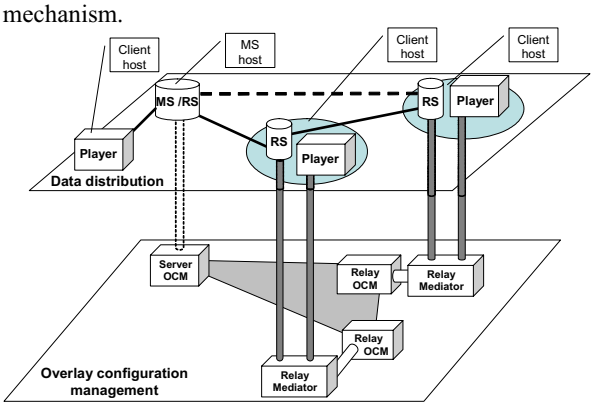

Figure 3 : Implementation components

Each of these clients has a local media server acting as RS, which is the entity responsible for contacting the media server. The media players in each client get the requested information using the local servers as relay nodes. We must note that a major requirement was to treat the players as black boxes, therefore no modifications on the player applications was performed. Evaluation was made with both open-source and commercial players. This ensures that the provided solution is open and can accommodate non custom software for media reproduction on the clients.

5.1. Functional description

5.1.1. Relay Mediator (RM). To provide the means for diverting the initial client request for a specific media file from a specific location on the network to the local relay server and to instruct the server to retrieve this data and relay it to the requesting client, a relay mediator is used. Its goal is to capture the initial request provided by the user who is unaware of the underlying topology, and to trigger the relay mechanism in order to receive (and retransmit upon request) the requested stream. Based on the decision the solution should be compatible with all existing player implementations, a special scheme for requesting the media file, compatible with all existing implementations was devised. In this, the user provides information about the requested media to the media player software through the use of a front end that instead of transforming the user request for a "MediaFile" from a MediaServer" to the standard: rtsp://MediaServerAddress/Mediafile, provides the player with the following request: rtsp://localhost/Mediafile?MediaServerAddress.

Using this format, the media player is directing the request for the media file to the media server located on the same host (localhost). However, since the media is not already on the local media server, the latter needs to know the address of the server from which it will get the requested media, before relaying it to the player. For this, the relay mediator module is used to translate the last part "**?MediaServerAddress**" which passes transparently inside the media player's request in order to determine the source of the media. At this point the architecture is able to provide the means for receiving the media, through a local relay. However, the target is to receive the media through a different relay point by using the relay mechanism. In order to do so, the relay mediator is not directly requesting the media file from the address resolved from the Media ServerAddress information, but provides this information to the Overlay Configuration Management (OCM) module, which is responsible for the configuration and maintenance of the overlay architecture.

5.1.2. Overlay Configuration Management (OCM). To provide the necessary functionality for overlay architecture management, each node in the architecture uses a special module responsible for the configuration and maintenance of the overlay architecture. We can distinguish 2 types of this module.

The **relay OCM module** located in each client host is responsible for receiving the connection requests from the Relay Mediator and performing the necessary actions in order to determine the best suitable distribution node. This may either be a relay node or the original server from which the media was requested. In order to perform these actions, the Relay OCM module needs to communicate with other OCM modules and exchange information related to the creation of the distribution scheme. The procedures and messages used are described in the next section. Apart from the active role in the selection of the most suitable distribution node, the relay OCM module may have a passive role in cases it is contacted by another relay OCM module. In such a case, the relay OCM will simply send to the requesting relay OCM the necessary information in order to assist it in discovering the most suitable distribution server.

The **server OCM module** is usually located on the server and is responsible for administrating the overlay distribution scheme. This module is the central point from which the formation of the overall architecture is monitored, and to which all media requests are addressed. In order to keep this module independent, the implementation has led to a module separate from the media distribution server. This means that in fact the module has the ability to be placed in a different location from the media server.

5.2. Implementation details

5.2.1. RM Implementation. The implementation of the RM, was based on the QuickTime Streaming Server (QTSS) [14][15]. QTSS was chosen since it is an open-source standards-based streaming server. It is also scalable in that it offers a programming interface for creating modules, which allow developers to supplement or enhance the server's functionality. Using this API, modules may register to receive notification for certain events such as an incoming RTSP request. Upon reception of an event, a module may either handle it directly and/or pass it back to the server to allow the default processing to take place. Furthermore, QTSS supports the notion of relay nodes providing the perfect platform for building a node that relays a media stream from a source such as a media server to a client such as

Proceedings of the 29th EUROMICRO Conference "New Waves in System Architecture" (EUROMICRO'03)
1089-6503/03 $17.00 © 2003 IEEE

a QuickTime player. In this scenario QTSS acts as a client in relation to the source and as a source in relation to the client.

It is worth noting though, that QTSS lacks built-in functionality to set up a relay session on demand. It is not possible for a client to request a stream through a particular relay. Any nodes that should operate as relays must be appropriately configured through QTSS's administrative interface before the client request.

Dynamic relay-session set-up is handled by the Relay Mediator which is implemented as a QTSS module. The Relay Mediator monitors every incoming RTSP message. Upon reception of an RTSP DESCRIBE message for a particular stream a check is performed on the message's command line in order to assert whether it is in the standard format: *"rtsp://mediaserveraddress/mediafile"* or in the enhanced format: "*rtsp://localhost/mediafile?mediaserveraddress"*. In the first case the request is passed back to the local server and the default processing takes place. If the local server is configured as a relay node for the MS then streaming of the *mediafile* will commence and the client will receive the requested stream through the server. In the latter case the command line is parsed and the *mediaserveraddress* along with the *mediafile* parameters are passed to the Relay OCM as it will be explained in the next paragraph. Consequently the Relay OCM returns the address of the 'closest' node able to relay the requested stream. Based on this address the RM module configures the local QTSS to start relaying the stream from the given node. Furthermore a message is sent to the Relay OCM indicating that this node is as of now an active relay node for the particular mediafile.

5.2.2. OCM Implementation. The OCM modules are also implemented as QTSS modules. The following diagram (Figure 4) presents the messages exchanged during the request for a media file.

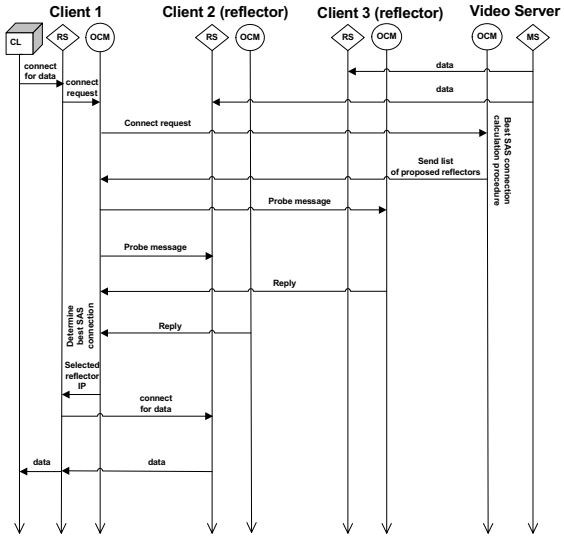

Figure 4 : OCM Message exchange diagram

As we can see, the request for media file from the client is directed to the local RM, which in turn is passing this request to the relay OCM module attached to it. This module will now initiate the relay node discovery procedure by sending to the server OCM module of the MS a connect request. The address of the MS is extracted from the enhanced media request format described earlier. The server OCM module, upon reception of this request will check on the list of available reflectors and based on a predefined set of criteria in order to perform a first level filtering of the nodes. The basic one is a special flag that indicates intention of each client to participate actively in the media redistribution process. If the user does not wish to offer his client workstation as a possible reflector, then the flag is turned down, informing the server that this node should not participate in the first level filtering described above.

The ones that will match the criteria set will be proposed to the requesting client through a response message containing the addresses of these RSs, as possible relay nodes. Upon reception of this message the relay OCM will initiate a probing procedure to each of the addresses contained in the reply from the MS. This probing procedure is based on the transmission of a "ping" like message to all RSs (including the MS). Once this message is received by the OCM module in each node, a reply is sent back containing the following fields:

- The roundtrip time (RTT) between the probing node and the RS. This time is used as a first metric of the distance between the two nodes.
- The connected clients to the RS. This is used as a second metric, in order to equalize the distribution of clients connected to relay nodes.
- The relay level. This is used along with the previous metric to control the uniform expansion of the tree in depth and width.
- The processing power of the RS. This is used as a third metric indicating the ability of each RS to act as a relay node.

Upon reception of the replies, the OCM on the requesting client initiates the selection procedure that takes into account the information received. Once the selection is done, the address of the selected RS is provided to the RS of the requesting client, and a connection between the requesting client RS and the selected RS is established.

After the establishment of the connection, the client's player is able to receive the requested media stream via the local RS, transparently from the user. At the same time, the OCM module sends to the server OCM (in the MS) that is administrating the media distribution process the following information:

- Its IP address and the address of the RS through which it receives the media stream.
- The QoS measured at application level. In terms of media, this information is provided from the RS and is the frame loss rate.
- An indication about the intention of the client to participate (or not) actively as a relay node for other clients
- A measurement on the processor occupancy of the client

Proceedings of the 29th EUROMICRO Conference "New Waves in System Architecture" (EUROMICRO'03)
1089-6503/03 $17.00 © 2003 IEEE

This information is also send periodically (once every 10 seconds) to the MS in order to keep it always updated about the status of the distribution scheme. The MS, based on this information is forming a distribution tree in which each node is represented by the necessary metrics presented earlier in the paper.

5.3. Deployment of multicast capabilities at local level

The process described so far is based on the use of unicast connections between the participating nodes of the architecture. However, in cases where multicast can be deployed at local level, this scheme is used for distributing the media stream from a RS to other clients within the same "neighbourhood" cluster. In such a case we can distinguish two scenarios according to the hops between the transmitter (MS or RS) and the receiver (RS):

The **first scenario** covers the case where the distance between the two nodes is greater than one hop. This is normally the case where the nodes are located in different LANs. In this case the intermediate routers need to be setup so as to support multicasting between the two endpoints. Such a communication scheme must be preconfigured statically, making this scenario appealing for solutions oriented towards the media source (MS – RS communication) or solutions that may serve large domains based on tree structures that spawn to more than one levels (super domains with many sub domains).

The **second scenario** covers the case where the two nodes are located in the same LAN (TTL distance equal to 1). In this case, multicast distribution of the information can be supported without any prerequisites on local router configuration. This scenario is ideal for supporting distribution between nodes that are located in the same LAN such as a small office, a department of a company or a Laboratory. This second scenario, being of more interest for our implementation has been integrated and tested through the use of the inherent capability of Darwin Streaming Server to support Multicasting.

In Figure 5, the process for enabling multicast distribution of the stream based on the second scenario is presented.

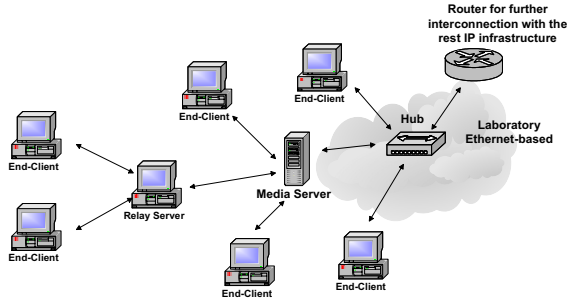

Figure 5 : Enabling multicast distribution of the stream

First,

- A multicast stream is sent to a group address. This means several client computers can tune in to the same stream.
- With a reflected multicast, the server receives a multicast stream, and then sends it to each client that tunes in to the stream.

Upon setting up the relay node (RS in our architecture) it listens to an incoming broadcast (either unicast or multicast), and forwards, or relays, the stream to one or more destination addresses. The destination addresses may be unicast or multicast and the server can be configured to relay multiple broadcasts at the same time using internal configuration files [15].

5.4. Recovery mechanism for responding to interruptions in the distribution chain

When a reflector starts, it registers itself with the OCM Server. The two modules retain a permanent TCP connection between them in order to avoid connection setup time and allow quick communication when needed. The cost of keeping up the TCP connection alive is minimal as the traffic carried is very sparse and messages exchanged are usually less than a hundred bytes.

This permanent TCP connection is also an effective mechanism to detect failures. As soon as this connection is closed, either because of a network timeout or a module failure, the server is immediately informed and can take appropriate action.

In the presence of a failure, as described above, there are many ways for the server to respond depending on the configuration. The simplest is to remove the reflector from all the relay trees in which participates together with all the reflectors that are children of the offending node, without taking any further action. This approach leaves recovery to clients. When users depending on the failed node discover that they no longer receive the video stream, they will try to reconnect, essentially repeating the procedure followed when they first joined the distribution tree. Since the server has already updated the relay trees, they will be redirected to a valid reflector. An enhancement of this technique is for the server to forcibly require from all affected nodes to re-connect. Thus, no intervention from the end-users is required and the tree will be re-constructed automatically.

This solution introduces a problem when a significant number of clients/reflectors are under a failed node as the re-connect procedure is initiated synchronously on all dependant nodes. Imagine a tree with 25 nodes where a reflector with 20 children dies. Those 20 children were probably distributed under a number of sub-trees, possibly forming an optimized structure. As all of them were removed from the tree when the root failed they will try re-connecting to the remaining 5 nodes forming an un-optimized structure.

To fight this, the server sends RECONNECT messages gradually, starting by the immediate children of the failed node and allowing a small time period for new connections to take place. Then the children one

Proceedings of the 29th EUROMICRO Conference "New Waves in System Architecture" (EUROMICRO'03)
1089-6503/03 $17.00 © 2003 IEEE

level deeper in the failed sub-tree are asked to re-connect and this repeats down to the lowest level. This way, the disconnected nodes would probably form a similar structure to the one before the failure. Care must be taken regarding the time breaks after the RECONNECT messages as the lowest nodes do not experience great delay.

The solution implemented, incorporates all three methods described as to accommodate the capabilities of the different underlying video distribution systems.

6. Trials and tests performed

The implementation of the architecture proposed in this paper has resulted in the provision of a system that can be tested as a stand alone platform for end to end streaming media distribution, or to provide a solution that can be deployed in parallel with existing overlay implementations such as CDNs. Up to now the trials performed, have focused on the first scenario, while a full scale trial of the second scenario is scheduled in the context of the OLYMPIC project.

The tests performed include the distribution of streaming video to a limited number of clients that act as relay nodes. The distributed video is MPEG4 type distributed at a rate of 150 to 500Kbps. The scope of the trials performed is to test:

The ability to integrate different media players (both commercial and opensource). For this, three media players supporting MPEG4 video playback were used. These were: Apple Quicktime player [16] MPEG4IP player [17] and the PHILIPS Platform4 Player [18]. The results were quite satisfactory, since all players were able to reproduce the transmitted stream. However, in cases where the user is able to control reception parameters such as timers controlling the response time from the server, it was necessary to increase the related value, since the time for discovering the most suitable relay point and for setting up a connection to the related RS is added to the setup time for media distribution.

The stability of the architecture in cases of failure of relay nodes. For this, the tests included the disconnection of a relay node which server (directly or indirectly) several other nodes. This action was performed "by force" in the means that the rest of the nodes were not informed about the eminent node disconnection. The results were quite satisfactory in terms of architecture reconstruction, since after a period of a few seconds all nodes reconnected and received the transmitted stream.

The degradation in quality of the streaming media information. The tests have demonstrated that apart from the problems imposed by the network due to transmission errors, other parameters such as processing power, memory usage and number of connected users on a relay node are affecting the quality of the media stream even to greater points sometimes. For this a minimum distribution chain level in combination with measurements on the processing power and memory available on each relay node must be used together with measurements on the network characteristics before the decision of the best relay node from a client.

The OCM monitor tool, a small application built to graphically monitor and administer the overall architecture, assisted the tests. The monitor resides on top of the server OCM module of the MS. A sample screenshot is presented in Figure 6.

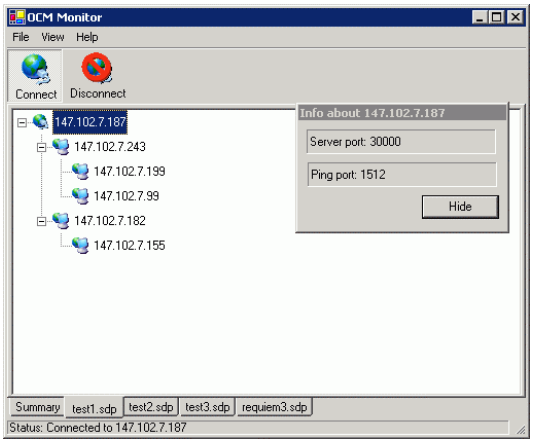

Figure 6 . OCM monitoring tool

In the instance presented, the MS is sending a media stream to five clients, based on two levels of distribution

(MS→ Level1:2RS → Level2: 3RS)

The trials performed proved that the implementation of the presented architecture can lead to a stable platform on which several commercial players may be used. The next step in the trial procedure which will be performed during the full scale trials will test the ability of the implementation to be integrated with other overlay network solutions, and the effect it will have on network performance.

7. Extensions to the architecture and future work

The presented solution has been tested as a stand alone solution in a laboratory environment not including any network part with real users (e.g. a service provider network). However, since the implementation is provided in the context of an EU IST project, the presented solution is planned to be integrated as part of larger testbed in the project trials. In this testbed, the overlay solution will be integrated with other overlay network implementations for real time media streaming, including CDN deployed over a service provider network. During the trial phase, the interoperability of the proposed architecture and its performance in real life situations will be tested.

Apart from the issue of testing the architecture in a real life scenario, there are also other issues that need to be investigated, regarding enhancements of the existing platform. Work towards these enhancements is underway, but the results are not yet available.

The first is the provision of a light version of the client, which will not incorporate the capability for

Proceedings of the 29th EUROMICRO Conference "New Waves in System Architecture" (EUROMICRO'03)
1089-6503/03 $17.00 © 2003 IEEE

media relaying. In this version, the Relay Server part will be substituted by a RTSP proxy module that will be responsible for acting as the mediator between the media player application and the media distribution point in order to exploit the capability of relay selection. This version will target clients with reduced processing power and clients with network connections with limited bandwidth (dial up users).

Another enhancement on the proposed solution is the provision of a handover mechanism between the relay nodes. This mechanism will incorporate the ability of a relay node to delegate the role of media relaying to other nodes whenever a user wishes to leave, while keeping the handover process transparent to the clients that are connected to leaving node.

Finally, an enhanced Relay Server version that can be used as a relay point for big numbers of served clients is also designed. This version will be able to incorporate different overlay network discovery and maintenance mechanisms utilising different protocols such as [7] or [10] and could be deployed in specific parts of the access network forming distribution islands that could be dynamically setup and adapt to changes in network parameters.

8. Acknowledgements

The work presented in the paper has been performed in the context of the EU IST OLYMPIC project.

9. References

[1] Ch. Z. Patrikakis, Y. Despotopoulos, A. M. Rompotis, , C. Boukouvalas, G. Pediaditis, A. L. Lambiris, "OLYMPIC: Using the Internet for real time coverage of major athletic events", *International Conference on Cross Media Service Delivery*, May 30-31, 2003-Santorini, Greece

[2] S. Deering, D. Cheriton, "Multicast Routing in Datagram Internetworks and Extended LANs", *ACM Transactions on Computer Systems,* May 1990

[3] Ayman El-Sayed, Vincent Roca, Laurent Mathy, "A Survey of Proposals for an Alternative Group Communication Service", IEEE Network, Jan 2003

[4] C. Diot, B. Levine, B. Lyles, H. Kassem, D. Balensiefen, "Deployment Issues for the IP Multicast Service and Architecture", *IEEE Network,* Jan. /Feb. 2000

[5] S. Bhattacharyya, C. Diot, L. Giuliano, R. Rockell, J. Meylor, D. Meyer, G. Shepherd, and B. Haberman, "An Overview of Source-Specific Multicast (SSM) Deployment", *Internet Engineering Task Force,* March 1999*, work in progress, Internet Draft, ietf-ssm-overview-02.txt.*

[6] Kumar S. et al. "The MASC/BGMP Architecture for Inter-Domain Multicast Routing", *SIGCOMM '98, 1998*

[7] J. Liebeherr and M. Nahas, "Application-layer Multicast with Delaunay Triangulations", *IEEE Globecom*, November 2001

[8] J. Liebeherr, T.K. Beam, "Hypercast: A protocol for Maintaining Multicast Group Members in a Logical Hypercube Topology", *Lecture Notes in Computer Science Vol. 1736,* 1999*,* pp. 72-89

[9] Dimitrios Pendarakis, Sherlia Shi, Dinesh Verma, Marcel Waldvogel, "ALMI: An Application Level Multicast Infrastructure", *in Proc. 3rd Usenix Sympodium on Internet Technologies & Systems,* March 2001

[10] Yatin Chawathe, "Scattercast: An Architecture for Internet Broadcast Distribution as an Infrastructure Service", *PhD thesis, University of California, Berkeley,* Dec. 2000.

[11] J. Park, Seok Joo Koh, Shin Gak Kang, Dae Yang Kim., "Multicast Delivery based on Unicast and Subnet Multicast", *IEEE Communications Letters, Vol. 5, No. 4*, APRIL 2001

[12] Y. Chu, S. Rao, H. Zhang, "A case for end system multicast", *Proceedings of ACM Sigmetrics*, June 2000

[13] Ch. Z. Patrikakis, Y. Despotopoulos, A. M. Rompotis, A. L. Lambiris, "PERIPHLEX: Multicast delivery using core unicast distribution with peripheral multicast reflectors", *Poster Session, Twelfth International World Wide Web Conference (WWW2003)* 20-24 May 2003, Budapest, Hungary

[14] QuickTime Streaming Server Modules, *Apple Computer, Inc.*, 1999-2002

[15] About Darwin Streaming Server, *Apple Computer*, *Inc.*, 2000

[16] http://www.apple.com

[17] http://www.mpeg4ip.net

[18] http://www.digitalnetworks.philips.com

[19] Mathew Liste, *Content Delivery Networks (CDNs) – A Reference Guide*, *Cisco* 2000

Proceedings of the 29th EUROMICRO Conference "New Waves in System Architecture" (EUROMICRO'03)
1089-6503/03 $17.00 © 2003 IEEE