Abstract—Following the actual continuous price drop on display devices with large dimensions and a relatively high range make them an attractive way to display and disseminate information. Content delivery also has an added value when it comes to advertising and content diffusion. These two concepts together turns content disclosure by digital signage in a attractive and complete concept. 

This thesis follows the content delivery by digital signage theme, creating a viable solution to develop and implement in small and medium organizations. The main requirements for this project are flexibility, adaptability, low cost and the possibility of a future functionality extension. 

The solution developed is based in the concept of content capture through remote desktop sessions. This way, it is possible to unify them, and allow their diffusion to different kind of platforms in the same way. To complement this idea, a grouping of target audience is done, in order to display the right type of content to an audience that is more interested and available to retain that information. The last concept integrated in this solution is the concatenation of distinct types of contents in a single video stream, in order to allow different displaying modes to the audience. 

Index Terms—signage, content, remote desktop, deliverysignage, content, remote desktop, deliverys

I. INTRODUCTION

Commonly used presentation displays, with great dimensions and a good displaying range, became a good and attractive way to deliver several types of information and contents, when the price of hardware dropped.

There are some solutions in today’s reality, with professional level, that are usually paid. However, freeware or open source solutions are still limited in some aspects, so a free, open source solution, able to integrate some different types of functionalities, that allow to simplify a possible implementation with low cost, does not exist. Also, considering small dimension spaces, like stores or some institutions. this kind of system is clearly a waste of equipment if it reaches only a small amount of target public, since it wastes equipment that has capabilities to do more than the required ones, and paying the cost of professional software is not justifiable. However, considering a bigger size institution or organization, there is a relatively high need to reach a larger amount of people, and more resources and announcing space, which leads them to obtain the best software available. So, in this context, there is the need for a system able to adapt to existing infrastructure easily, able to deliver different types of content, no matter if it is dynamic or static, delayed or almost real-time, or even if content is shown in displaying points with different capabilities.

A. Existing problem and solution proposal

In order to solve the main problems and compensate the lack of an existing system with low costs, the main goal of this thesis is to create a new solution that aggregates some of the existing functionalities. 

The goal is to develop a solution able to be applied in small and medium organizational environments with needs at content digital signage level. Considering this scenario, it is defined as a priority that the solution developed must be adapted to the existing infrastructures and technologies. Another requirement is a simple user interface, in other words, the users shouldn’t need specific and specialized formation in order to interact with the new software. Finally, it is this solution must be totally free, to avoid paying software licences and maximizing the usage of the existing hardware. 

With this context in mind, it is defined, as a request, the uniformity of contents in just one type. This allow content reception to be totally transparent to the user. In order to allow this, desktop capture is used when playing content, turning all of it into a single video stream, completely independent from the content type. 

Another of the main specifications of this solution is delivering different types of contents to different groups of people, each one aggregated by common properties. This way, information delivery to someone is more efficient, considering that one specific content is delivered to his target audience. Another advantage is the network resource saving, by avoiding the delivery of unnecessary content to someone that will not be receptive to it. The main types of content expected to be delivered is media content, like videos or images, but other less common types, like presentations, can be shown, or even combinations of all of them. 

The ability to adapt to small or medium organization environments is also very important in this solution context. To allow this, and also avoid unnecessary costs, the main technologies to develop this solution are free, open source, software and hardware independent. One last, and interesting, requirement is the ability to add more functionalities to the solution later, to fit each organization’s needs. 

In terms of hardware, the goal is to be able to apply the solution on common computers, despite the components or Operating System. This solution allows a wide variety of presentation devices to be used, from common personal computers playing a media client, able to receive the video, to new generation smarTVs. 

In summary, the solution is presented as a system that captures content from a remote desktop session, ensuring a good level of flexibility. The contents received are modified and then presented to audiences as a standard video stream.
This way, it is possible to save some network resources during the transmission. Being developed with free tools and technologies allows this solution to be free, an important factor in them small and medium organizations context.

B. Solution/thesis contributions

The work developed during this thesis intends to provide an implementation of a content delivery system to be applied to digital signage, in small or medium organization environments, such as schools or universities. The main idea is to deliver information on a transparent and unified way, to different audiences with characteristic properties, in order to have an effective way to provide people with contents. In a high level point of view, and taking the environment where the solution will be deployed into account, another functionality - besides the main one - is developed. This functionality allows to record, in a permanent way, to a video file, a desktop session. This can be used, for example, to record tutorials or to save content sequences, so that they can be easily moved to other servers or used later. Other two same-based functionalities will also be available: mixing and concatenation of different content from different sources. With these tools, it will be possible to take new types of information to the audience. These functionalities will increase the content that can be created and made available, like videowall presentations, or gathering different contents and displaying them at the same time, to complement each other. The last functionality added consists in a “content plus webcam” system, which can be used, for instance, to create better tutorials or even help with content explanation, such as an expert providing more visual information during a presentation.

In short, this group of functionalities can be used not only at the enterprise level, but also in an academic environment, allowing some activities related with e-learning or content providing to students or staff.

II. RELATED WORK

In this section we will discuss some of the related works in which this system is inspired and based.

A. Content Delivery Networks

Content Delivery Network (CDN) is an intermediate layer of infrastructure that aims to efficiently deliver the ever increasing multimedia content from content providers to a large community of geographically distributed clients [5].

CDNs are considered an important mean to deliver digital media or information goods[1]. There are several kinds of CDN implementations, applied according to each of the environment needs. A CDN is characterized according to its infrastructure, the number of clients, the number of servers and how this elements interact with each other.

There are some solutions to this kind of infrastructures, free or paid, and with diverse features.

B. Digital Signage

The concept of digital signage represents the evolution of the traditional content presentation systems, that is based mainly in the concept of information digitalization. The most relevant digital signage systems allow the presentation of different kinds of contents, like video, text, image, audio and animations.

Despite these possibilities, the main problem with this kind of systems relies on the fact that the presented content must be static and can’t be changed “on the fly”, which reveals an inability to adapt content to new context factors, like timing or audience. In order to adapt these systems to new realities, a new application with the different information must be set up, before it can be delivered.

To solve this problem, more recent versions of digital signage systems are interactive, making it easier to change content.

As a main objective, these systems intend to send a message to the public, independently of the architecture or functionalities. So, the systems are highly used to send advertising. In order to be efficient, by reaching the target audience, these systems are conditioned by factors like:

- Spatial identification
- Audience properties
- Content diffusing cost
- Possible conflicts with existing systems in the same area
- Limitations in the type of contents to be delivered

There are several possible architectures for digital signage systems, from the simple computer plus display used on small shops or organizations, to the complex architectures with several levels and different kind of functional options.

C. VNC

The Virtual Network Computing (VNC) system is based on a display protocol, the Remote Framebuffer (RFB) [6][7], and with independence from the platform where is being used[3].

The main objective of its development was the possibility of making mobile computing without hardware requirements[9].

The RFB protocol, where VNC is based, works over any reliable transport protocol, like TCP. The working model is based on two endpoints, one that is the user input responsible - called VNC client or viewer - and the other where are generated the framebuffer changes, named VNC server.

The protocol base[9] consists in a simple graphical primitive, in which the client receives an information from the server with the following content: "place the pixel data rectangle on the position x,y". This representation form allows several types of data codification, but imply a high level of bandwidth inefficiency and lack of processing speed or contents presentation. Figure 1 represents the most primitive functional model of RFB protocol.

Since they are RFB based systems, and most of the VNC distributions are based on GPL, it is a well known standard with a great variety of market available platforms. As an extensible protocol, functionality development is continuous on new VNC generations. The protocol requires that all
versions support edition 3.3, which currently provides all the base functionalities of a VNC system[8].

In summary, the main advantages of the VNC protocol are its flexibility, that comes from multi-platform nature, and its independence from the Operating System, or hardware, allowing applications on a client and a server with different properties[4].

III. PROPOSED ARCHITECTURE

This chapter is dedicated to analysing the system required properties and the solution’s architecture is proposed.

A. General System Description

Nowadays, the need of bring different kinds of content in an unified and transparent way to a group of clients is growing, and became extremely important in the choice of a digital signage system. According to these requirements, the content delivery network assumes an important role in the way the system is designed and implemented.

The proposed solution consists in a content delivery system, named \textit{d-sign CDN}. This system is designed with the ability of adapting to the context where it is implemented, fulfilling its specific requirements.

The main objective is to deliver multimedia content from server to client. It is necessary to unify all contents and the way how they are sent to the client. In other words, the system must convert varied multimedia content to a standard video stream. This conversion allows the data stream to be standardized, so it will always be received in the same way by the client.

To turn this concept into a practical solution, the system architecture’s base is a set of applications able to guarantee that functionality. The way how it is done consists on the usage of VLC desktop capture systems, which allows the creation of a video stream of that desktop while it is playing contents.

Other requirements include the capability to be deployed in small or medium size organizations and the reduced cost. In order to guarantee these requirements’ fulfilment, the technologies used to implement \textit{d-sign CDN} are cost free and, must be available in the most diverse kind of Operating Systems possible, to avoid the need of a specific software or hardware.

Considering the content management, one of the requirements is the adaptation of content to the target audience. To perform this, this system divides the receivers in groups, as represented in figure 2, and diffuses contents related to the groups. This will result in better reception acceptance by the groups. To implement this, multicast groups are used.

One other important property for the developing process of the system is extensibility. Using an application with module creation available, which allows some modifications and adaptations, it is guaranteed that the system can be expanded.

On the other hand, with this solutions some properties like security, redundancy or availability are not important requirements due to the kind of system developed and the non critical data used.

In summary, \textit{d-sign CDN} is designed to fulfil the requirements defined as important and is based on two levels, network level and application level.

B. Network Architecture

The solution for the system’s network architecture is designed on a theoretical level. This system is based on a client-server architecture, where the server is the main entity, with the responsibility to process all the contents, making the client a passive element on this structure. This design allows a bigger control over the contents. Controlling the contents can limit its delivery, ensuring that a content that can not, or is not meant to, be displayed to a certain group of receivers will not be diffused to them.

In terms of centralization, this is based on a centralized client-server architecture. This choice was made taking into consideration that there is a need to group all the servers in a block, allowing a major control level on the servers, as much as a good control level over their connection between each others. With this option to group servers, if one of them fails, the others ensure the system’s functionality, only adding a limitation in certain areas. This architecture design is applicable to both virtual and real servers.

The organization of the network is also based on a hierarchical server architecture, as shown in 3. This option allows different levels of control over network operation and over the contents. At logical level, the network is based on a group of private sub-networks, one for the servers group and another
associating each area server to a group of clients. To make this association, the content delivery is made using multicast groups, associated to each one of the area server and clients sub-network.

According to figure 3, there are three main components in this architecture: high level servers, area servers and clients.

The high level server is the main control unit in the system. It is the administrator and manager of servers sub-network, containing itself and the area servers. Considering his functionalities, it allows the creation, upload and edition of contents, allowing their association with themes, being each theme associated with a different multicast group.

Area servers work as the storage of contents that will be shown to audiences. Their distribution is made at area level, in other words, each area server is responsible for a target area. Inside this elements are the virtual VNC servers, where the capture of contents is made in each desktop. This capture is made by a streamer application. Each of this servers will allow the contents to be captured in one or more VNC servers, in order to supply the streamer application. Then, the contents are delivered by this application to the clients using multicast.

Finally, the clients are the meant to display the contents to the audience. The only requirements for this element is to be multicast compatible and able to open a RTP stream codified in MPEG-2 or MPEG-4. Clients can range from common workstation running Video Lan Client(VLC) to new generation smartTVs.

C. Applications Architecture

The application level is the implementation part of the system. So, considering this, it was developed using four distinct applications with the main goal of capture and delivery of contents.

The four applications have some design properties in common, being based basically in reception, conversion and diffusion of video streams. Considering the block architecture represented by 4, the first step is the capture of diverse types of multimedia contents.

First the applications connect to the VNC server, getting the VNC video stream corresponding to that server’s captured desktop, which must be a content played there. Then, some operations like colorspace change, video encoding and video multiplexing, are executed, and this varies from application to application. Finally, the stream is sent to the network in MPEG format encapsulated in a RTP video stream. This content diffusion is based on multicast groups.

1) Streamer application: This particular application is the main implementation focus in this work. The reason is that this application is a root in system execution and functionality. If this application fails, no content will be delivered.

With this application, the contents are only displayed to clients connected in a multicast group. In a generic way, this application is based on a block architecture shown in figure 5, each one of them with a different function, and pieced together create the application’s total functionality.

This application is based on a three-part structure where the parts are:

- **First block** is a remote desktop VNC server, that captures every type of contents in the form of a video stream.
- **Second block** consists in an application composed by four parts: a VNC client that connects to the VNC server, a colorspace filter to set the color format to MPEG, an encoder to codify video stream in a usual format, like MPEG-2 or MPEG-4, an application that encapsulates information in an RTP video stream and finally a multicast sink to make it available to client multicast access.
- **Third block** consists in the client that opens the video RTP stream and displays it.

This group of blocks allows a VNC capture to be turned into a video stream multimedia content, diffusible into multicast, that is the main goal of this application.

2) Other applications: The other applications are similar to the streamer one, but each one of them with some different properties.

First of all, the VNC session recording application, consists in an application that is able to connect to a VNC server and record that running session to a video file. Its structure is based on a source that connects to VNC server through a VNC client
element. Then the colorspace is changed to MPEG colorspace. In order to record into the file, the video stream is codified in MPEG-4 format and multiplexed to mp4 format. Last but not least, the file is sent to a file sink.

The other two applications consist in a streamer with webcam application, and a video wall application. These two are based in a multi source block structure, where the sources can be VNC servers, webcams or other compatible sources. Each of the sources is converted into an MPEG colorspace, and a video filter is applied to them in order to obtain the right resolution and color format. Then, the different streams are mixed with a videomixer element that gathers all of the streams in just one output video stream. In the end, the process to diffuse is the same as the one used on the streamer application.

All these applications intend to give some extra value to the digital signage system, such as e-learning, concatenation of variable content to displayed at the same time and to record tutorials or important desktop sessions, for later usage.

IV. IMPLEMENTATION

The implementation is done on open source and free technologies, in order to build an open source system. The technologies used are the Java programming language [2] and the Gstreamer framework. These two choices are based in the good deployment of each technology, each of them used largely and with great support. Based on them, there is an API used to program the applications, namely gstreamer-java, which allows Gstreamer based operations inside Java language.

To specify the implementation only the main application is being described, the streamer application. All other applications are derived from it, and the changes needed were done according to the specifications made in the Architecture chapter.

A. Streamer Application

Amongst all the applications needed, the streamer implementation has to be the most generic one. By analysing the requirements of the streamer, the final goal was to have an application able to receive a video stream from a VNC server, process it in an efficient way, and make it available to multicast groups over a usual IP network.

First of all, to establish the communication with the VNC server, which is the data source for application, a plug-in contained in the Gstreamer framework is used. This plug-in emulates a VNC client, connecting to VNC server and making the video stream available. The connection is made using the RFB protocol.

After, there is the need for a colorspace unification, due to MPEG encoding being used next. This is achieved using a colorspace filter. Then, with the stream available in MPEG colorspace, it can be encoded with an MPEG codec, in this case MPEG2, in order to be able to stream videos with a high quantity of movement.

With the stream encoded, the next step is to use an MPEG multiplexer plugin from Gstreamer, used for real time video. Afterwards, this multiplexed steam is encapsulated into a RTP payloader, in order to stream that video through the network in an appropriate format.

To finish the implementation, the stream is sent to the network through an UDP multicast-able sink, allowing the created stream’s diffusion. The stream is available for clients in the correct multicast group by accessing the address: rtp://224.0.0.0:port.

All of these plug-ins and elements are gathered in a pipeline, that sets execution line and the way that they all work together.

V. CONCLUSIONS AND FUTURE WORK

The main goal of this work was to create a solution able to deliver multimedia contents in several ways in digital signage systems. This system, d-sign CDN, was designed in order to be used in a specific context, namely small and medium dimension organizations.

An architecture of the solution and operation mode was designed, according to the base requirements. This resulted in an implementation of said architecture using a specific scenario of deployment - an academic campus. This provided functionality to the d-sign CDN architecture.

The work developed is a practical implementation of the concept of digital signage, applied in a permanent and structured way, inside a context where typically only temporary and limited solutions are deployed. Considering the application implemented, there has been an effort to extend and unify the ways to provide multimedia content to people. The additional functionalities obtained with the applications intend to make this kind of systems, and particularly d-sign CDN, more attractive and usable by the target type of organizations.

In the future, more functionalities can be added to the d-sign CDN system, as well as improving some of the existing ones. More properties not yet considered can be added too, such as security, redundancy and fault tolerance. Also, content management should be considered, through the implementation of a content manager application. Finally, more work can be done to improve the video quality.

REFERENCES


