A New Approach for Slideshow Presentation at Working Meetings

Bernardo Gaspar[1], Carlos Pinho[2], Rui Campos[2], Ricardo Morla[2]
FEUP[1], INESC Porto[2], Rua Dr. Roberto Frias, 378, 4200-465 Porto, Portugal
Phone: +351-222094000, Fax: +351-222094250, e-mail: ee99092@fe.up.pt[1], {cpinho,rcampos,rmorla}@inescporto.pt[2]

Abstract — Today, presentations at most working meetings are supported by computer slides. The traditional approach is to display the slides with a projector. A less traditional approach follows from research into interactive rooms, where presentations can be carried out with the help of multiple, typically embedded displays. This paper explores a third approach that is motivated by a simple observation – nowadays people often bring into meetings their own laptops, which are capable of supporting slideshow applications and can internetwork with other devices. We present DiS, a proof-of-concept application that implements this approach building on existing slideshow software and on network and distributed system protocols. We perform a comparative evaluation of our application with related work, focusing on two parameters: network traffic and visual responsiveness.

I. INTRODUCTION

Presentations at most business and academic meetings today are supported by computer slides. The traditional approach is to display the slides with a projector that is typically connected through a VGA cable to the computer that stores the slides and runs the slideshow application. In meetings with more than one presentation – such as conferences or project meetings where some participants present their individual contributions – this approach has one of two disadvantages. Either the speaker cannot do last-minute changes to the presentation, if he has uploaded the presentation in advance, or there is the need for often cumbersome and time-consuming cable changing and screen reconfigurations that can break the flow of the meeting, if the speaker uses his own laptop.

A less traditional approach to computer slide presentation follows from research into interactive rooms – such as *MIT's iRoom* [1]. These rooms are typically embedded with a wide range of interaction-supporting technologies (e.g. wall mounted displays, projectors, voice interfaces, and motion detectors) that can be used to enhance traditional meetings and presentations. Despite the potential of interactive rooms for supporting livelier and smoother presentations and meetings than the traditional projector approach, interactive rooms are still not commonly available.

This paper explores a different approach to support slide presentation that neither has the disadvantages of the traditional approach nor requires the additional hardware and infrastructure of interactive rooms. The new approach is based on the observation that participants often bring their laptops into meetings – laptops that can support fully functional slideshow applications and interact with other network-connected devices. To explore this approach, we implemented a distributed application called Distributed Slideshow (DiS) that enables the slideshow presentations to be displayed locally and synchronously on the laptop of each participant. The application can issue or receive messages (e.g. to control the flow of the presentation – "next slide") and interact with existing slideshow software to locally display the current slide. Participants can thus watch the slideshow in their laptops and a projector is no longer required. DiS may be beneficial in ad-hoc meetings where a projector is not always available. This application may also improve the flow of meetings where a projector is available, as it is sufficient to have one of the laptops connected to the projector for the entire meeting. The presentations of the other participants can be displayed via the laptop that is connected to the projector. A meeting in which DiS is used has the advantage of not requiring changing projector cables or downloading participants' presentations in advance.

We present requirements and the specification of DiS in Section 2, followed by its implementation in Section 3. We review other applications that are related to distributed slideshows in Section 4 and in Section 5 we compare network traffic and visual responsiveness of our application with those of the probably most used chat and application-sharing program, MSN Messenger.

II. REQUIREMENTS AND SPECIFICATION

A distributed slideshow application such as the one proposed in this paper aims at: (1) providing support for synchronous slide presentations in the laptop of each meeting participant; and (2) enabling control of the flow of the meeting through the laptop of one person at a time (i.e. the current speaker).

Meetings supported by the application have local scope, i.e. meeting participants are assumed to be all in the same room or office. Consequently, the application needs to support the visualization of presentation slides but does not need to support the transmission of audio or video. A local network (wired or wireless) to which the participants' laptops can connect must be available; no other infrastructure (e.g. projector, internet connection, and servers) is required.

The application has the following user requirements. Each meeting should be considered as a session with one or more slide presentations and should scale with the increasing number of participants. The application must provide support for announcing different sessions with one or more slide presentations and that may be taking place at the same time (e.g. in different conference rooms). Users should be able to choose which meeting to join. Participants can assume one or more of three roles: moderator, speaker and listener. Each meeting has a single moderator that starts the session and is responsible for its management, e.g. deciding who the current speaker is. Each meeting has only one speaker at a time and can have several listeners.

The application should explore an alternative to the Virtual Network Computing (VNC) screenshot export approach [2] such as transferring a file with the presentation slides to the laptops of the listeners beforehand and issuing presentation flow commands through the network (e.g. "next slide").

Prior to implementing DiS, we specified a number of technologies that the application should use in order to meet the above requirements:

- **Microsoft PowerPoint and Microsoft Office Automation**, given that it is the most widely used slideshow application. We believe that in the future other office suites (e.g. OpenOffice) can be integrated as well.
- **IP multicast**, given its potential of effectively reducing network traffic when announcing meeting sessions, issuing session commands such as "next slide", and transferring the presentation files prior to the presentation.
- **Session Description Protocol (SDP)** [3], given that it is independent of the underlying session transport protocol and provides a standard approach to announce and describe session details (e.g. session creator, session name, and session IP address and port number) that can be useful for participants to select their session.
- **Multicast XML-based protocol**, to support session management (e.g. participants joining a session) and slide flow controlling in a session given XML's [4] extensibility, ease of programming, and numerous support libraries available.
- **Multicast File Transfer Protocol (MFTP)** [5], for automatically transferring the presentation file over the network from the speaker to listeners given that it enables on-line file transfer (cf. off-line file transfer with e.g. a USB memory stick) and its potential to save bandwidth when compared to unicast transfers (e.g. FTP).

III. IMPLEMENTATION

In order to support the requirements and specification presented in the previous section, DiS's implementation focused on three major modules: the Graphical User Interface (GUI), the network protocol interface, and the application's business logic that processes events from the previous two. Figure 1 shows a diagram of these application modules. We describe each of them in more detail in the remaining of this section.

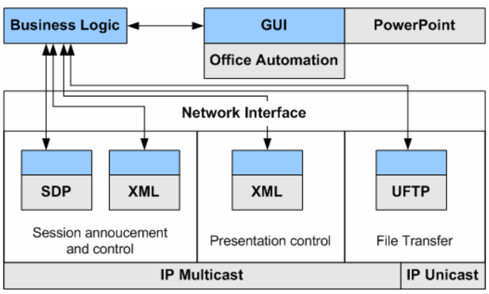

Fig. 1. Diagram of application modules.

A. Graphical Interface Module

The GUI provides support for users to create, join, and manage a session as well as for watching and controlling the flow of slides in a presentation. The Microsoft OLE Automation API is used to control the instance of Microsoft PowerPoint that is embedded in DiS.

The application provides a dialog box that can be used to create new meeting sessions. The user must supply a name for the session and for the session owner. The IP multicast address and port numbers are randomly generated by the application and in case of conflict with other sessions the user can type in a different address and port. Additionally, users can select a directory where presentation files received from the speaker are stored.

Session and owner names filled out in the create session dialog box are disseminated using IP multicast, enabling users on other laptops to discover and join existing sessions using the join session dialog box. Similarly to the create session dialog, the join session dialog allows users to select a username as well as the directory to save received files. A list of available sessions and a join session button allow users to select the session they want to join.

Once the user has created or joined a session, the application will open the window shown in Figure 2. This window allows users to: (1) manage, when acting as moderators, the session they have created by selecting the current speaker or removing a participant. This is supported by the list of meeting participants ("Session members") and by the "Speaker" and "Kick Out" buttons, respectively; (2) watch the current slide presentation – a Microsoft PowerPoint object was embedded into our application that is controlled using the Microsoft Office Automation interface and occupies most of the application window (see Figure 2); (3) transfer a slideshow presentation file to the other participants ("Transfer File" button) and control the flow of the presentation using the "Session Flow" next and previous slide buttons. "Speaker" and "Kick Out" buttons are only active for the user that created the session, while "Session Flow" and "Transfer File" buttons are only active for the current speaker.

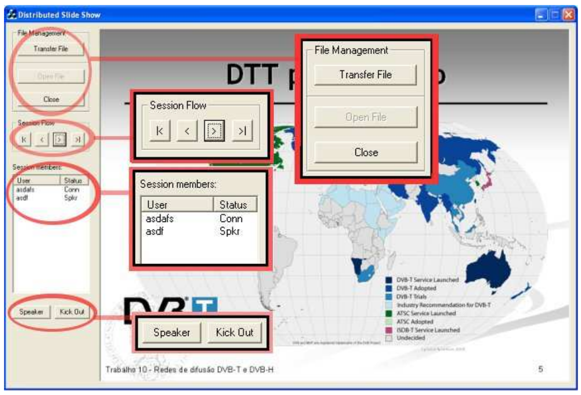

Fig. 2. Main application window.

B. Network Interface Module

The network interface module is a wrapper for the different network technologies that the application uses. The major network component is the asynchronous multicast socket module. This module allows the application to announce or join session multicast groups. The announce group has a fixed IP address (229.123.123.1) and port number (10000). Session announce messages are periodically sent by the application if it is being used to create a session. A session multicast group is created for each session with randomly generated IP multicast address and port number in the range 229.x.x.x and 10000 to 50000, respectively. Multicast traffic has link-local scope given that access points usually block unreferenced multicast traffic. As such, this simple random-based IP and port number collision avoidance mechanism is likely to be sufficient for DiS.

The other network component of the Network Interface Module is the multicast file transfer module, which is based on an existing multicast FTP implementation called UFTP [6]. UFTP has three phases: the announce and registration phase, in which the file server announces the availability of a slideshow file to download and clients register their interest in receiving that file; the transfer phase, in which the slideshow file is split and sent over the multicast channel in different UDP packets; and the confirmation phase, in which clients confirm that they have received all the packets of the slideshow file. The UFTP announce address used in this application is the same as the session multicast address, whereas the port number is one plus the session port number.

C. Application Business Logic Module

The business logic module connects the GUI module with the network module by processing the messages that arrive from the network and updating the GUI and slideshow application at the same time that it processes the GUI events and issues network commands such as “next slide”.

Session announce messages are sent in SDP format. The business logic module uses existing SDP libraries to generate and parse SDP session announce messages received by the network module on address 229.123.123.1 and port 10000.

The other network messages are formatted in XML. The XMLite tool was used to help generating and parsing these messages. Ten messages were defined in XML, including an application keep-alive message, four slide flow messages (“next slide”, “previous slide”, “jump to beginning”, “jump to end”), “speaker change” and “acknowledge” messages, a “kick user”, a “repeated user name”, and a “file transfer about to start” messages.

IV. RELATED WORK

In this section we focus on related work mostly based on VNC that can support distributed meetings addressing either remote or local collaboration scenarios.

VNC [2] is a desktop sharing system that can remotely control graphical user interfaces based on one graphic primitive: “Put a rectangle of pixel data at a given x, y position”. The VNC server runs on a machine that shares its screen and sends small rectangles of the frame buffer to the VNC client, which can be used to watch the remote desktop and interact with the server. In its simplest form VNC uses a considerable amount of bandwidth; multiple encoding methods have been developed to reduce it. VNC is platform-independent and supports multiple clients connecting to the VNC server at the same time.

MSN Messenger [7] is probably the most popular Instant Messaging program in the world. MSN gradually started including other features such as audio and video support, file sharing, and application sharing. MSN’s application sharing feature is based on VNC so that virtually any application can be shared between participants. Nevertheless, only two participants are currently supported which represents a major disadvantage for working meetings.

CentraLive for eMeetings [8] is a commercial application that enables application sharing between multiple users connected to a central server. Since it is based on VNC, it allows any application to be shared. Its main objective is to support remote collaboration between participants; therefore it also supports audio and instant messaging communications. However, the latter feature is not so relevant when considering face-to-face meetings.

Multicast PowerPoint (MPPT) [9] has been developed by Microsoft Research and considers two approaches for sharing presentations based on IP multicast: (1) presentation slides are transferred one-by-one as the presentation goes on, along with control information – the next slide is transferred in advance trying to avoid potential transition delays; (2) presentation slides are transferred in advance – during the presentation only control information is transmitted. The second approach is similar to the one considered in DiS. However, MPPT focuses on the presentation and lacks a mechanism to deal with session creation, session advertisement or session join. Thereby, the application does not work out of the box, which is likely to mean there will be an additional difficulty for the typical user to learn how to use the application.

Windows Meeting Space [10] is a new application to be released with Windows Vista that targets both local and remote collaboration environments. It includes features similar to those considered by DiS (e.g. session creation and presentation sharing) and supports other features, such as file sharing and joint file edition. Yet, it uses session invitations instead of session advertisements considered in our application. In spite of providing better security, this approach may be burdensome from the session creator viewpoint, since he will have to explicitly invite every participant in a meeting; we argue that security may be achieved by other means, such as by using public-private keys for user’s authentication. In addition, Windows Meeting Space uses the same desktop sharing mechanism as MSN to share applications between up to 10 participants. DiS has a more limited application domain (i.e. distributed slide presentation) but neither limits the number of participants nor uses the desktop sharing mechanism to share presentations.

V. EVALUATION

This section evaluates the visual responsiveness and the network traffic generated by DiS when compared with MSN Messenger Application Sharing (termed MSN in the rest of the paper). The selection of MSN was made taking into account the limited number of free application sharing software available and considering its huge utilization worldwide. In addition, since MSN uses VNC as the enabling mechanism for application sharing, its evaluation may allow us to draw conclusions about other applications also based on VNC.

The test scenario used for the evaluation considered two applications running on distinct laptops connected to a local IEEE 802.11b wireless network, which in turn was connected to the Internet (as required by MSN). One of the MSN applications acted as “server” in order to share a local presentation and the other acted as “viewer” displaying locally the remote presentation. A third laptop running Ethereal [11] – a popular network protocol analyzer – was used to capture the generated network traffic and ease its measurement. Since MSN only supports application sharing between two users, the visual responsiveness and generated network traffic comparison involving several users is not presented below.

The slideshow used for the test was composed of 14 slides (sequences of 2 images slides followed by 2 text slides). We have simulated a typical slideshow presentation by considering that slide transitions were made in intervals of approximately 5 seconds.

We focused our evaluation on the generated network traffic, namely the real-time traffic, and visual responsiveness of slide transition; the former directly influences the latter. In particular, we measured the interval between the instant a slide is displayed in the speaker's computer and the instant in which it is completely displayed in the remote computer. In order to measure the visual responsiveness of the application, the slide transitions on both speaker's and listener's laptops were captured with a webcam and further analyzed using the Windows Movie Maker application. We were able to observe that MSN has worse visual responsiveness than DiS when considering the number of frames counted between the frame in which the slide is fully displayed in the speaker's laptop and the frame in which the slide is fully displayed in the listener's laptop. In our test, the MSN slide transition took up to 7 frames with an average of 4.8 frames while DiS only took up to 1 frame with an average of 0.3 frames. Since the MSN application uses VNC-based desktop sharing system, it will progressively display on the listener's laptop the parts of the image it receives, as illustrated in Figure 3.

Figure 3. MSN slide display delay.

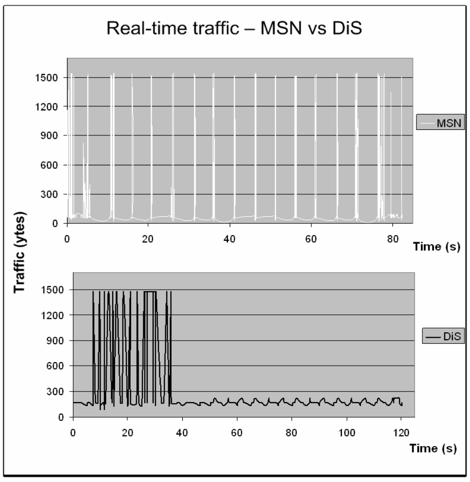

Figure 4. Real-time traffic – DiS versus MSN

Figure 4 presents the comparison of real-time network traffic generated by DiS and MSN for the same slideshow. We observed that most of the network traffic generated by DiS occurs during the slideshow file transfer before the beginning of the presentation (0-40 sec.). After that period, the generated network traffic is residual since it is mainly composed of small signaling control packets. On the other hand, MSN generates bursts of packets on every slide transition. This can be accounted for by MSN's VNC-based functioning principle. Overall, the larger the number of slide transitions the worst is the network traffic performance of MSN compared to DiS. Before the end of the presentation, the total amount of traffic generated by MSN (800 kBytes) had surpassed that of DiS (600 Kbytes, most of which before the beginning of the slide transitions). We believe similar visual and network results are achieved when more listeners are involved, given the use of IP multicast and multicast file transfer.

We used our application in two live presentations at INESC Porto. There were two speakers and three listeners with laptops. Listeners were able to see the presentation both in the laptops and in a projector connected to one of the laptops. The application behaved according to its specification.

VI. Conclusion

This paper presented an innovative approach for supporting slideshow presentations at working meetings. The approach is implemented by means of a new software application – DiS, aimed at running in each participant's laptop and is based on the fact that in today's working meetings each participant is likely to have his own laptop. DiS outperforms the state of the art standard application in terms of generated network traffic and visual responsiveness. Our application may be extended to include, for instance, support for other presentation file formats (e.g. pdf and html) and mouse position sharing; allowing speakers to point out important information appearing in slideshow presentations. To be best of our knowledge, our application is the first, from those targeting local scope collaboration within working meetings, that exhibits visual responsiveness comparable to that achieved with the traditional projector approach and that supports session creation, advertisement, joining, and management in the same software infrastructure.

References

[1] M. Coen, "The future of human-computer interaction, or how I learned to stop worrying and love my intelligent room.", IEEE Intelligent Systems, 14(2):8–10, March 1999.

[2] T. Richardson et al. "Virtual Network Computing", IEEE Internet Computing, Feb. 1998 pp. 33-38.

[3] M. Handley et al., "SDP: Session Description Protocol", IETF RFC 4566, July 2006.

[4] Tim Bray (Ed.) et al., "Extensible Markup Language (XML) 1.0 (4th Edition)", W3C Rec., Sep. 2006.

[5] k. Obraczka, "Multicast Transport Protocols: A Survey and Taxonomy", IEEE Communications Magazine, Jan. 1998, pp. 94-102.

[6] UFTP, http://www.tcnj.edu/~bush/uftp.html.

[7] MSN Messenger, http://messenger.msn.com.

[8] CentraLive for eMeetings, http://www.centra.com.

[9] J. Gemmell et al., "A Scalable Multicast Architecture for One-to-many Telepresentations", in IEEE Int. Conf. Multimedia Computing and Systems, Austin, USA, June 1998, pp. 128-139.

[10] Windows Meeting Space, http://www.microsoft.com/windowsvista/features/forbiz/sharing.mspx.

[11] Ethereal, http://www.ethereal.com.