Bringing Thin Clients to the Mobile World

Vankeirsbilck Bert[1], Simoens Pieter[1], Deboosere Lien[1], Dhoedt Bart[1], Westphal Fritz-Joachim[2], Plantier Thomas[3], Lécroart Benoit[4], Prêteux Françoise[5], Dejonghe Antoine[6], Point Jean-Charles[7].

[1]INTEC – IBCN, Ghent University - IBBT, Belgium; [2]T-Systems Enterprise Services, Berlin, Germany; [3]Prologue, Paris, France; [4]NEC Technologies (UK), United Kingdom ; [5]Institut Telecom, France; [6]IMEC, Leuven, Belgium; [7]JCP-Consult, France

{bert.vankeirsbilck, pieter.simoens, lien.deboosere, bart.dhoedt}@intec.ugent.be

Abstract: **The thin client computing paradigm shifts computation and storage into the network, in order to relieve the client device. This approach is well suited for mobile battery-powered devices, which are resource constrained. Current thin client technologies are performing adequately for classical applications (such as text processing, spreadsheets etc.) in a stable, high bandwidth network environment. However, these solutions do not cope well with the characteristics of wireless networks and are not optimized for client energy efficiency. This paper presents the vision of the MobiThin project that will enable the execution of applications, also beyond the scope of office applications, anywhere, using any device on any network.**

Keywords: Thin Client Computing, Mobility, Wireless Networks, Quality of Experience.

1 INTRODUCTION

The thin client computing concept basically comes down to moving the users' applications to a distant server and running a thin client protocol between user and server. The client device only deals with user interaction and the protocol forwards user input (e.g. keystrokes) to the server and delivers graphical updates back to the client for presentation (on screen).

The advantages of thin client solutions concerning total cost of ownership reductions, including simplified maintenance, data security and privacy, ubiquitous data and service access and more efficient use of resources have made them very successful in wired LAN environments.

The concept is promising for usage in wireless environments, if certain modifications are developed. It can bring a solution for offering demanding applications to a resource constrained device with limited battery autonomy. The needed solution to accomplish this differs from the current available technology in and that the user base is large, and geographically wide-spread, which mandates an advanced management, and that a wireless, low bandwidth network is involved.

The characteristics of wireless access networks and mobile devices cause problems in ensuring Quality of Experience (QoE) with current thin client technology in wireless wide area networks. The QoE requirement that makes thin client computing over wireless networks so

challenging is the maximum round trip delay: 150 ms for office-type applications [1,2], for games the upper bound is even lower, 80 ms [3,4,5]. For example, this is the time it takes from the moment the user presses a key until the graphical output is shown on his screen.

A thin client protocol operates between the client device and the server where the application is executed. However, simultaneously serving a large number of mobile users requires a number of distributed servers to meet the strict interactivity requirements defined before. An intelligent management framework is needed that is able to intervene at a high level.

One of these high level interventions is to select a well-chosen server to decrease effects of delay. When the end-to-end delay increases due to the mobility of the user, the varying network characteristics or the load of the server, the management framework orders a user session migration. This migration action seamlessly relocates the computing environment that serves the user to a better-suited server at runtime.

Some important aspects of QoE, such as client device battery lifetime and fluent graphical presentation are not dependent on end-to-end delay exclusively and thus can not be controlled solely through the intelligence of the management framework. Enhancements can be made on the thin client protocol level: the protocol can adapt to the environment it operates in, such as client device battery status, available network bandwidth, the graphical resolution of the client device or the kind of graphical content to be shown on the device's screen. Classic thin client protocols such as Microsoft Remote Desktop Protocol (RDP), Virtual Network Computing (VNC), Citrix Independent Computing Architecture (ICA) or FreeNX [6,7,8,9] do not cope well with dynamic content, even over high bandwidth network links [10]. Our proposal is to use an adaptive thin client protocol that is aware of the amount of motion in the graphical content and decides on streaming the image for high motion or a classic thin client protocol for low motion content.

MobiThin [11] is developing an end-to-end solution and addresses all important blockers of the wide adoption of the wireless thin client computing paradigm. These include architecture and technology issues (wireless medium optimization, dedicated video codec and user pattern research, software/middleware, performance and energy saving oriented solutions) as well as economic ones (business roles and models). The realization of

Corresponding author: Bert Vankeirsbilck, Intec - IBCN, Ghent University – IBBT, Gaston Crommenlaan 8 bus 201, 9050 Ghent, Belgium, +32 9 331 49 36, bert.vankeirsbilck@intec.ugent.be

MobiThin will enable running applications anywhere, to use any device on any network.

This paper presents the vision of the MobiThin project and is structured as follows. In section 2 the overall architecture is presented. It shows the infrastructure needed to support a large number of mobile thin client users. Section 3 discusses the possibilities to support user sessions, and presents session migration as a tool for lowering network delay by influencing the distance between client and server. Section 4 presents how a thin client protocol can adapt to the graphical content on the client device's screen to ensure QoE. Conclusions are drawn in section 5.

2 ARCHITECTURE

The MobiThin solution will work with the current available Internet and wireless communication network infrastructure. This infrastructure basically consists of three main parts: the access network, the aggregation network and the core network. The MobiThin project assumes the first part of the network path from client device to server to be the bottleneck link. Different network access methods will be supported as first mile, such as LTE, UMTS, or wired line.
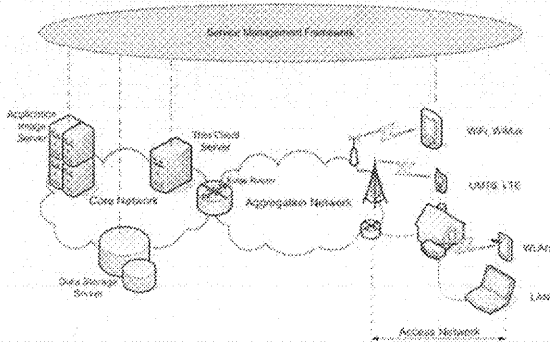

Figure 1: MobiThin architecture, supporting different network technologies as first mile in the access network. Service Management Framework has overall control. Application Image Server and Data Storage Server are stand-alone components for scalability.

Besides the network itself, Figure 1 presents the infrastructural components that constitute the MobiThin system.

Thin Client Server: the server that executes the user's applications. It receives user input and returns audio-visual output.

Application Image Server: a repository for application images. When a thin client server does not have a certain application installed locally, it gets it from an application server. This server does not run the applications but contains and distributes them to thin client servers.

Data Storage Server: a repository for private user data (not for data about a user). Since the clients are to be thin, data storage is shifted into the network. This data will be accessed during thin client sessions, or can be accessed independently, without connecting to the thin client

server. Another reason to store the data in the network is to relieve the network (in particular the bottleneck link), by reducing the need of transmitting the data to the application over and over again.

Thin Client Device: the mobile device through which users consume thin client services.

Service Management Framework: this framework has the overall control in managing network and services. When a user logs in, the framework searches an appropriate thin client server to connect to. It also manages the sessions once they have been initiated. Some functions of the service management framework are resource management, application management, load balancing, resilience, session management and business support such as billing and logging. Another aspect that is also covered by the service management framework is AAA (Authentication, Authorization and Accounting).

The application server, data storage server and thin client server are presented as separate network elements as this is likely to improve manageability. However, these functions could be merged in a practical setting. Multiple instances of these components could also exist for high availability and scalability. The latter is very important for MobiThin to meet the ambition of comfortably serving a large crowd of simultaneous mobile users with numerous applications.

The design of the MobiThin system takes into account that, depending on the business model, the thin client servers can be shifted into the aggregation network to be closer to the users, with further reduction of end-to-end delay in mind.
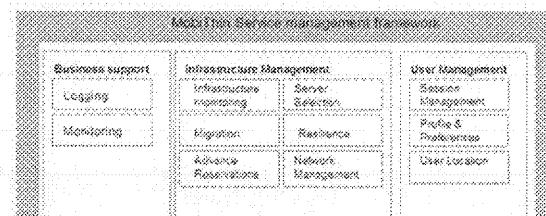

Figure 2: Service Management Framework overview, showing the tree main functions.

Figure 2 goes deeper into the structure of the service management framework. Through monitoring and logging, information is gathered to support business-related tasks such as billing and accounting.

Another block in the diagram addresses user management. This includes tracking the user location, building up profiles and storing preferences, and managing sessions on a per user basis. This information about the users' habits and profile will be useful for prediction and management of the total load of the MobiThin system, and will allow reserving resources in advance to increase the speed of service.

The last group of components in the service management framework focuses on infrastructure management. It gathers knowledge about the infrastructure configuration via monitoring. It will contain algorithms and techniques

to ensure resilience and to reconfigure the network for optimal use of the available resources. Advance reservations will be made to ensure users perceive an enjoyable quality of the delivered thin client services. These reservations concern resources in the network (e.g. bandwidth reservation) and on the server. An important part of the infrastructure management is the selection of an appropriate thin client server for the user, which is mainly dependent on the location of the user, the load of the servers and the state of the network between user and server. When problems arise during the user session in terms of delay or server load, a session can be migrated to another thin client server at runtime. This function relies on the server selection algorithms to select a new, more suitable server.

3 REMOTE APPLICATION EXECUTION

Traditionally, applications are delivered to remote users through *Server Based Computing* (SBC) technologies (Figure 3), such as Windows Terminal Services [6] and Citrix XenApp (formerly known as Presentation Server) [8]. Recently, interest in *Virtual Desktop Infrastructure* (VDI) has been growing: desktop machines are replaced by virtual environments running on servers in a datacenter which are accessed through a thin client protocol (Figure 4). Some examples are VMWare's Virtual Desktop Infrastructure [12] and Ericom's PowerTerm WebConnect [13].
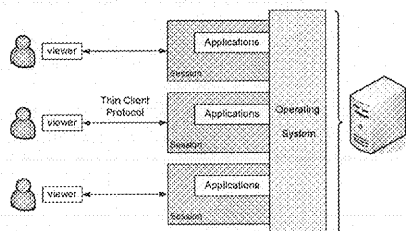

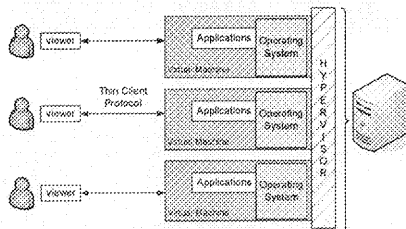
Figure 3: Server Based Computing (SBC)

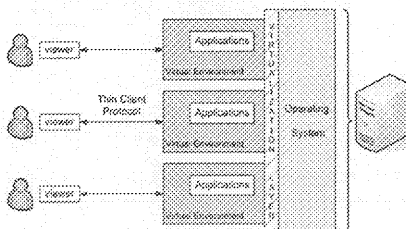Figure 4: Virtual Desktop Infrastructure (VDI)

Figure 5: OS-Level Virtualization

As stated in section 2, it is important that a user session can be migrated efficiently from one thin client server to another. Migrating an SBC session from one server to another relies on process migration technologies (e.g. Condor [14]), while migrating a VDI session relies on the migration tools of the used virtualization technology also referred to as *live migration* (e.g. VMWare's VMotion). The advantage of migrating a complete virtual environment compared to a single application (i.e. SBC session), is that all problems with process-level migration (e.g. open files, network connections, etc.) are avoided [15]. For the MobiThin project, virtualization is the most obvious choice for the user session architecture, since it is important to migrate sessions in an efficient, reliable manner. The disadvantage of VDI compared to SBC is the overhead of the virtualization in terms of resources at the thin client server. However, the overhead depends on the virtualization technology used: hypervisor-based or OS-level virtualization (Figure 4, Figure 5). With hypervisor-based solutions, each session contains its own Operating System (OS). With OS-level virtualization, the OS of the host system is shared amongst all sessions while still providing complete isolation between different Virtual Environments. The overhead of hypervisor-based solutions is larger than for OS-level virtualization due to the overhead of having a separate OS for each session [16, 17].

In [18] it is shown that virtual machines can be migrated in a WAN/MAN environment with near second downtime.

4 PROTOCOL ADAPTIVITY

Extensive communication is established between thin client server and the device. Different types of traffic are exchanged: audiovisual information, user events and signalling information. These traffic types all have different QoS requirements. Furthermore, the MobiThin thin client protocol suite must support both static and dynamic variations of the environment and the network. Static variations are, for example, the user switching to a new device with different display characteristics or connecting to a different wireless access network. Dynamic variations are induced by, for example, the inherently time-varying nature of the wireless medium or the decreasing energy level of the device battery.

From these requirements, the need for a flexible, adaptive protocol emerges. While static variations are merely configuration options that can be set through signalling, coping with the dynamic variations of the different traffic types is more complex. It requires a close and orchestrated interaction between various system components, such as the different layers of the protocol stack. Strong algorithms are required to monitor, analyze, remedy and optimize settings of the protocol stack.

4.1 MobiThin cross-layer approach

An overview of the MobiThin cross-layer approach on a wirelessly connected thin client device is presented in Figure 6. Contrary to existing thin client protocols, the MobiThin protocol will be aware of its environment

through monitoring and by taking into account user profiles and preferences.

The optimization of the image transmission mainly relates to the real-time dynamic combination of video and graphics elements with the goal of optimizing both the transmission bandwidth and the decoder complexity. Existing thin client protocols such as VNC, RDP and ICA are optimized to handle static displays (i.e. low motion content) with infrequent display updates, typically for office applications [10].

However, MobiThin targets running demanding applications on a thin client, which include the fluent presentation of video content or gaming graphics. This type of graphics is characterized by fine-grained, complex colour patterns and few inter-frame similarities [19]. The MobiThin protocol will comprise several image transmission methods, such as 2D primitive commands or video streaming. These solutions are designed so as to comply with the widest used open standard requirements and specifications, with a particular focus on MPEG-4. Due to the MobiThin novelty, potential extensions of these standards should also be taken into account. The selection of the appropriate transmission mode will depend on the application type, the available network resources and the client device characteristics.
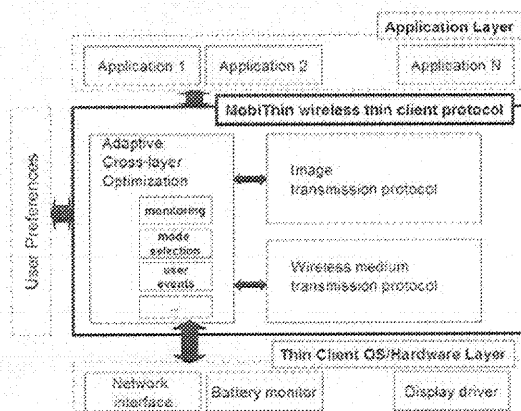

Figure 6 - MobiThin cross-layer adaptive approach to realize an efficient, wireless thin client protocol.

The cross-layer adaptation component will continuously analyze the available monitoring information on the transported traffic, on the network and on the device status. For example, when the client is running out of battery power, the cross-layer adaptation component could decide to switch to a graphical format that consumes less energy for the decoding. Similarly, when the wireless link bit error rate increases, the cross-layer adaptation component can prioritize the graphical updates.

4.2 Adaptation to content

At the current time a hybrid protocol that is adaptive to the graphical content has been implemented. High-motion graphical content is sent through video streaming, low-motion content through the VNC protocol. The architecture is shown in Figure 7.

At the thin client server, an Xvnc server was modified. The images generated by the application are analyzed. When a lot of difference between subsequent frames is detected, the images are encoded by an H.264 encoder and streamed as video to the client. The latter mode is called desktop streaming. Otherwise, the VNC default mode is activated.
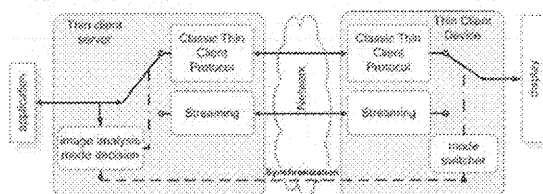

Figure 7 – Operation of the current implementation of the adaptive protocol: switching between classic thin client protocol and streaming mode based on the analysis of the graphical content.

Figure 8 shows the bandwidth for a trace comprising text editing (1s – 20s), watching a video in a small window (20s – 34s) and watching a video in full-screen (34s – 45s). For low-motion images, such as when using a text editor, VNC and desktop streaming use the same amount of bandwidth. However, decoding the video stream incurs a higher CPU load and energy consumption on the client. For high-motion images, desktop streaming is a viable approach, since the VNC bandwidth requirements are too high and in general insufficient QoE is reached. It can be seen that the hybrid approach is clearly the most bandwidth efficient when watching a video.
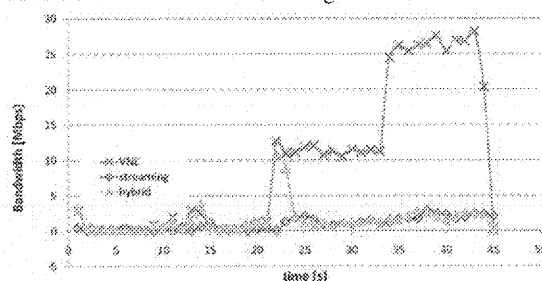

Figure 8 - Bandwidth for a trace comprising text editing (1 - 20s), watching a video in a small window (20s - 34s) and watching a video in full screen (34s - 45s). The hybrid approach is clearly the most bandwidth efficient when watching a video.

On lower layers of the protocol stack, the wireless medium transmission optimization includes two challenges: supporting Quality of Service on the inherently error-prone and time-varying wireless connection, and energy scalability, as the radio function takes a significant fraction of the overall client energy consumption. MobiThin will tackle this challenge by thoroughly assessing the different kinds of traffic to be transported (e.g. regular data, audiovisual data and upstream user events) and define appropriate QoS classes. By including the capability of performance scaling, i.e. adapting the performance to the energy conditions, it

becomes possible to realize significant energy gains in comparison with traditional designs.

Conclusions

In this paper the MobiThin vision is presented of bringing existing, possibly resource consuming applications to any mobile user with resource constrained devices and limited battery autonomy. Problems and first solutions associated with the thin client protocol, mainly due to the wireless, low bandwidth nature of the interconnection, and with the infrastructure management, caused by the large, geographically spread user base were highlighted.

Acknowledgements

References

[1]   Niraj Tolia, David G. Andersen, M. Satyanarayanan. *Quantifying Interactive User Experience on Thin Clients*. IEEE Computer, Volume 39 – 3, pages 46-52, March 2006

[2]   Niraj Tolia, David G. Andersen, M. Satyanarayanan. *The Seductive Appeal of Thin Clients*. February 2005.

[3]   Pantel, L. *On The Impact of Delay on Real-Time Multiplayer Games*. International Workshop on Network and Operating System Support for Digital Audio and Video, 2002.

[4]   Dick, M. *Analysis of Factors Affecting Players' Performance and Perception in Multiplayer Games*. Proceedings of 4th ACM SIGCOMM workshop on Network and system support for games, 2005.

[5]   A.F. Wattimena, et al. *Predicting the perceived quality of a First Person Shooter: the Quake IV-model*. 5th Workshop on Network & System Support for Games, Netgames, 2006.

[6]   Microsoft, Windows Remote Desktop Protocol (RDP) and Windows Terminal Services, http://www.microsoft.com.

[7]   Virtual Network Computing (VNC), http://www.realvnc.com.

[8]   C. Inc., Citrix Independent Computing Architecture (ICA) and Citrix XenApp. http://www.citrix.com.

[9]   FreeNX, http://www.nomachine.com.

[10]   L. Deboosere, J. D.Wachter, P. Simoens, F. D. Turck, B. Dhoedt, P. Demeester. *Thin client computing solutions in low- and high-motion scenarios*, in:"Proceedings of International Conference on Networking and Service (ICNS)", Athens, Greece, 2007.

[11]   MobiThin, http://www.mobithin.eu

[12]   VMWare Virtual Desktop Infrastructure, http://www.vmware.com/products.

[13]   Ericom PowerTerm WebConnect http://www.ericom.com/serverbased.asp.

[14]   D. Thain, T.Tannenbaum, M. Livny. *Distributed computing in practice: The condor experience*. 2004

[15]   C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, A. Warfield. *Live Migration of Virtual Machines*. Proceedings of the 2nd ACM/USENIX Symposium on Networked Systems Design and Implementation (NSDI), Pages 273-286, 2005

[16]   P. Padala, X. Zhu, Z. Wang, S. Singhal, K. G. Shin. *Performance Evaluation of Virtualization Technologies for Server Consolidation*. Enterprise Systems and Software Laboratory, HP Laboratories Palo Alto technical report, April 2007.

[17]   S. Soltesz, H. Potzl, M. E. Fiuczynski, A. Bavier, L. Peterson. *Container-based Operating System Virtualization: A Scalable, High-performance Alternative to Hypervisors*. EuroSys 07: Proceedings of the 2nd ACM SIGOPS/EuroSys European Conference on Computer Systems 2007, ACM, March 2007.

[18]   F. Travostino, P. Daspit, L. Gommans, C. Jog, C. de Laat, J. Mambretti, I. Monga, B. van Oudenaarde, S. Raghunath, P. Y. Wang. *Seamless live migration of virtual machines over the MAN/WAN*. Future Generation Computer Systems Volume 22, Pages 901-907, 2006.

[19]   D. Commander, Background on VirtualGL, http://www.virtualgl.org.
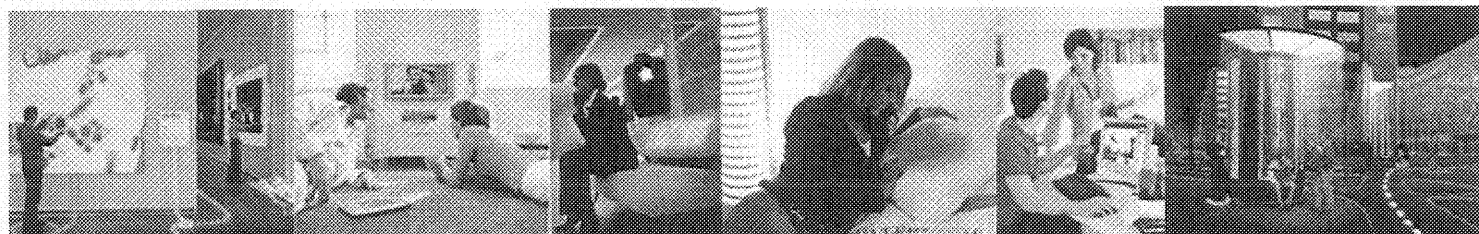
October 13-15, 2008
Saint-Malo, France
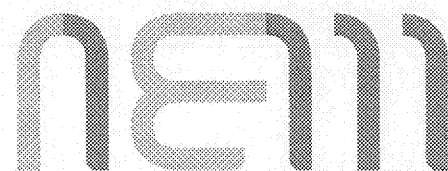
2008 NEM Summit

"*Towards Future Media Internet*"

Conference
Proceedings