A Conceptual Architecture for Adaptation in Remote Desktop Systems Driven by the User Perception of Multimedia

Marcin Lubonski, Valérie Gay, Andrew Simmonds
University of Technology, Sydney, PO Box 123, Broadway NSW 2007, Australia
{marcin, valerie, simmonds}@it.uts.edu.au

Abstract - **Current thin-client remote desktop systems were designed for data-oriented applications over low-quality LAN links and they do not provide satisfactory end-user performance for more and more popular in enterprise environment graphical and multimedia applications. To improve perception of those applications in thin-client environment we designed architecture of a server-side Quality of Service (QoS) management component responsible for mapping application QoS requirements into network QoS. We analyze how service differentiation and traffic management techniques combined with user perception monitoring can be used in order to adjust network level resource allocation when performance of multimedia applications in remote desktop environment is not meeting user requirements. Our objective is to provide QoS-aware remote desktop systems which will be able to manage available resources in intelligent manner and meet end-user performance expectations.**

I. INTRODUCTION

Remote desktop systems such as Citrix Metaframe Server [1][2], Microsoft Terminal Services [3], X-Window System [4], AT&T VNC [5] and Tarantella [6] use thin-client protocol to virtualize user desktops. The protocol encapsulates display updates of remote application, sends them from the server to the client and transmits back user input. Remote desktop protocols were designed and optimized to work for the standard data-oriented applications in a LAN environment over low-bandwidth links. Although working efficiently in this kind of environment they perform poorly in contemporary office environment where multimedia applications such as: multimedia presentations, e-learning or audio and videoconferencing started playing more significant role. Nieh et al. [7] performed a comparison of the performance of the most popular on the market thin-client remote desktop systems. His results show that remote desktop protocols cannot operate efficiently in networks having significant latency and with quickly changing network characteristics typical for the best-effort network such as the Internet.

In contemporary remote desktop systems there is clearly an existing gap between applications performance requirements and that what the underlying network can provide. This results in particularly low performance for the network-sensitive applications such as multimedia which significantly decreases total end-user's satisfaction.

Additionally, the everyday experience shows that despite many problems with providing QoS in the Internet more and more multimedia applications successfully work even over best-effort links. The applications worth mentioning include audio and video streaming (such as Helix from Real Networks or Windows Media Services) or popular VoIP services (e.g. Skype), including the voice conversation features in most of the popular instant messengers (e.g. Yahoo! or MSN messengers). All of those solutions employ internal adaptation mechanisms using specifics of human voice or video perception in order to minimize resource usage and cope with congested links.

Therefore we argue that both from the business and performance point of views, there is a need for intelligent application-level adaptation mechanisms in remote desktop systems which will able to compensate for temporary decrease in network performance or disruptions in the transmission between a client and a server.

This paper gives an overview of our work in progress and summarizes the initial research in providing QoS-aware mechanisms in remote desktop protocols based on the ITU T.125 [8] standard. In our approach we incorporated the most relevant techniques from the domains of traffic management, user perception and multimedia in order to provide truly QoS-aware systems. The results enabled us to propose the architecture of a server-side component which can perform dynamical mapping between *user-perceived QoS*[1] and *network QoS*. The introduced component assesses applications' perceptual performance based on the stored user profiles and attempts to improve user perception by intelligently allocating available network resources.

[1] We use the terms *Quality of Service (QoS)* and *user-perceived QoS* with the following meaning: 'QoS is the collective effect of service performance which determines the degree of satisfaction of a user of the service' [22], expressed in terms of network-level QoS parameters such as delay, jitter, loss, etc. and respectively user-perceived QoS (sometimes referred as *Quality of Experience*): "the level of quality for particular application which the users believe they experienced" [23] expressed for instance as Mean Opinion Score (MOS).

Such optimization of resource usage, by making the remote desktop protocols QoS-aware, introduces a powerful control over the differentiated application-level traffic. Additionally, by prioritizing the traffic we want to control performance of network-sensitive applications such as multimedia.

The rest of the paper is organized as follows: Section 2 presents related work, Section 3 gives details about the architecture of the server-side component and Section 4 concludes the paper.

II. RELATED WORK

As soon as the network-demanding applications have been introduced in the Internet the real problems arose.

There have been many attempts to improve performance of those applications in the best-effort WANs both on the networking and application level. The paper by Ashton et al. [9] categorizes the existing performance enhancements in WAN into three main groups which can be defined as:

- QoS/Traffic management
- infrastructural elements such as caches, stream multipliers, etc.
- and adaptive encoding or compression

However application of infrastructure elements such as caches or proxies are successful for such services as WWW, or even video-on-demand in remote desktop scenario they are infeasible. The main reason is the dynamic and user-specific character of the display updates sent periodically between remote server and its users.

It is also difficult to integrate remote desktop systems directly with common network-level QoS mechanisms because of the protocol specifics. For instance, the application level protocols used in the remote desktop environments are often closed and proprietary. Additionally, business related requirements might enforce application level encoding of these protocols.

However, some of the commercial solutions (e.g. Citrix) can be integrated with layer 7 performance optimizers such as Packeteer Packet Shaper [10] or Expand Network Accelerator [11] to provide traffic prioritization and bandwidth guarantee for their proprietary protocols. But their impact on multimedia performance amelioration is limited. This is because the prioritization of the traffic is done on the per connection basis according to the rules for particular type of application-level protocol. But the remote desktop protocols such as Microsoft RDP or Citrix ICA [12] which are based on the ITU T.125 specification and can contain several *virtual data channels* in a single connection. In result, network performance optimizers are unaware of the internal structure and QoS requirements of different flows within a remote desktop connection. Additionally they work only on the edge of WAN and have to be integrated with network technologies (e.g MPLS) used in WAN along the path from end-users to the server in order to provide real end-to-end QoS control.

The most promising approaches, attempting to provide a link between network QoS mechanisms and application performance, are in the area of QoS-aware applications. This research is based on the results from many domains. The best overview of the application-level adaptation techniques is given by Vandalore et al. [13]. So far most of the existing solutions focus only on a single type of application, mostly multimedia. A very interesting insight in the mechanisms used for video streaming can be found in [14]. In the area of thin-client computing [15] proposes adaptive encoding of desktop display updates in Virtual Network Computing (VNC) system based on the information about available bandwidth.

The existing solutions trigger adaptation changes based on the assessed performance of the end-user. This could be achieved by constructing a mapping between network and application domain QoS parameters. For instance, the DAVE system [24] uses user model incorporating both semantic as well as special/temporal information to change video encoding parameters. [16] uses mean square error (MSE) and peak noise-signal ratio (PSNR) to map layered video into different QoS classes and [17] introduces utility functions to assess user Mean Opinion Score (MOS). Among those approaches the last approach seems to be the most general, providing means to personalize it for a particular user, terminal device and access technology.

Among the more general approaches, not limited to the single type of traffic is adaptive packet marking in DiffServ-enabled networks. The main benefit of this approach is smart and on-the-fly network resource reallocation in a shared link for the multiple data flows with different QoS requirements [18][19].

In similar manner to our approach MPEG-4 specification recognizes the need to differentiate between different types of flows within one multimedia stream and provides a framework for QoS mapping to a suitable transport protocol stack using Delivery Multimedia Integration Framework (DMIF) layer [20][21].

The only limitation of this approach is static resource allocation on the path from source to destination of the multimedia stream in order to provide guaranteed end-to-end QoS level.

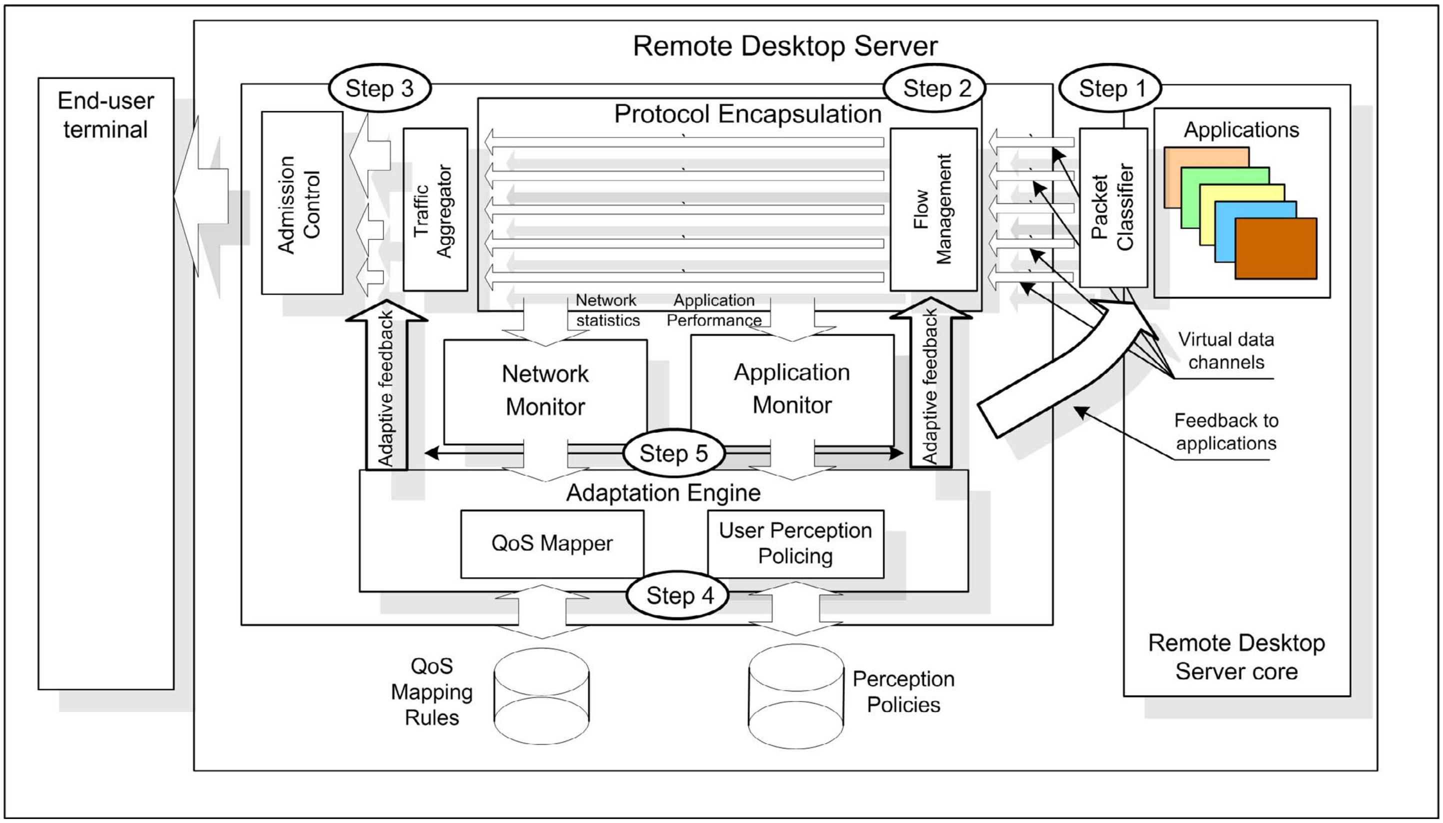

Figure 1. Remote Desktop Protocol Adaptation – conceptual architecture

Despite many solutions attempting fill the gap between network and application performance there is no single framework able to provide dynamic QoS management based on the application-level perceptual information. Our contribution to the previous research is integration of suitable approaches into a single solution in order to provide better than best-effort QoS for network-sensitive applications in remote desktop systems.

III. SERVER-SIDE COMPONENT FOR MAPPING THE USER-PERCEIVED QOS ONTO THE NETWORK RESOURCES

In order to provide true QoS-awareness to remote desktop systems we introduce a server-side component to of remote desktop server. Instead of DMIF's static network resource allocation for each data flow when it is created, we use end-user perception information to dynamically adapt to changing network conditions. Resource allocation for each application flow of distinguishing characteristics within a single user connection can be changed based on the information about current network state and end-user perceived performance. This mechanism can be used to prioritize the most demanding data flows such as multimedia streams. Apart from per-flow adaptation mechanisms our component can optimize network resource usage among multiple client connections sharing the same link.

The component was designed to ensure guaranteed services level for the most critical data flows such as control connection, print job spooling or administrative applications. Additionally we tried to design a component as transparent to the existing systems as possible in order to limit possible integration problems such as changes in existing mechanisms and protocols.

The general configuration of remote desktop systems consists of:

- Remote users in access network working with remote desktop client installed on their workstation
- A link over WAN connecting remote branch or office with the company headquarters
- Remote desktop server/farm of servers hosted in enterprise data centre

Due to server-push mode of updates, which proves to be the most efficient and is employed in most popular remote desktop systems, the proposed component is placed on the server side.

Its architecture is composed of several functional elements corresponding to the core mechanisms used as its basis. Those mechanisms include: user perception monitoring, mapping between user-perceived and network QoS, application adaptation mechanisms as well as traffic management mechanisms – traffic differentiation, traffic shaping and aggregation as well as admission control. The architectural elements can be categorized into two main groups depending on the scope of the adaptation mechanisms they implement (see Figure 1).

Per application flow adaptation, including:

- **Remote Desktop Protocol Encapsulation** (with Flow Management and Packet Classifier) – responsible for flow marking and protocol encapsulation according to the adaptation rules
- **Network Monitor** – monitoring network performance along the path from the server to the user
- **Application Monitor** – monitoring application performance and triggering application adaptation changes
- **Adaptation Engine** (with QoS Mapper and User Perception Policies) – responsible for QoS mapping

Per connection adaptation, including:

- **Traffic Aggregator** – responsible for aggregation of traffic with similar characteristics between user connections
- **Traffic Admission Control** – responsible for mitigation of mutual influence of flow aggregates with diametrically different characteristics

Additionally, the architecture includes two rules databases: **QoS Mapping Rules** storing mapping rules between network QoS parameters and application level QoS and **Perception Policies** containing user perception policies.

The QoS management mechanisms, based on the user perception, benefit from the observation that some changes in application level parameters result in significant increase or decrease of user-perceived QoS while the others remain unnoticed. For instance, an analysis of temporal and spatial parameters of video streaming applications conducted by Srinivasan and Nepal [24][25] showed significant deterioration in user-perceived performance when the number of displayed frames per second or color depth was reduced below some threshold.

This information is used to create a user perception model. It is then stored in Perception Rules database. When a user connects to the server over remote desktop requesting access to this particular application user perception model is extracted from the database. It is then used throughout the time of client connection as a reference for monitoring user perception changes.

During user connection to the server Network Monitor collects statistics about the network link to the client and compares them with appropriate service level rules for the particular type of user. This information serves as a basis to User Perception Policing to verify whether the user perception requirements are met. If there is significant and persistent violation of limits of the parameters required for particular data flow, the system triggers the adaptation procedure. The first step is to modify the parameters within user connection; for instance allocating more bandwidth to particular data flow if possible. Failing that, the admission control would try to modify parameters of the flow aggregate to which this data flow pertains in order to meet requested requirements. Finally, if that fails the feedback is provided to the application and if any adaptation mechanisms within application itself are available, they would have a chance to compensate for with the current network state. Within this procedure the adaptation component verifies influence of possible changes on other data flows and assures that none of the critical applications are disadvantaged.

If the user connection consists of more than one data flow with distinctive characteristics – upon a change in the network state – Adaptation Engine would try to adjust resource allocation in such a way that user-perceived QoS levels for all flows are preserved. If such an allocation is impossible, an adaptation procedure tries to compromise between QoS requirements for each flow. It consequently distributes resources according to the information about user perception, current network state and QoS mapping rules. Depending on its priority a particular flow might be gradually shut down until the required network resources are available again.

The traffic shaping mechanism uses information provided by Packet Classifier – responsible for the classification and marking of each application data flow - to reflect its traffic characteristics and user-perceived QoS requirements.

In the client-server model it is quite likely that there is be more than one connection to server so different user sessions can influence each other. Traffic Aggregator and Admission Control are responsible for mitigation of negative effects caused by sharing the same link among different users. Their role is to aggregate the traffic into one of the predefined classes of data flow and provide fair access to available resources within the same class. For data flow aggregates with different characteristics Admission Control component manages their interactions and optimizes network resource usage.

Possible adaptation scenario using presented architecture (see Figure 1):

Step 1. Packet Classifier, a part of a remote desktop sever, marks the traffic with distinctive characteristics and classifies into different virtual channels.

Step 2. Flow Management component is responsible for traffic prioritization and shaping.

Step 3. Traffic Aggregator and Admission control provide admission control functionality.

Step 4. Adaptation Engine assesses user perception of each application based on QoS mapping rules and perception rules.

Step 5. Adaptation Engine controls Flow Management and Admission Control modules using information about network performance provided by network monitoring and over return channel from end-user.

IV. CONCLUSIONS

In this paper we address the existing limitations of remote desktop computing over best-effort network, with regard to network-sensitive multimedia services and applications. We extend the functionality of the existing solutions by introducing architecture of a server-side QoS management component. In our architecture, flexible adaptation mechanisms can dynamically map user-perceived QoS defined for each application and terminal device to the current network state in order to maximize user experience. Additionally, by combining perception information with traffic management techniques we are able to prioritize the multimedia flows (such as video streaming) in a single thin-client connection as well as provide fair resource allocation among multiple user connections to the same server. The component architecture provides a set of QoS management mechanisms for remote desktop protocols and is a solid basis for further enhancements.

ACKNOWLEDGMENTS

The authors would like to thank the members of Citrix Systems Asia Pacific Pty Ltd., in particular S. Parry and T. Low for their valuable comments on this work.

REFERENCES

[1] Citrix MetaFrame Presentation Server - "Product Overview", Citrix Systems, available at: http://www.citrix.com/site/ps/products.asp

[2] Citrix Systems, "Citrix MetaFrame 1.8 Backgrounder", *Citrix White Paper*, June 1998.

[3] Microsoft Corporation, "Windows 2000 Terminal Services Capacity Planning", *Technical White Paper*, Redmond, WA, 2000.

[4] R. W. Scheifler and J. Gettys, "The X Window System", *ACM Transactions on Graphics*, 5(2), Apr. 1986.

[5] Tristan Richardson, Quentin Stafford-Fraser, Kenneth R. Wood, Andy Hopper, "Virtual Network Computing", *IEEE Internet Computing* 1999.

[6] Tarantella ® Enterprise 3 ™ Software, "A Technical Overview", *A Tarantella White Paper*, August 2001

[7] Albert Lai and Jason Nieh, "Limits of WideArea ThinClient Computing", *Proceedings of the ACM SIGMETRICS 2002*, Marina del Rey, CA, June 15-19, 2002.

[8] ITU T.125 Recommendation - Multipoint Comm. Service Protocol Specification, http://www.itu.int/ITU-T/asn1/database/itu-t/t/t125/1998/, February 1998.

[9] Ashton, Metzler & Associates. "The Three Components of Optimizing WAN Bandwidth", *Packeer White Paper*, March 2003

[10] Packteer Inc., "Controlling WAN Bandwidth and Application Traffic", 2004.

[11] Expand Inc., "Expand Networks Accelerators, A Technical Overview", April 2004.

[12] Citrix Systems, "Supporting Client Devices Using Citrix Virtual Channel SDK", *Citrix White Paper*, 2004.

[13] B. Vandalore, W. Feng, R. Jain, and S. Fahny. "A survey of Application Layer Techniques for Adaptive Streaming of Multimedia". *Journal of Real-time Systems (Special Issue on Adaptive Multimedia)* , January 2000

[14] D. Wu, Y. T. Hou, W. Zhu, Y-Q Zhang, and J. M. Peha. "Streaming Video over the Internet: Approaches and Directions". *IEEE Transactions on Circuirs and Systems for Video Technology*, VOL. 11, NO.3, March 2001.

[15] Mohamed Al-Turkistany and Abdelsalam (Sumi) Helal, "Intelligent Adaptation Framwork for Wireless Thin-Client Environments," – *Proceedings IEEE Symposium on Computers and Communications ISCC'2003*, July 2003

[16] Kumwilaisak, W.; Hou, Y.T.; Qian Zhang; Wenwu Zhu; Kuo, C.-C.J.; Ya-Qin Zhang "A cross-Layer quality-of-service mapping architecture for video delivery in wireless networks", *IEEE Journal on Selected Areas in Communications*, Page(s): 1685-1698

[17] C. Krasic, J. Walpole. "QoS Scalability for Streamed Media Delivery", *OGI CSE Technical Report CSE-99-011*, September, 1999.

[18] Wu-Chang Feng, Dilip D. Kandlur, Debanjan Saha, and Kang G. Shin. "Adaptive Packet Marking for Maintaining End-to-End Throughput in a Differentiated-Services Internet", *IEEE/ACM Transactions on Networking*, VOL. 7, NO. 5, OCTOBER 1999

[19] J. C. De Martin, D. Quaglia. "Disortion-based Packet Marking for MPEG Video Transmission over DiffServ Networks", *Proceedings IEEE International Conference on Multimedia and Expo*, 2001

[20] ISO/IEC 1446-6 CD, "Delivery Multimedia Integration Framework, DMIF", October 1997.

[21] Jean-François Huard, Aurel A. Lazar, Koon-Seng Lim, and George S. Tselikis, Xbind, Inc., "Realizing the MPEG-4 Multimedia Delivery Framework", *IEEE Network*, November/December 1998.

[22] Terms and definitions related to quality of service and network performance including dependability, *ITU-T Recommendation E.800*, August 1994

[23] Communications Quality of Service: A framework and definitions, *ITU-T Recommendation G.1000, Revised Version*, November 2001

[24] Surya Nepal, Uma Shrinivasan, "Adaptive Video Highlights for Wired and Wireless Platforms", *Proceedings IEEE International Conference on Multimedia and Expo,* Baltimore, USA, July 2003.

[25] Surya Nepal and Uma Srinivasan. "Quality Adaptation of Video for Delivery in Both Wired and Wireless Platform", *CSIRO MIS Report*, 2002.