

DESKTOP SHARING IN AUDIO AND VIDEO CONFERENCES

Martin Zukal

Master Degree Programme (2), FEEC BUT

E-mail: xzukal02@stud.feec.vutbr.cz

Supervised by: Petr Číka

E-mail: cika@feec.vutbr.cz

ABSTRACT

This contribution is devoted to the desktop sharing description and the design and the implementation of a new application for desktop sharing. The proposed solution utilizes the RFB protocol and the Eclipse Rich Client Platform. Both of these technologies are shortly introduced in the text. The paper is ended up by the description of current state of the work.

1 ÚVOD

Jedním z oblíbených kolaboračních nástrojů je sdílení plochy. Sdílení plochy umožňuje v nej-jednodušším případě propojení dvou počítačů a to tak, že jeden z nich je prezentátorem a druhý pasivně sleduje, co se děje na ploše prezentátora. Pasivních účastníků sledujících plochu prezentátora může být i více. V těchto případech se jedná o řešení podobající se videokonferenci. Ve spojení s dalšími prostředky, jako je na příklad konverzace IM (Instant Messaging) případně hlasová komunikace (většinou s využitím VoIP), se jedná o nástroj, který nabízí široké možnosti využití. Jako příklad uved' me výukové účely: žáci a vyučující mohou být od sebe vzdáleni tisíce kilometrů, ale s využitím sdílení plochy mohou žáci vidět vše, co vyučující na svém počítači provádí.

2 VIRTUAL NETWORK COMPUTING

Virtual Network Computing [1] je systém pro sdílení plochy založený na protokolu RFB (Remote Framebuffer [2]). Protokol RFB zahrnuje jak protokol pro ovládání relace (navázání a ukončení spojení a další signalizace), tak zobrazovací protokol (samotný přenos dat reprezentujících plochu počítače). Jedná se o protokol založený na modelu klient – server. Server běží na stroji jehož plocha má být zobrazována (případně ovládána) a klient je program běžící na počítači, kde se má plocha zobrazovat. Protokol RFB využívá pro přenos zpráv spolehlivé spojení (nejčastěji Transmission Control Protocol – TCP). Zprávy protokolu RFB jsou binární a jsou vždy zarovnané tak, že jejich délka je násobkem 4 bytů. Pokud zpráva sama o sobě uvedené kritérium nesplňuje, je použita výplň (padding).

Zobrazovací protokol je založen na jednoduché primitivě: “put a rectangle of pixel data at a given x,y position” [2]. Snímek plochy se rozdělí na menší pravouhlé oblasti, které jsou odeslány

ke klientovi, a ten z nich znovu sestaví výslednou plochu. Intervaly pro posílání nového snímku stanovuje klient v závislosti na svých možnostech (rychlost připojení, výpočetní výkon apod.). Před samotným odesláním snímku ke klientovi je možné na odchozí data aplikovat různá kódování, jejichž účelem je snížit objem přenášených dat.

3 ŘEŠENÍ

Vyvíjené řešení by mělo splňovat následující požadavky:

- kompatibilita s jinými aplikacemi,
- přenositelnost mezi platformami,
- rozšiřitelnost.

Aby byla zajištěna přenositelnost mezi různými platformami, bude aplikace napsána v programovacím jazyce Java. Pro docílení kompatibility s jinými aplikacemi se bude jednat o implementaci protokolu RFB.

3.1 KNIHOVNA PRO VNC

Před započatím vývoje byl proveden průzkum, zda neexistuje knihovna napsaná v jazyce Java, která by již protokol RFB implementovala a kterou by bylo možné využít pro další práci. Jediná existující knihovna [3] se ukázala jako nevhodná (jednak implementuje zastaralou verzi protokolu RFB a v některých ohledech je autory knihovny zvolená implementace poměrně nešťastná). Proto bylo rozhodnuto vytvořit vlastní implementaci protokolu RFB.

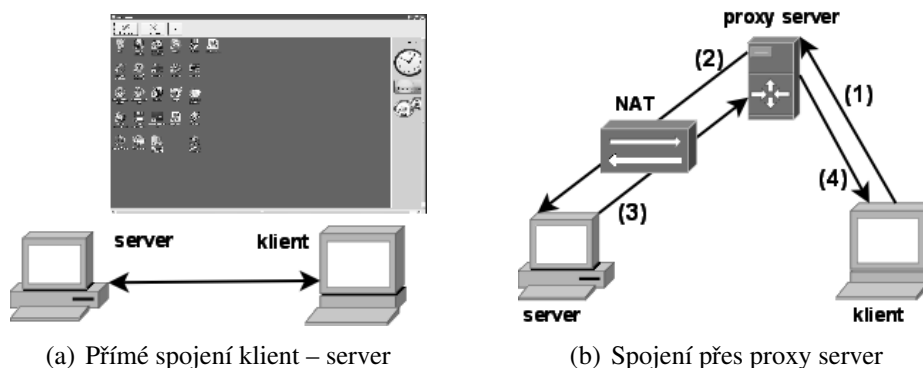
3.2 RICH CLIENT PLATFORM

Eclipse Rich Client Platform (RCP) [4] je open-source platforma pro tvorbu vizuálně bohatých aplikací v jazyce Java. Jedná se o framework, který nabízí prostředí pro tvorbu modulárních aplikací v jazyce Java. Eclipse RCP dále využívá tzv. Extension / extension point model, který umožňuje definovat body, na které je možné navázat případná rozšíření aplikace. Právě díky využití Eclipse RCP a vhodným vytvořením bodů pro rozšíření bude dosaženo požadované rozšiřitelnosti aplikace.

3.3 ARCHITEKTURA SYSTÉMU

Jako vhodné řešení, které se nabízí, je přímá komunikace mezi klientem a serverem, jak to ukazuje Obrázek 1(a). Jedná se o klasický model klient–server, kdy server naslouchá na předem známém portu a klient se k němu připojuje. Toto řešení bude dobře fungovat jak v lokální síti, tak i v případě, kdy server disponuje veřejnou IP adresou.

V síti Internet je častým jevem využití techniky zvané Network Address Translation (NAT). Odchozí spojení není NATem nijak omezeno, avšak většina routerů podporujících NAT je nakonfigurována tak, že nové příchozí spojení nepovolí. Proto je nutné router buď správně nastavit tak, aby např. veškerou komunikaci směřující na zvolený port posílal na vybraný počítač v lokální síti za routerem, nebo je nutné využít mezilehlého uzlu (často označovaného jako proxy server). Úkolem tohoto uzlu je udržování spojení s počítačem umístěným za NATem



Obrázek 1: Možnosti fungování

pomocí tzv. keep alive zpráv. Klient, který chce komunikovat s počítačem nacházejícím se za NATem nekomunikuje přímo s ním, ale právě s proxy serverem, který přeposílá komunikaci mezi koncovými uzly. Příklad komunikace ukazuje Obrázek 1(b): proxy server obdrží zprávu od klienta (1), přepośle ji serveru (2), vyčká na jeho odpověď (3) a tu pak zašle klientovi (4). Proxy server má tedy dvě role. Vůči klientovi vystupuje jako server a vůči serveru vystupuje jako klient.

4 ZÁVĚR

Aplikace je nyní ve fázi vývoje, který probíhá iterační metodou: nejprve bylo implementováno jádro aplikace a v každé iteraci se postupně přidávají požadované funkce. Jádro aplikace obsahuje pouze takové funkce, které zajišťují správné fungování aplikace. Mezi základní funkce patří navázání a ukončení spojení se serverem, výměna inicializačních zpráv a pravidelná výměna aktualizovaných snímků plochy a jejich zobrazování. V neposlední řadě bylo vytvořeno grafické uživatelské rozhraní umožňující základní ovládání aplikace. V dalších iteracích byla věnována pozornost implementaci kompresních algoritmů vhodných pro kódování snímků plochy. Základní tzv. RAW kódování (tj. přenáší se přímo hodnoty jednotlivých pixelů), které musí podporovat každý klient, je součástí jádra aplikace.

Kromě tvorby samotných zdrojových kódů byl kladen důraz i na vytváření jednotkových testů, které budou v konečné fázi sloužit k ověření správné funkčnosti všech částí vytvářené aplikace.

REFERENCE

- [1] *Virtual Network Computing : from ORL* [online]. The Olivetti & Oracle Research Lab, c1998 [cit. 2010-02-08]. Dostupné z URL: <<http://grox.net/doc/apps/vnc/index.html>>.
- [2] RICHARDSON, Tristan. *The RFB Protocol* [online]. Version 3.8. RealVNC Ltd, 2009 , Last updated 5 August 2009 [cit. 2010-02-08]. Dostupné z URL: <<http://www.realvnc.com/docs/rfbproto.pdf>>.
- [3] *LGPL VNCj* [online]. c2010 [cit. 2010-02-08]. Dostupné z URL: <<http://sourceforge.net/projects/vncjlgpl/>>.
- [4] MALÉŘ, Jakub. *Eclipse RCP (Rich Client Platform)* [online]. prosinec 2005 [cit. 2010-02-08]. Dostupné z URL: <<http://nb.vse.cz/zelenyj/it380/eseje/xmalj35/EclipseRCP.htm>>.