UNIVERZITA PARDUBICE
Fakulta elektrotechniky a informatiky

Analýza protokolů pro vzdálenou správu

Ondřej Krejčí

Bakalářská práce
2014

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: **2013/2014**

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Ondřej Krejčí**
Osobní číslo:	**I10100**
Studijní program:	**B2646 Informační technologie**
Studijní obor:	**Informační technologie**
Název tématu:	**Analýza protokolů pro vzdálenou správu**
Zadávající katedra:	**Katedra informačních technologií**

Zásady pro vypracování:

Cílem práce je provést komparativní analýzu dostupných protokolů pro vzdálenou správu a přístup. Autor práce v teoretické části představí a srovná moderní protokoly (VNC a jeho varianty, RDP a jeho verze a Terminal Services). V praktické části práce se bude autor zabývat implementací protokolů pro vzdálenou správu, provede jejich implementaci a odchycení provozu pomocí programu Wireshark. Dále bude poukázáno na zásadní rozdíly mezi jednotlivými protokoly a jejich možnostmi jejich nasazení.

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

Articles on Remote Desktop, Including: Citrix Systems, Independent Computing Architecture, Technical Support, Remote Desktop Protocol, Citrix Xenapp,. 1. vyd. Hephaestus Books: Hephaestus Books, 2011. ISBN 9781242970122.
SURHONE, Lambert a Miriam TIMPLEDON. Remote Desktop Protocol. 1. vyd. Betascript Publishing, 2010. ISBN 9786130418823

Vedoucí bakalářské práce: **Ing. Soňa Neradová**
Katedra softwarových technologií

Datum zadání bakalářské práce: **20. prosince 2013**
Termín odevzdání bakalářské práce: **9. května 2014**

Univerzita Pardubice Česká republika

L.S.

prof. Ing. Simeon Karamazov, Dr.
děkan

Ing. Lukáš Čegan, Ph.D.
vedoucí katedry

V Pardubicích dne 31. března 2014

Prohlášení autora

Prohlašuji, že jsem tuto práci vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 7. 5. 2014 Ondřej Krejčí

Poděkování

Tímto bych rád poděkoval Ing. Soně Neradové za věcné rady a trpělivost v průběhu zpracování této bakalářské práce. Poděkování také patří mé rodině za podporu a vytvoření kvalitního zázemí během studia na vysoké škole.

Anotace

Cílem práce je provést komparativní analýzu dostupných protokolů pro vzdálenou správu a přístup. Autor práce v teoretické části představí a srovná moderní protokoly (VNC a jeho varianty, RDP a jeho verze a Terminal Services). V praktické části práce se bude autor zabývat implementací protokolů pro vzdálenou správu, provede jejich implementaci a odchycení provozu pomocí programu Wireshark. Dále bude poukázáno na zásadní rozdíly mezi jednotlivými protokoly a jejich možnostmi jejich nasazení.

Klíčová slova

VNC, RDP, RFC, VPN, SSH, Wireshark

Title

Analysis of protocols for remote management

Annotation

The aim of thesis is carry out a comparative analysis of available protocols for remote management and access. The author of thesis in theoretical part will introduce and compare modern protocols (VNC and his variants, RDP and his versions and Terminal Services). In the practical part of the thesis the author will address the implementation of the protocol for remote management, performs their implementation and catching traffic using Wireshark. It will also be pointed out major differences between individual protocols and their potential deployment.

Keywords

VNC, RDP, RFC, VPN, SSH, Wireshark

Obsah

Seznam zkratek

3DES	Triple DES
AES	Advanced Encryption Standard
CBC	Cipher Block Chaining
DES	Data Encryption standard
EAP-TLS	Extensible Authentication Protocol - Transport Layer Security
FIPS	Federal Information Processing Standards
GPL	General Public Licence
GSSAPI	Generic Security Service Application Program Interface
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
ISO/OSI	International Standards Organization/Open Systems Interconnection
ITU	International Telecommunication Union
L2F	Layer 2 Forwarding
L2TP	Layer Two Tunneling Protocol
MPPE	Microsoft Point-to-Point Encryption
MS-CHAP	Microsoft - Challenge-Handshake Authentication Protocol
PDU	Protocol Data Unit
PPP	Point-to-Point Protocol
PPTP	Point-to-Point Tunneling Protocol
RC4	Arcfour
RDP	Remote Desktop Protocol
RFB	Remote Framebuffer
RLE	Run-length encoding
RSH	Remote Shell
SSH	Secure Shell
SSL	Secure Sockets Layer
SSTP	Secure Socket Tunneling Protocol
STLS	Specified Transfer Listing Service
TCP/IP	Transmission Control Protocol / Internet Protocol
TLS	Transport Layer Security
VNC	Virtual Network Computing
VPN	Virtual Private Network
WAN	Wide Area Network
ZRLE	Zlib Run-Lenght Encoding

Seznam obrázků

Seznam tabulek

Úvod

V dnešní době, kdy čas hraje velkou roli, může vzdálené ovládání počítače ušetřit nejen mnoho času, ale i finančních nákladů. Díky vzdálené správě uživatel nemusí být fyzicky přítomen u spravované platformy. Vzdálená správa může být využita nejen v komerční sféře, ale i k pomoci méně zkušeným uživatelům. Další velkou výhodou je možnost správy více počítačů nebo inteligentních zařízení jako smartphone, serverových či síťových zařízení.

Hlavním cílem této bakalářské práce je provést podrobnou analýzu dostupných protokolů pro vzdálenou správu a přístup. Celá komunikace je zachycena pomocí síťového nástroje, který je schopen zachytit kompletní příchozí i odchodí pakety na zkoumaném síťovém rozhraní.

První kapitola práce je věnována základním pojmům spojenými se vzdálenou správou. Je v ní nastíněn proces zachycení a analýzy paketů. Dále v ní jsou podrobně vysvětleny tunely pro zabezpečení komunikace vzdálené správy mimo lokální síť. Součástí této kapitoly je i krátký popis referenčního ISO/OSI modelu a jeho vrstvy spojené s protokoly RDP, RFB a VNC.

Druhá kapitola práce je zaměřena na představení VNC. Obsahuje především popis historie dálkové správy, vlastnosti a zabezpečení původního protokolu VNC. V další části kapitoly je podrobně popsána architektura protokolu RFB, moderní protokoly VNC a jeho varianty. Závěr druhé kapitoly tvoří implementace, zachycení provozu a porovnání nabízených služeb zmíněných protokolů VNC.

Třetí kapitola této práce má za cíl představit protokol RDP. Je zde podrobně popsána architektura protokolu společně s verzemi RDP protokolu. Dále jsou v ní představeny RDP klienti a Terminal Services pro různé operační systémy. Na závěr této kapitoly je provedena implementace protokolu na dva operační systémy a zachycení komunikace.

Čtvrtá kapitola je zároveň poslední kapitolou této práce. Je zaměřena především na porovnání protokolů, které slouží ke vzdálené správě. Je zde poukázáno na zásadní rozdíly mezi protokoly VNC a RDP. Závěrem této kapitoly jsou popsány možnosti nasazení zkoumaných protokolů v praxi.

V této bakalářské práci jsou veškeré obrazové dokumentace vytvořeny autorem, pokud není uvedeno jinak.

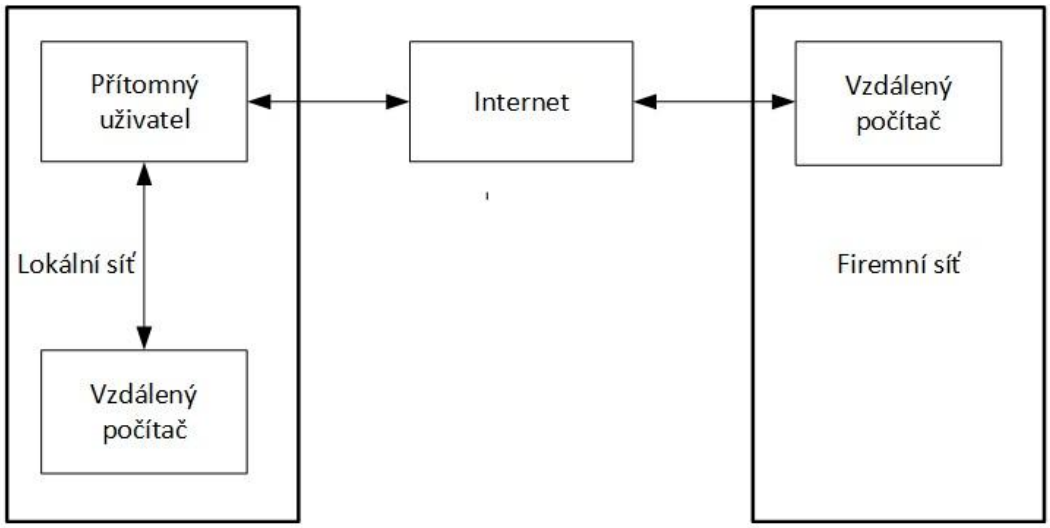
1 Vzdálená správa

Vzdálená správa je služba, která umožňuje zásah na počítači uživatele bez nutnosti fyzického kontaktu s počítačem. Počítač lze efektivně spravovat buď vzdáleně v lokální síti, nebo mimo lokální síť. Hlavní motivací pro zavedení vzdálené správy je:

- instalace a nastavení nových zařízení či aplikací,
- vzdálená aktualizace operačního systému,
- podpora klienta při správě počítače/ serveru,
- podpora pro ICT outsourcing,
- zvýšení efektivity práce.

Velkou výhodou jsou současné programy, které uživatelům nabízí snadnou instalaci a jejich ovládání nevyžadují přílišné znalosti v dané problematice.

Hlavní nevýhodou vzdálené správy je její bezpečnost. Pokud není přenos dostatečně chráněn, může dojít vcelku snadno k jeho odposlouchávání.



Obrázek 1 – Blokové schéma vzdálené správy

2 Základní pojmy

2.1 Analýza paketů a program Wireshark

Analýza paketů je proces, při kterém dochází k zachycení a ztvárnění aktuálních dat přenášených po síti. Díky tomu lze lépe porozumět komunikaci v dané síti a odstranit případné závady. Analýza paketů nám umožňuje zjistit:

- uživatele v síti,
- vlastnosti sítě,
- špičkový čas využití sítě,

- neefektivní a nezabezpečené aplikace,
- nebezpečné aktivity a možné útoky,
- kdo a co využívá dostupnou šířku pásma.

Zachycení a analýzu paketů umožňuje paketový sniffer. (Sanders, 2012, s. 20-21) K dispozici existuje mnoho nástrojů, které jsou produkovány jak v komerční tak i bezplatné verzi. Každý nástroj je navržen k jiným účelům a při jejich výběru je nutné zohlednit nejméně tyto faktory:

- podporované protokoly,
- podpora operačního systému,
- podpora programu a dokumentace,
- pořizovací náklady.

V této práci byl zvolen program Wireshark.

Program Wireshark byl vytvořen Geraldem Combsem v roce 1998 s původním názvem Etheral. Wireshark je bezplatný software šířený pod licencí GPL, která umožňuje osobní i komerční využití. (Sanders, 2012, s. 53-55)

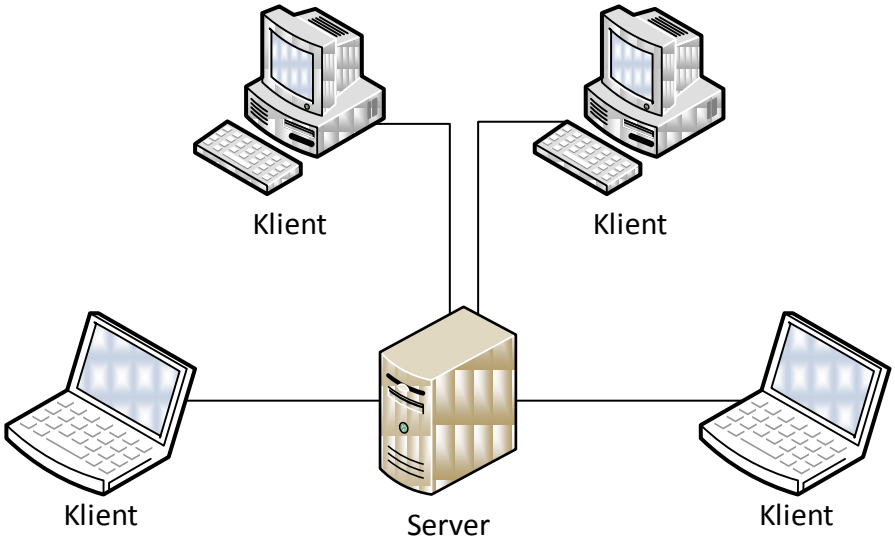
2.2 Klient-server

Síťová architektura klient-server nejen ve vzdálené správě rozlišuje dva typy koncových zařízení, která mezi sebou komunikují prostřednictví počítačové sítě. Zařízení se označují jako klient a server. Komunikace probíhá na základě dotazů a odpovědí. (HORÁK, a další, 2006, s. 49)

Klientem je označováno zařízení, které zajišťuje uživatelské rozhraní. Klient zahájí komunikaci vysláním dotazu na server. Z hlediska požadavků na hardware, klient zpravidla obsahuje vstupně - výstupní zařízení.

Server je jádrem počítačové sítě. V něm jsou shromažďována a jím jsou zároveň i zpracovávána data. Server tato data a služby poskytuje klientům, na základě jejich požadavků. Obvykle se náročnější operace provádějí na straně serveru. Tím je klient ušetřen zpracovávat náročné operace a dostává pak pouze výsledné informace.

Výhodou této architektury je nezávislost platformy klienta na platformě serveru a opačně. Při správné a účelné komunikaci lze za pomocí architektury klient-server minimalizovat objem přenášených dat.

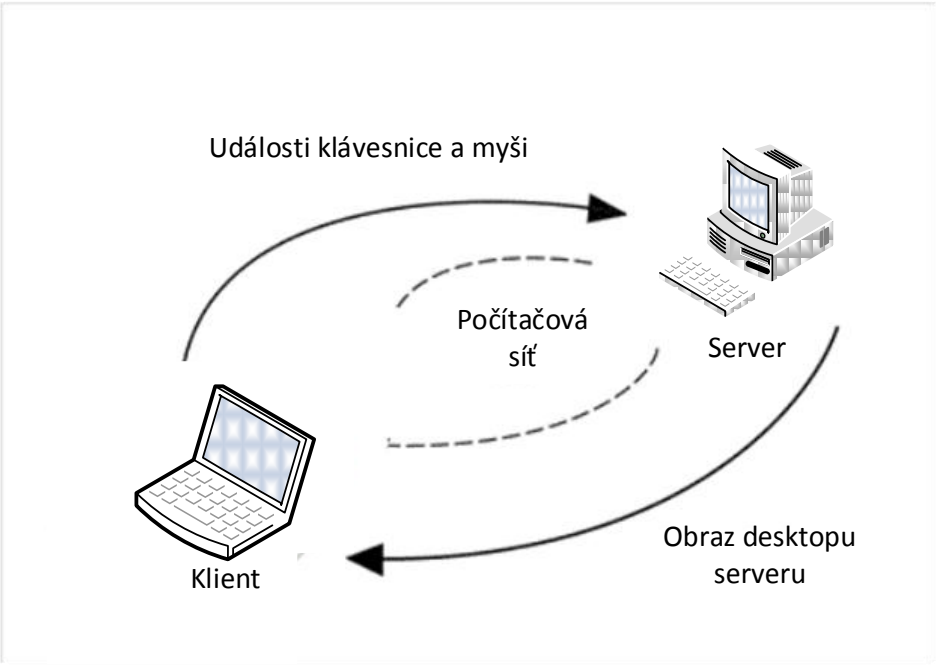


Obrázek 2 – Obecný model klient-server

Z pohledu vzdálené správy lze klient-server chápat takto.

Server je program, který sdílí prostředky a obrazovku. Klient je program, který interaguje se serverem. Lze jednoduše říci, že klient ovládá server. Pokud dojde ke ztrátě spojení a jeho následnému obnovení, server a všechny jeho aplikace zůstanou ve stejném stavu, jakém byly před ztrátou spojení.

Hlavním účelem komunikace klient-server je snaha o minimalizování objemu dat přenášených po síti. Model klient-server využívají aplikace pro vzdálenou správu počítače typu VNC i RDP. (WHITE, 2007, s. 282-283)



Obrázek 3 – Vzdálená správa klient-server

2.3 Veřejná a neveřejná IP adresa

Veřejná IP adresa představuje jedinečnou adresu, která identifikuje zařízení v rámci celé sítě Internet. Tato adresa navíc umožňuje přímé spojení protokolů TCP/UDP. Dostupnost z celé sítě Internet zvyšuje riziko útoku. Z tohoto důvodu je nutné počítač chránit pomocí síťových prvků. Veřejnou IP adresu přiděluje poskytovatel internetového spojení. Z hlediska vzdálené správy je veřejná IP adresa velice vhodná, ale nikoli nezbytná. I přes správně nastavenou veřejnou IP adresu, může nastat konflikt při požadavku o spojení. Ten neprojde skrz místní router. V takovém případě je možné provést konfiguraci routeru neboli přesměrování portů.

Neveřejná IP adresa je opakem IP adresy veřejné. Tato IP adresa je zpravidla z Internetu nepřístupná. Z pohledu vzdálené správy mimo lokální síť existuje řešení. Tím je zavedení vlastní privátní sítě VPN, tunelového spoje nebo pomocných programů např. LogMeIn Hamachi.

2.4 VPN

VPN je označení pro virtuální privátní síť umožňující propojení koncových zařízení nebo celých sítí skrz veřejnou počítačovou síť. Propojením sítí pomocí VPN lze vytvořit privátní zabezpečenou linku z veřejné nezabezpečené sítě. Tím je odstraněna hrozba odposlechu. Takto vytvořené VPN okruhy jsou podobné linkám mezi sítěmi u rozsáhlých WAN sítí. Z uživatelského pohledu jsou síťové prostředky přístupné stejným způsobem jako v rámci jedné lokální sítě.

Laicky řečeno při propojení sítí nebo koncových zařízení pomocí VPN je dosaženo stavu, ve kterém zařízení mohou komunikovat, jako kdyby byly propojeny v jedné lokální síti. Toho je možné docílit pomocí tunelovacích protokolů obsažených v TCP/IP modelu.

2.4.1 Zabezpečení VPN

Jak již bylo uvedeno výše VPN vytváří zabezpečenou linku a brání tím odposlechu komunikace a ztrátě dat. Hlavní úkoly zabezpečení jsou:

- Autentizace - zabrání neoprávněnému vstupu do virtuální primární sítě.
- Integrita dat - odhalí manipulaci s daty u přenášených zpráv.
- Šifrování dat - při zachycení paketu pomocí snifferu, útočník uvidí pouze šifrované informace.

Zabezpečení je zajištěno pomocí tunelovacích protokolů a šifrování. Vytvořené spojení pomocí tunelu zapouzdří pakety protokolu do datagramu jiného protokolu. K řešení se používají standardně protokoly PPTP, L2TP a SSTP. Protokoly částečně vycházejí z funkcí protokolu PPP.

Protokol PPTP při přenosu zapouzdřuje rámce protokolu PPP do datagramů IP. PPP rámce jsou šifrovány pomocí MPPE s využitím šifrovacích klíčů MS-CHAPv2 nebo EAP-TLS. Šifrovací klíč MS-CHAPv2 byl prolomen a není považován za bezpečný.

Následný protokol L2TP vychází z protokolů PPTP a L2F. Protokol využívá šifrovací služby IPsec protokolu. Šifrování zpráv je založeno na DES nebo 3DES algoritmu. Oproti PPTP již nepodporuje MPPE šifrování PPP datagramů.

Nejnovější protokol SSTP využívá SSL kanálu protokolu HTTPS pro zapouzdření PPP přenosu. (MICROSOFT)

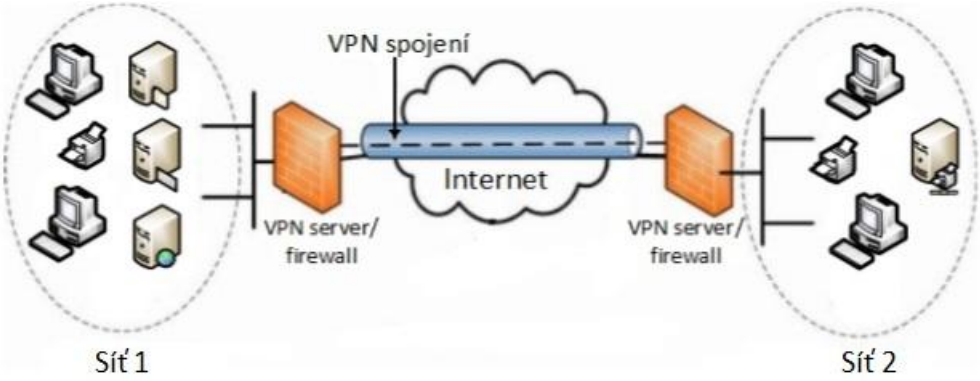
2.4.2 Dělení VPN

VPN linky lze rozdělit dle zapouzdření:

- Tunelovaná linka
  - zapouzdřený a zašifrovaný je celý paket.
- Transportní linka
  - zapouzdřená a zašifrovaná je pouze datová část paketu.

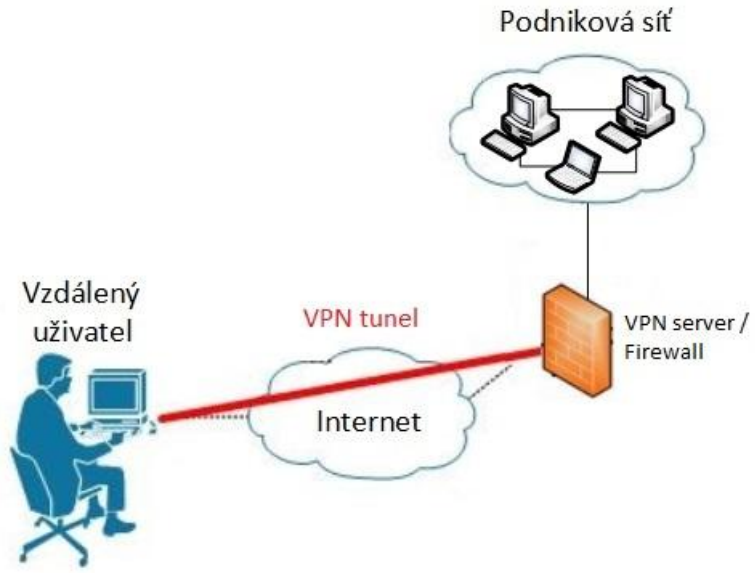
VPN lze dále rozdělit na typy Site-to-site a Remote access.

Site-to-site VPN je označení pro spojení dvou vzdálených sítí. Tato možnost je využívána pro spojení např. dvou vzdálených firemních poboček. Pro úspěšné spojení je nutné vytvořit nejprve spojení lokální sítě s VPN serverem. Následně lze vytvořit VPN mezi servery.



Obrázek 4 – Site-to-site VPN

Remote Access VPN je označení pro vzdálený přístup na server z domácí sítě s využitím veřejné sítě. Tato možnost je využívána např. pro připojení klienta do podnikové privátní sítě.



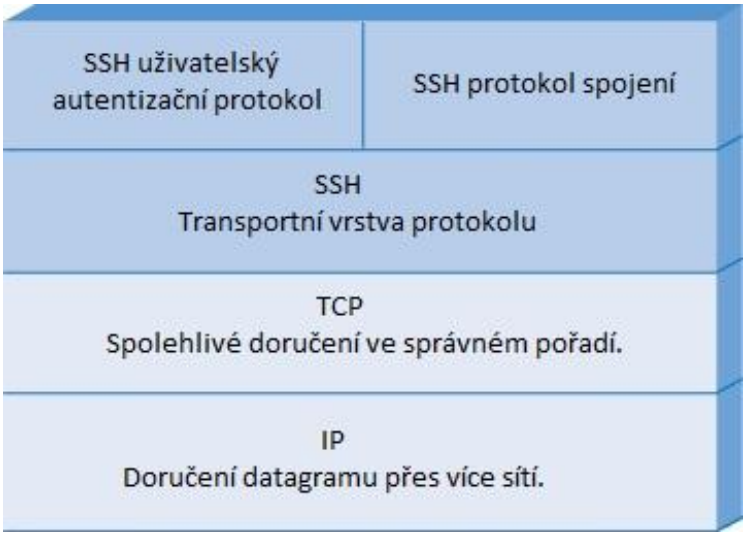
Obrázek 5 – Remote Access VPN

2.5 SSH

SSH protokol vznikl pro zabezpečení síťové komunikace s cílem jednoduché a levné realizace. Počáteční verze SSH-1 měla nahradit stávající nezabezpečené protokoly Rsh, Telnet a další systémy vzdáleného přihlášení. Rok později od zveřejnění SSH verze 1 byl vydán SSH protokol verze 2. Ten vylepšil SSH-1 v mnoha ohledech, bohužel ale nezajistil kompatibilitu s SSH-1.

SSH protokol je realizován na principu klient-server. Protokol je dostupný pro většinu operačních systémů čímž se stal oblíbenou volbou pro vzdálené přihlášení a tunelování. Protokol naslouchá na portu 22. Struktura protokolu je složena ze tří částí:

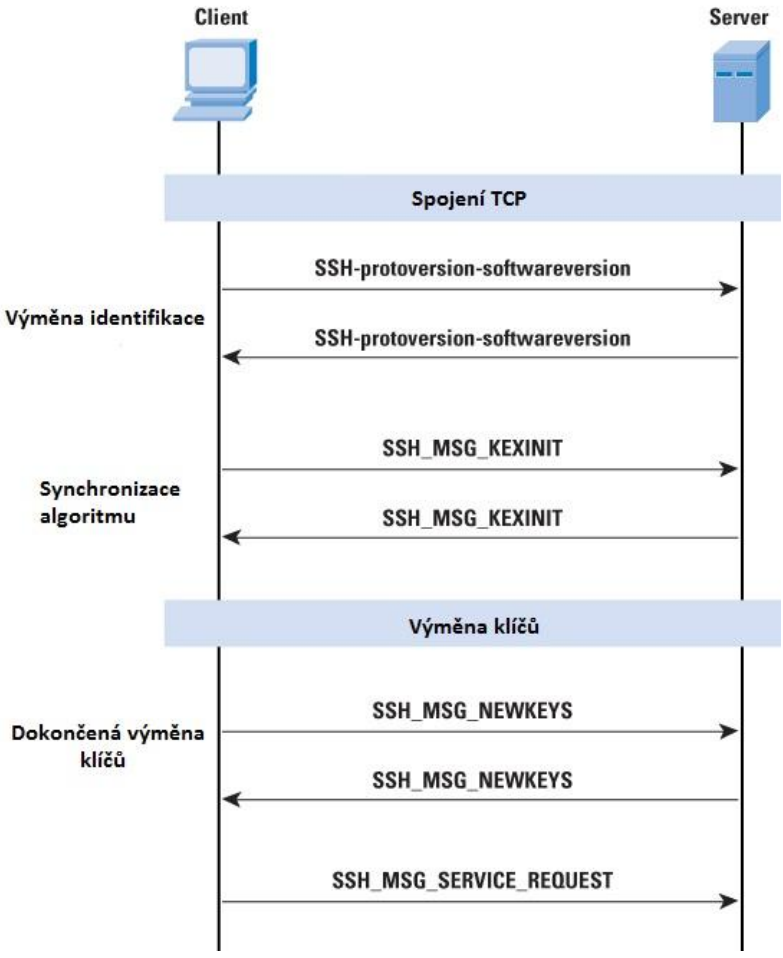
- transportní vrstva,
- autentizační vrstva,
- vrstva spojení.



Obrázek 6 – SSH Protokol Stack

2.5.1 Transportní vrstva

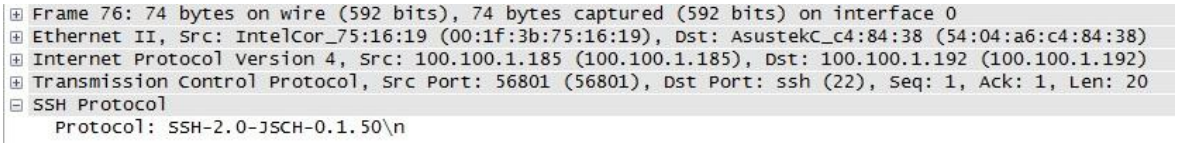
Transportní vrstva protokolu SSH zajišťuje autentizaci serveru, kompresi, počáteční výměnu klíčů a ověření integrity.



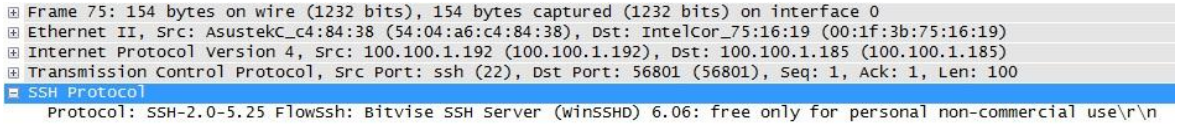
Obrázek 7 – Transportní vrstva, výměna paketů

Funkce transportní vrstvy lze rozdělit do tří základních kroků, jak je zobrazeno na obrázku Obrázek 7.

V prvním kroku dojde k navázání TCP spojení mezi klientem a serverem. Po navázání spojení si klient a server vymění paket, který obsahuje identifikační řetězec ve tvaru *SSH-protoversion-softwareversion SP komentář CR LF*. Řetězec přenáší informace o verzi protokolu, softwaru a znaky ukončení.

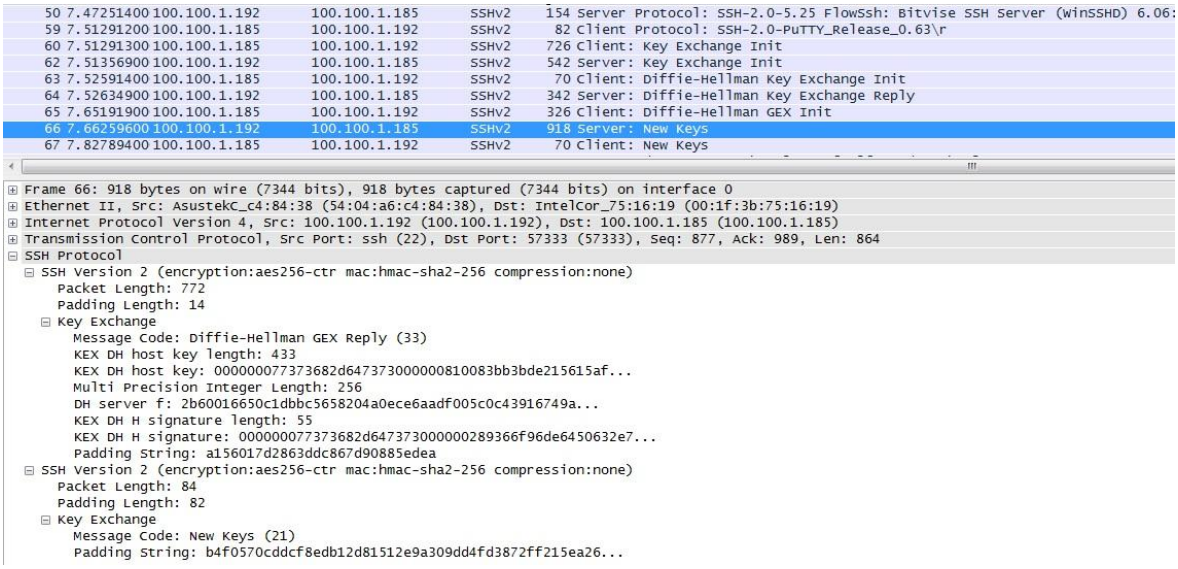


Obrázek 8 – Paket s verzí protokolu na straně klienta



Obrázek 9 – Paket s verzí protokolu na straně serveru

Druhý krok slouží k synchronizaci šifrovacího algoritmu. Klient a server si vzájemně vymění *SSH_MSG_KEXINIT* pakety. Tyto pakety obsahují seznam seřazených a podporovaných šifrovacích algoritmů. Algoritmy zahrnují výměnu klíčů, šifrování, algoritmus komprese a MAC algoritmus. Poté je vybrán první společně podporovaný algoritmus. Pořadí je určeno ze seznamu klienta.



Obrázek 10 – SSH, proces výměny klíčů

Tabulka 1 – Definované šifrovací algoritmy

Šifra	Popis	Použití
3DES-CBC	3DES v CBC módu	povinné
Blow fish-cbc	Blow fish v módu CBC	volitelné
Twofish256-CBC	Twofish v CBC módu s 256bit klíčem	volitelné
Twofish192-CBC	Twofish v CBC módu s 192bit klíčem	volitelné
Twofish128-CBC	Twofish v CBC módu s 128bit klíčem	volitelné
AES256-CBC	AES v módu CBC s 256bit klíčem	volitelné
AES192-CBC	AES v módu CBC s 192bit klíčem	volitelné
AES128-CBC	AES v módu CBC s 128bit klíčem	doporučené
Serpent256-CBC	Serpent v módu CBC s 256bit klíčem	volitelné
Serpent192-CBC	Serpent v módu CBC s 192bit klíčem	volitelné
Serpent128-CBC	Serpent v módu CBC s 128bit klíčem	volitelné
Arcfour	RC4 s 128bit klíčem	volitelné
Cast128-CBC	Cast 128 v režimu CBC	volitelné

Poslední krok slouží k výměně klíčů. Klient a server si vymění pakety *SSH_MSG_NEWKEY*. Tyto pakety signalizují úspěšné dokončení výměny klíčů.

Závěrem klient odešle *SSH_MSG_SERVICE_REQUEST*, na který naváže protokol spojení nebo autentizační protokol. Od tohoto momentu je celá komunikace šifrována. (STALLINGS, 2010)

2.5.2 Autentizační vrstva

Autentizační vrstva poskytuje prostředky, kterými je klient ověřen na straně serveru. Na počátku autentizace klient odešle na server *SSH_MSG_USERAUTH_REQUEST* žádost. Tato žádost obsahuje uživatelské jméno, název služby (typicky SSH protokol) a název autentizační metody, která je použitá k ověření žádosti. Server přijme požadavek o autentizaci. V případě úspěšné autentizace odešle klientovi *SSH_MSG_USERAUTH_SUCCESS* odpověď a autentizační protokol je u konce. V opačném případě odešle *SSH_MSG_USERAUTH_FAITURE* odpověď. V ní je obsažen seznam jedné nebo více metod, které jsou potřebné k autentizaci. Server může požadovat jednu nebo více z následujících autentizačních metod:

- password,
- hostbased,
- publickey,

- GSSAPI,
- keyboard-interactive. (STALLINGS, 2010)

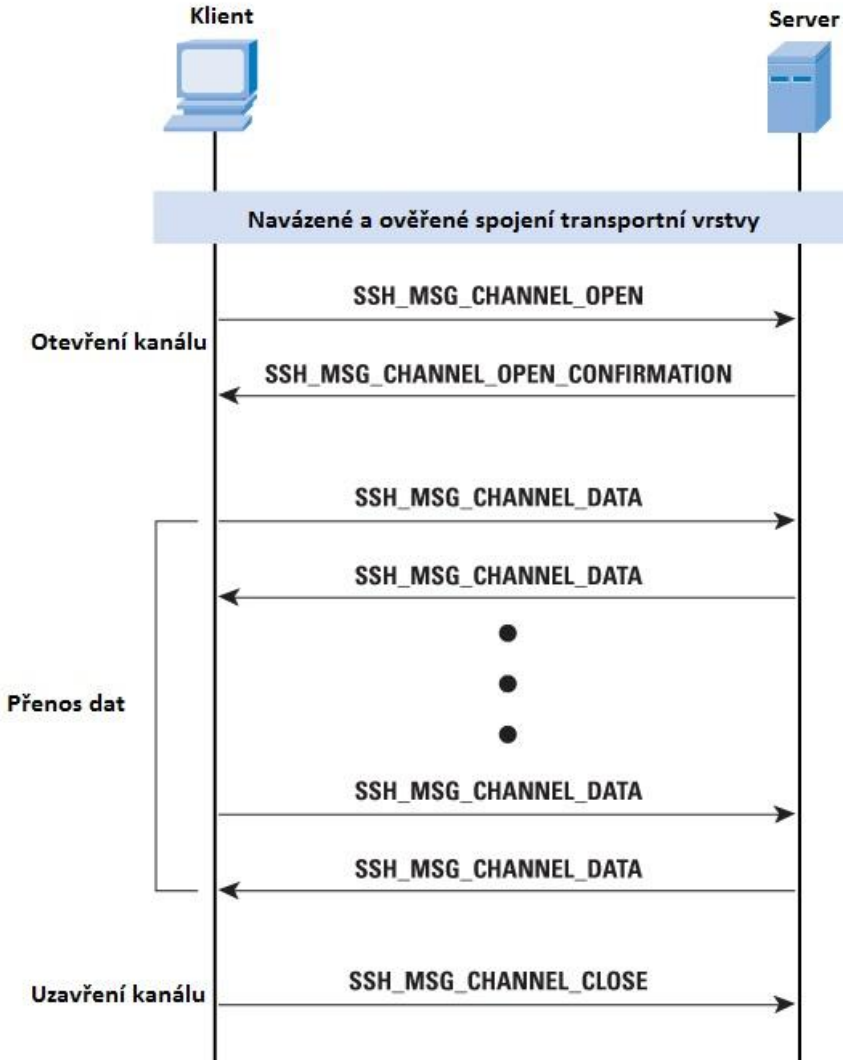
2.5.3 Vrstva spojení

Vrstva spojení definuje koncept kanálů a jejich požadavků, pomocí kterých jsou realizovány SSH služby. Všechny typy komunikace jako je např. terminálová relace jsou rozděleny na samostatné kanály. Pro každý kanál je definováno unikátní číslo, které nemusí být stejné na obou koncích komunikace.

Životnost kanálu je rozdělena do tří fází: otevření kanálu, přenos dat a uzavření kanálu. K otevření kanálu dochází zasláním zprávy *SSH_MSG_CHANNEL_OPEN*, která obsahuje číslo odesílatele kanálu, číslo příjemce kanálu a velikost paketu. V případě že kanál nebyl úspěšně vytvořen server odešle *SSH_MSG_CHANNEL_OPEN*_FAIURE zprávu, ve které je obsažen kód s příčinou selhání. Po úspěšném otevření kanálu probíhá přenos dat za použití zprávy *SSH_MSG_CHANNEL_DATA*. Tato obousměrná komunikace může pracovat do uzavření kanálu. V případě požadavku na uzavření kanálu dochází k odeslání *SSH_MSG_CHANNEL_CLOSE*, která obsahuje číslo příjemce kanálu a kanál je uzavřen.

SSH rozlišuje tyto typy kanálů:

- Direct,
- Forwarded-tcpip,
- Session,
- X11. (STALLINGS, 2010)



Obrázek 11 – Vrstva spojení, výměna zpráv

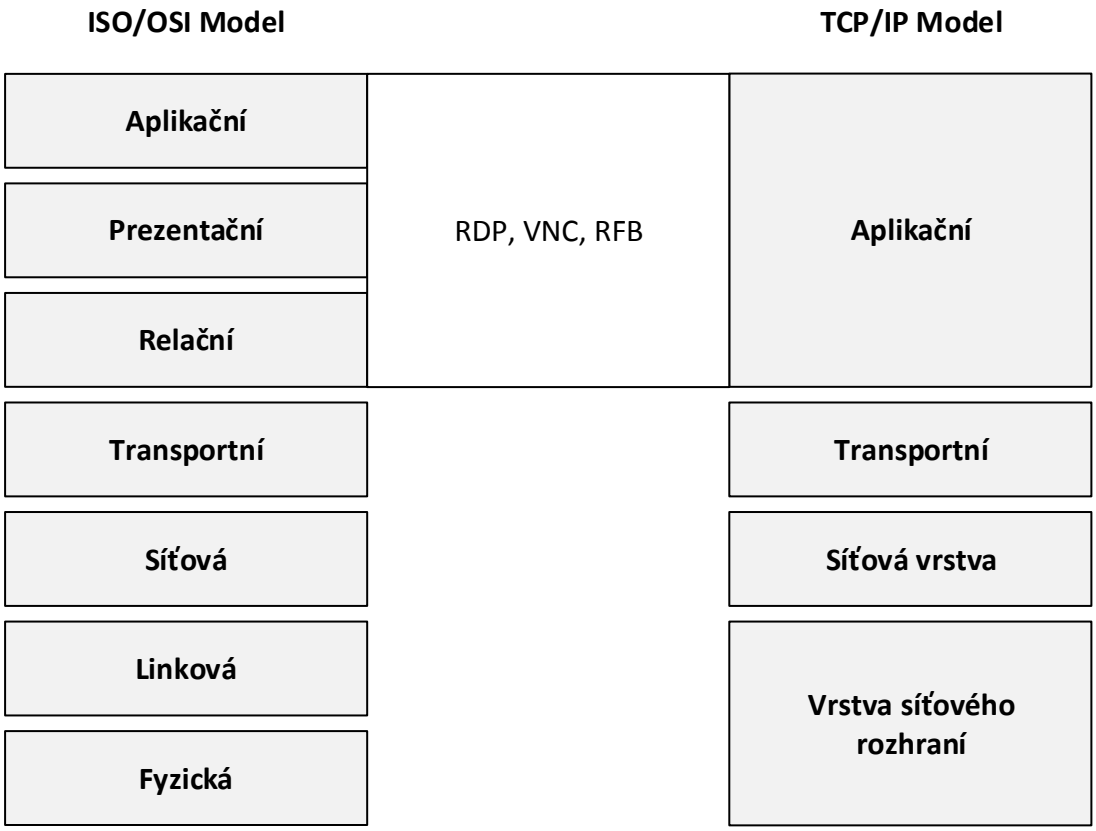
2.6 Tenký a tlustý klient

Tenký klient je entita, která při vykonávání své funkce silně závisí na svém serveru. Tento klient se zpravidla vyskytuje ve vícevrstvé architektuře. Nejčastější tenký klient je webový prohlížeč, který s prezentační vrstvou komunikuje přes HTTP protokol a zajišťuje pouze zobrazování dat. Nároky na jeho instalaci a konfiguraci jsou minimální.

Tlustý klient je entita v architektuře klient-server nebo síti, která obvykle obsahuje mnoho funkcí nezávislých na centrálním serveru. Tlustý klient stále vyžaduje alespoň občasné připojení k síti nebo centrálnímu serveru. Dokáže vykonávat řadu funkcí, aniž by musel být připojen. Na druhé straně tenký klient provádí nejmenší možný počet operací a při zpracování se dotazuje na server, kde jsou data zpracována nebo ověřena. Nároky na jeho instalaci a konfiguraci jsou podstatně náročnější.

2.7 Model ISO/OSI

Referenční model ISO/OSI standardizuje návrh řešení komunikačních zařízení v počítačové síti pomocí vrstveného modelu. Model nespecifikuje implementaci otevřeného systému. Uvádí pouze obecné principy a síťové struktury, které se skládají ze sedmi funkčních vrstev. V rámci modelu má každá vrstva jasně specifikovaný úkol, který bude řešit. Dále jsou specifikovány funkce a služby, které každá vrstva poskytuje vrstvě vyšší a také služby, které vyžadují od vrstvy nižší. Každá funkce vrstvy je implementována protokoly. Naopak jeden protokol může provést i několik funkcí. (ŠIMONOVÁ, 2004, s. 24-27)



Obrázek 12 – Porovnání ISO/OSI modelu s TCP/IP

2.7.1 Prezentační vrstva

Prezentační vrstva referenčního modelu ISO zajišťuje čitelnost informací odeslané z aplikační vrstvy do jiného systému aplikační vrstvy. V případě potřeby prezentační vrstva realizuje komprese přenášených dat, eventuálně i jejich šifrování. Pokud jsou přijata šifrovaná data, dojde k jejich dešifrování před předáním zprávy aplikační vrstvě. (KOLEKTIV, 2003, s. 737)

2.7.2 Aplikační vrstva

Aplikační vrstva referenčního modelu OSI poskytuje síťové služby, které jsou nejblíže k uživateli. Tato vrstva naváže komunikaci s určenými partnery a synchronizuje postupy pro

obnovení a kontrolu integrity dat. Protokol RDP pracuje na této vrstvě OSI modelu. Programy, jako jsou například RealVNC, Internet Explorer a další aplikace koncového uživatele, jsou příkladem programů, které využívají služby poskytované aplikační vrstvou pro uživatele. (KOLEKTIV, 2003, s. 737)

3 Představení VNC

VNC je systém sloužící k vzdálené správě počítače s plně grafickým uživatelským rozhraním. Kompletní systém se skládá z VNC serveru, klienta a komunikačního protokolu. Systém pracuje jako klient-server. Komunikace probíhá přes počítačovou síť pomocí protokolu RFB. Značnou výhodou VNC je nezávislost na platformě. Tato nezávislost umožňuje například připojení VNC klienta systému Linux k VNC serveru systému Windows.

3.1 Vznik VNC

Na počátku vývoje VNC stojí firma ORL (Olivetti & Oracle Research Lab). Původní zdrojový kód byl publikován jako open source pod licencí GPL. Za pomoci firmy AT&T byl v roce 2002 výzkum ukončen. Po uzavření ORL byl zformován nový projekt RealVNC, který pokračoval ve vývoji VNC. Nezávisle na RealVNC vznikly další implementace VNC jako jsou UltraVNC a TightVNC. Proto je možné setkat se různými výrobci a typy VNC. (LABORATORIES, 1999)

3.2 Vlastnosti VNC

VNC podporuje vícenásobné připojení klientů k jednomu serveru. Využití je v praxi efektivní pouze v případě, kdy jeden klient ovládá prostředky. Ostatní klienti by měli být ideálně v režimu view. Ten nabízí stejný obraz serveru bez možnosti jeho ovládání.

Komunikace standardně probíhá na portech 5900- 5906. VNC klient se připojuje defaultně k portu 5900. Při změně konfigurace klienta i serveru lze použít připojení pomocí webového prohlížeče s podporou Javy. Tento klient komunikuje na portech 5800-5806.

3.3 Zabezpečení VNC

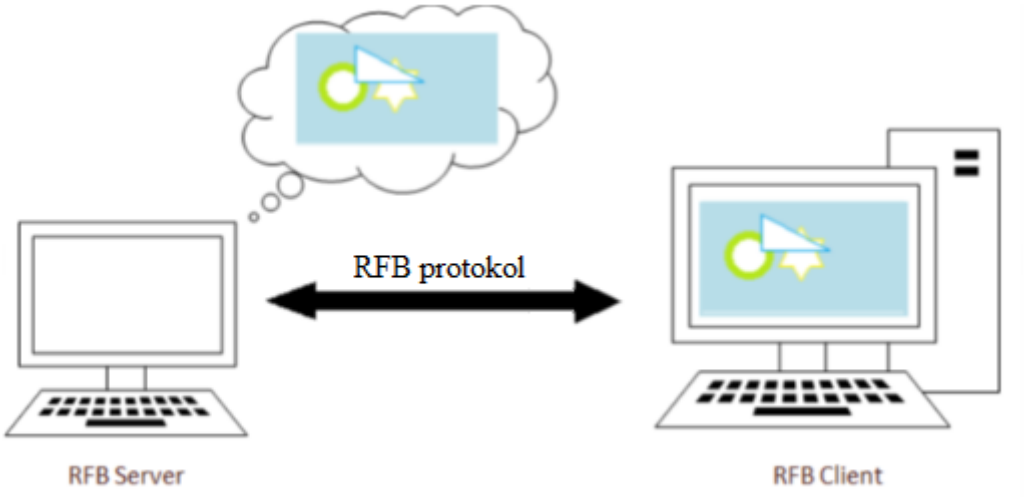
Před zahájením komunikace (je-li tato fáze nastavena) musí klient projít autentizací. Autentizace je zabezpečená pomocí systému challenge-response. Tento typ zabezpečení je založen na základě náhodné výzvy a ověřované odpovědi. Při shodě informací dojde k povolení přístupu a navázání spojení se serverem.

Po kladně vyřízené autentizaci dochází k samotné komunikaci, která standardně není zabezpečená. Síťový provoz není šifrován a může dojít k jeho odposlouchávání třetí stranou. Ta pak může snadno zachytit nejen obraz serveru, ale i příkazy ze strany klienta. Hrozbu odposlechu lze snadno odstranit pomocí VPN či zabezpečeného SSH tunelu. Ten

pak umožní nezabezpečené relaci VNC komunikovat skrz zabezpečený tunel. (VAŠEK, 2009)

3.4 Protokol RFB

RFB je jednoduchý protokol pro vzdálenou správu s grafickým uživatelským rozhraním. Vzhledem k jeho činnosti na úrovni framebufferu je možné ho využít na každém grafickém a okenním operačním systému. RFB protokol je velice často zastoupen ve VNC aplikacích. (RICHARDSON, 2010)



Obrázek 13 – Komunikace RFB protkolu

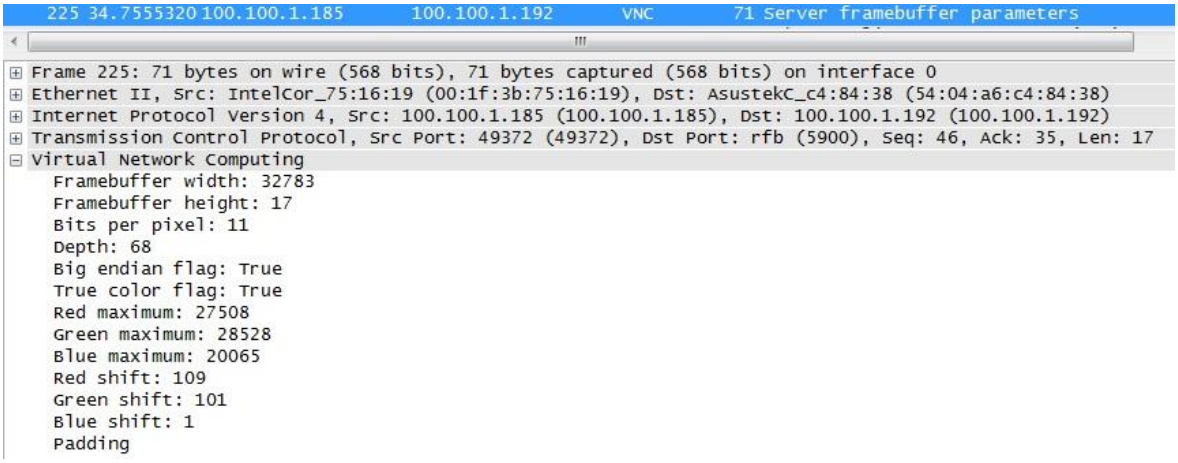
3.4.1 Architektura protokolu

RFB protokol je tenký klient. V protokolu je kladen důraz na velmi malé požadavky ze strany klienta. Tím je zajištěna funkčnost na velkém množství hardwaru.

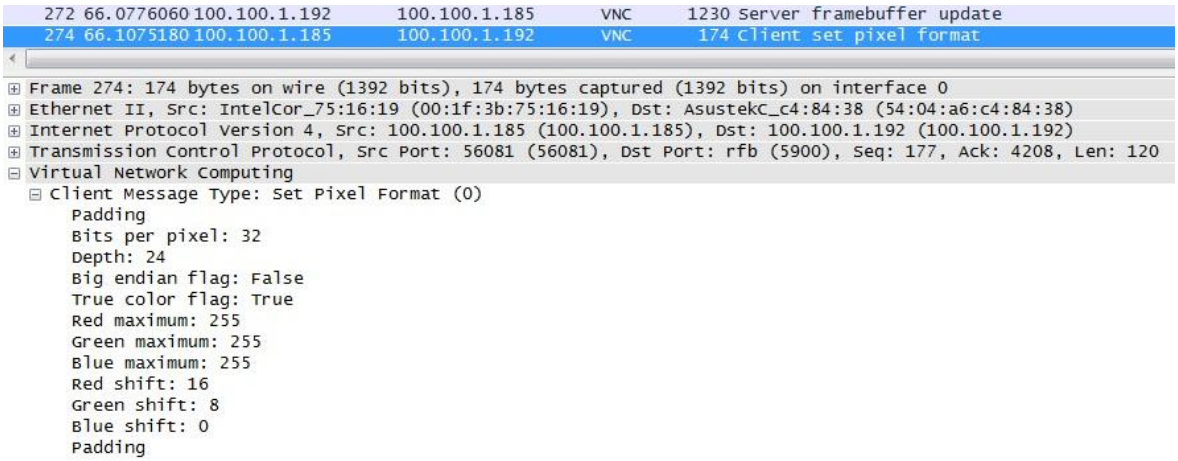
Protokol umožňuje klientovi stát bez příslušnosti. Pokud dojde k odpojení od serveru a jeho opětovnému připojení ke stejnému serveru, stav uživatelského rozhraní zůstane zachován.

Jak již bylo uvedeno, framebuffer je důležitou součástí protokolu. Díky framebufferu je RFB protokol možné využít na mnoha systémech jako je Windows, Mac OS ale i na dalších systémech, které jsou postaveny na grafickém režimu správy oken.

Framebuffer je složen z více částí neboli bufferů. Buffer uchovává informace o barevné hloubce pro každý pixel. Hodnoty jsou rozdělený podle počtu barev. Dále se uchovává alfa kanál, který se používá ve spojitosti s průhledností daného pixelu. Pomocí barevné hloubky, velikosti palety a rozlišení výstupního signálu lze vypočítat množství paměti potřebné pro framebuffer.



Obrázek 14 – Framebuffer - parametry serveru

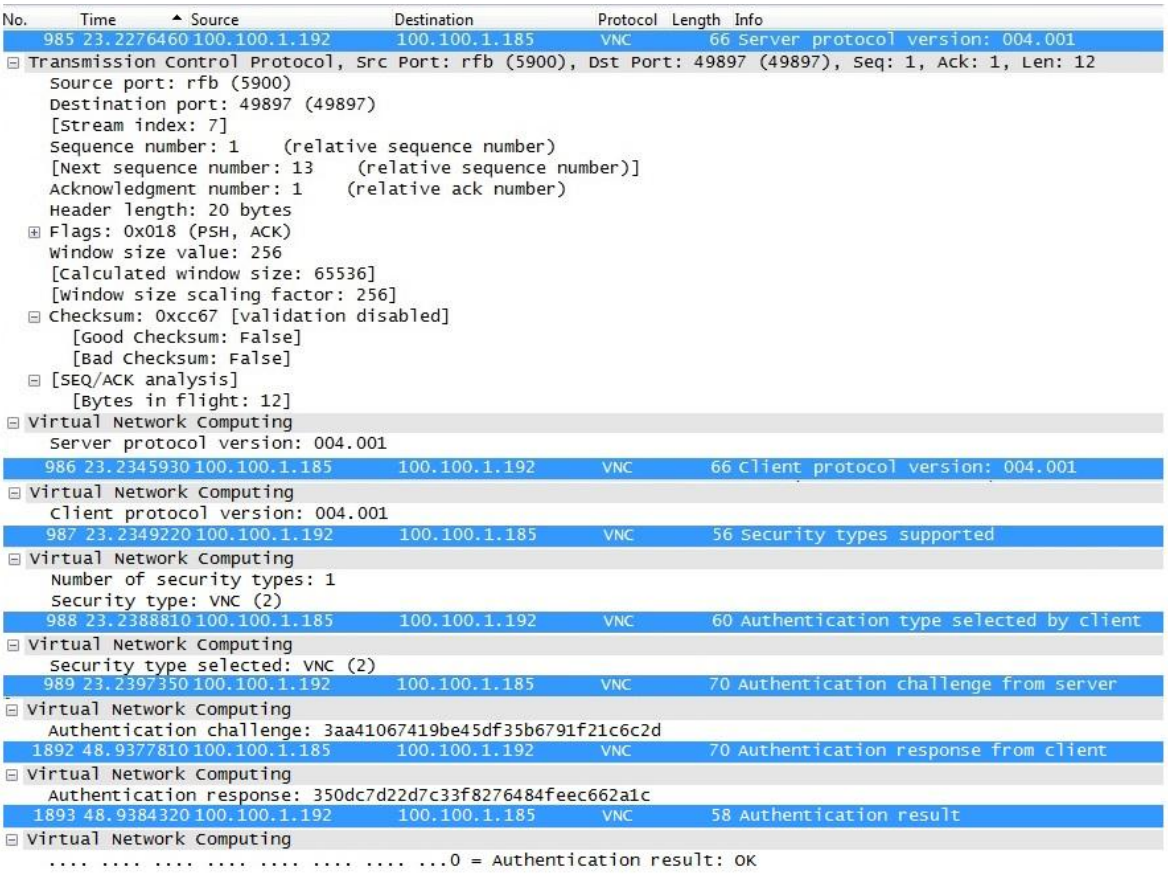


Obrázek 15 – RFB - nastavení pixelu

Navázání komunikace klienta se serverem je rozděleno do tří fází.

V první fázi dochází k synchronizaci verze protokolu a typu zabezpečení. Server navrhne nejvyšší možnou verzi protokolu. Po přijetí zprávy klient odpoví zvolenou verzí protokolu. Zabezpečení nevyžaduje autentizaci, při níž se používá šifrování DES.

V druhé fázi probíhá inicializace, ve které dochází k výběru přístupové formy mezi klientem a serverem. Forma přístupu je rozdělena na sdílený a výlučný přístup k serveru.



Obrázek 16 – Navázání spojení VNC

Třetí fáze slouží k běžné komunikaci mezi klientem a serverem. Zde server odpovídá na obdržená vstupní data odesláním obrazových dat ke klientu. Kódovací metody ZRLE, RAW a další, zmenšují vytížení přenosové linky. (RICHARDSON, 2010)

3.5 Implementace VNC

VNC, jak již bylo uvedeno v úvodu této kapitoly, má mnoho implementací. Příkladem může být RealVNC, UltraVNC nebo TightVNC. Nastavení klientských i serverových částí je podobné. Všechny tři zmíněné VNC deriváty umožňují nastavení hesel pro autentizaci administrátora resp. klienta v režimu view. Dále je možné nastavení portů a kvality přenosu dat. Všechny implementace nabízejí i Java klienta. Ten bývá zaveden pro větší možnost implementace. V případě UltraVNC Java klient řeší zároveň kompatibilitu s operačním systémem Linux. Výše zmíněné VNC deriváty se liší převážně v množství poskytovaných služeb, zabezpečení přenosu, podpory pro operační systémy a mobilní telefony.

3.5.1 RealVNC

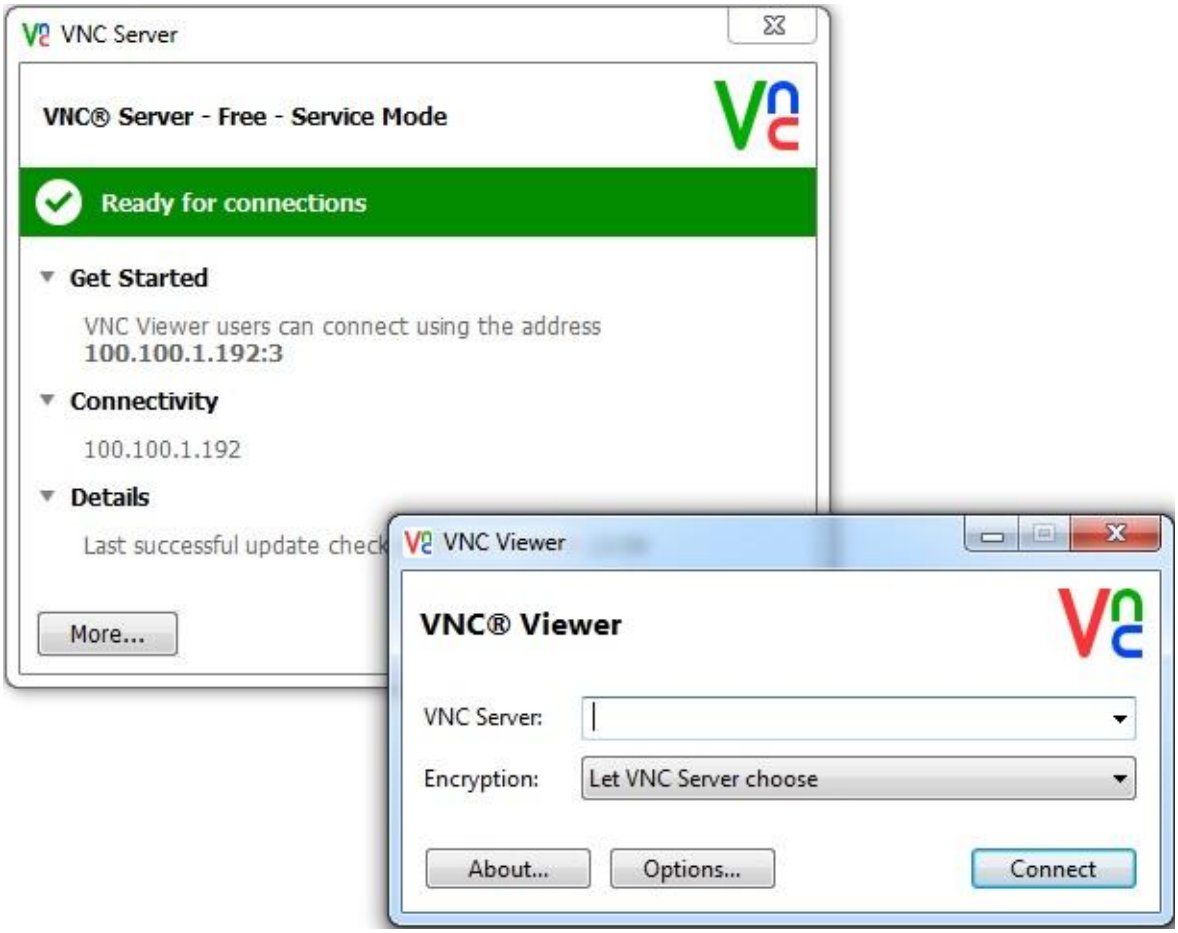
RealVNC byl založen v roce 2002 z původních vývojářů VNC. RealVNC jako jediný z výše uvedených softwarů spadá do kategorie komerčních. Je založen na jednoduché peer-to-peer architektuře. Díky tomu není potřeba žádný další síťový prvek nebo centrální server. Velkou výhodou tohoto softwaru je podpora operačních systémů. Instalaci lze

provést na počítačích s operačním systémem Windows, UNIX, Linux a MAC. Klientský software lze provozovat i na mobilních platformách Windows Mobile, Android a iOS.

Software je rozdělen do tří licenčních skupin: Free, Personal a Enterprise. Bezplatná Free verze poskytuje pouze autentizaci a standardní funkce jako je nastavení kvality přenosu a změny portu. Ostatní rozšiřující funkce jsou placené. Rozdělení licencí a poskytovaných služeb je zobrazeno v tabulce Tabulka 2. (RealVNC, 2002)

Tabulka 2 – RealVNC, rozdělení licencí

Vlastnost / licence	Free	Personal	Enterprise
Multiplatformní vzdálená správa	ANO	ANO	ANO
VNC autentizace	ANO	ANO	ANO
Osobní využití	ANO	ANO	ANO
Komerční využití	NE	ANO	ANO
128bit AES šifrování	NE	ANO	ANO
256bit AES šifrování	NE	NE	ANO
Autentizace systému	NE	ANO	ANO
Optimalizace výkonu	NE	ANO	ANO
Tisk	NE	ANO	ANO
Přenos soborů	NE	ANO	ANO
Chat	NE	ANO	ANO
Vyhrazený kanál pro podporu	NE	ANO	ANO



Obrázek 17 – RealVNC Server a Viewer

```
293719 185.166006 100.100.1.185      100.100.1.192      VNC      1514 Client framebuffer update request

⊞ Frame 293719: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interface 0
⊞ Ethernet II, Src: IntelCor_75:16:19 (00:1f:3b:75:16:19), Dst: AsustekC_c4:84:38 (54:04:a6:c4:84:38)
⊞ Internet Protocol Version 4, Src: 100.100.1.185 (100.100.1.185), Dst: 100.100.1.192 (100.100.1.192)
⊞ Transmission Control Protocol, Src Port: 56880 (56880), Dst Port: 5901 (5901), Seq: 115113735, Ack: 22586549, Len: 1460
⊟ Virtual Network Computing
  ⊟ Client Message Type: Framebuffer Update Request (3)
      Incremental update: True
      X position: 40484
      Y position: 30347
      Width: 33421
      Height: 61166
```

Obrázek 18 – Žádost klienta o aktualizaci framebufferu

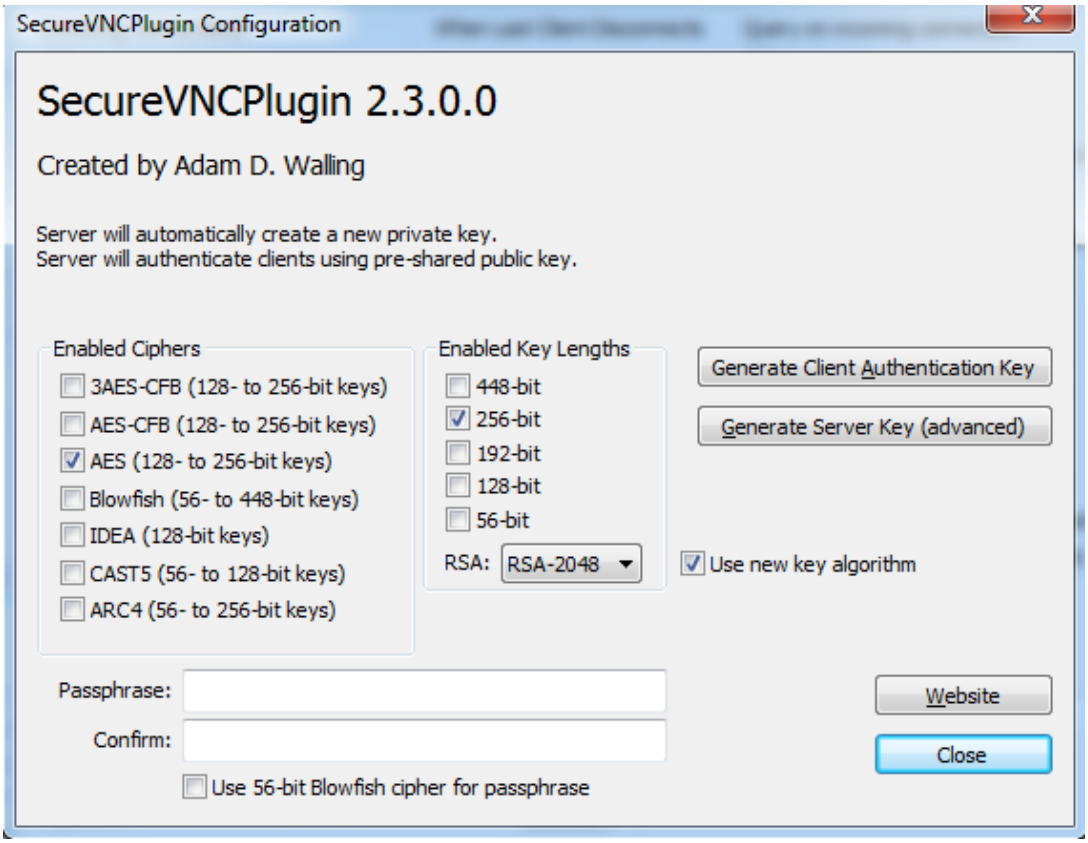
```
293689 185.157855 100.100.1.185      100.100.1.192      VNC      1514 Client pointer event

⊞ Frame 293689: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interface 0
⊞ Ethernet II, Src: IntelCor_75:16:19 (00:1f:3b:75:16:19), Dst: AsustekC_c4:84:38 (54:04:a6:c4:84:38)
⊞ Internet Protocol Version 4, Src: 100.100.1.185 (100.100.1.185), Dst: 100.100.1.192 (100.100.1.192)
⊞ Transmission Control Protocol, Src Port: 56880 (56880), Dst Port: 5901 (5901), Seq: 115107895, Ack: 22586549, Len: 1460
⊟ Virtual Network Computing
  ⊟ Client Message Type: Pointer Event (5)
      .... ...0 = Mouse button #1 position: Not pressed
      .... ..0. = Mouse button #2 position: Not pressed
      .... .1.. = Mouse button #3 position: Pressed
      .... 0... = Mouse button #4 position: Not pressed
      ...0 .... = Mouse button #5 position: Not pressed
      ..0. .... = Mouse button #6 position: Not pressed
      .1.. .... = Mouse button #7 position: Pressed
      0... .... = Mouse button #8 position: Not pressed
      X position: 27375
      Y position: 35525
```

Obrázek 19 – Zachycení pohybu myši klienta

3.5.2 UltraVNC

První verze UltraVNC byla zveřejněna v roce 2005. Jedná se o výkonný, bezplatný a snadno použitelný software. Vychází z původního zdrojového kódu VNC. Spolu s VNC využívá také protokol RFB. Díky tomu je kompatibilní s větší částí aplikací založených na tomto standardu.

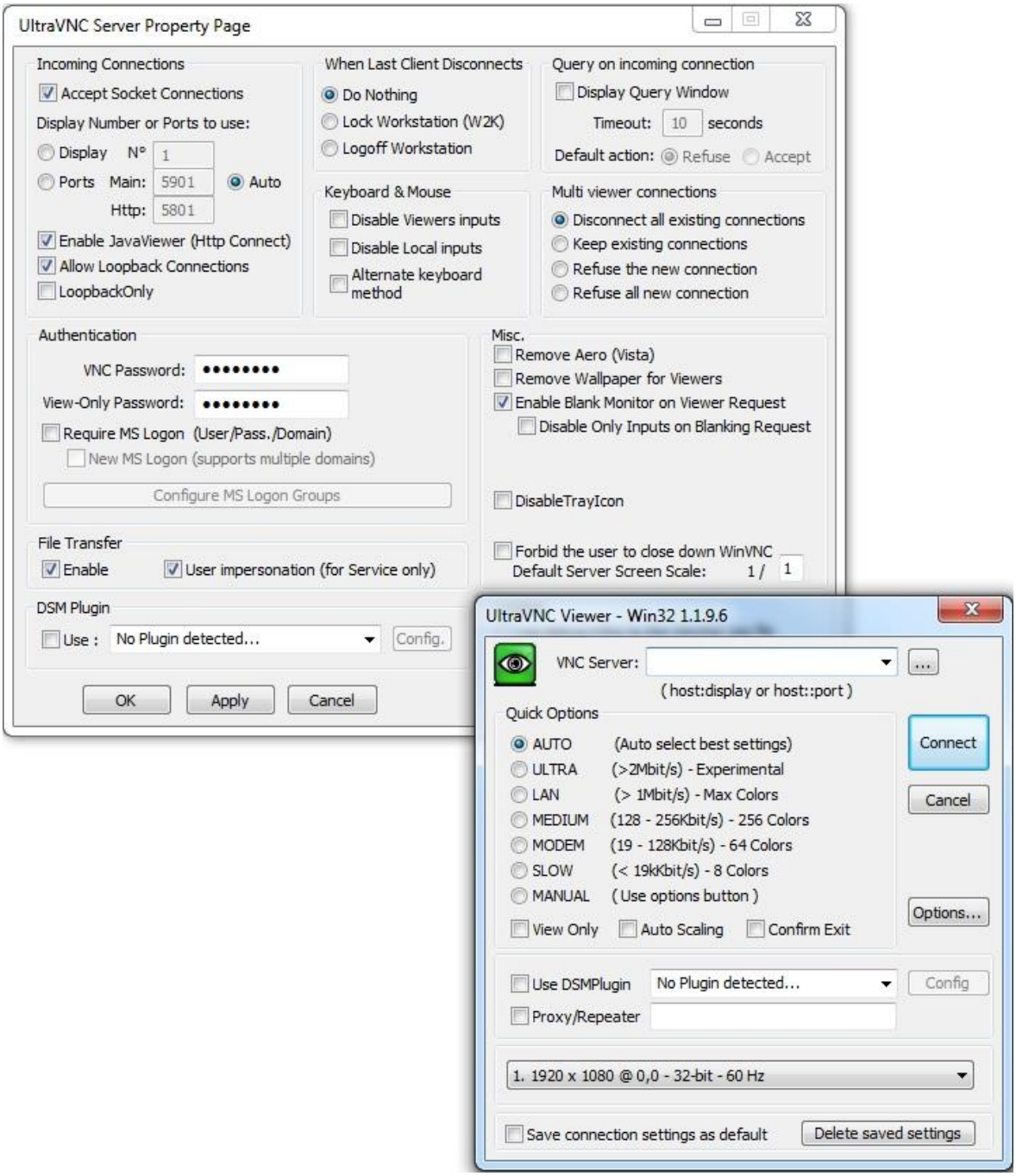
UltraVNC lze implementovat na operační systémy Windows od verze Windows XP. Dále obsahuje Java Viewer, pomocí něhož se lze připojit a zároveň provést přenos dat z jednoduché aplikace na UltraVNC server. Tento Viewer lze využít i na systému Linux a Mac. K šifrování komunikace je možné využít SecureVNCPlugin. Ten nabízí uživateli možnost výběru šifry a délky klíče. Na straně serveru je vygenerován privátní klíč, umístěný na straně klienta a veřejný pro stranu serveru. Pro správnou funkci pluginu musí být zaveden jak na straně serveru, tak i na straně klienta. Plugin nelze využít v případě křížení UltraVNC s jiným VNC derivátem. (UltraVNC, 2008)



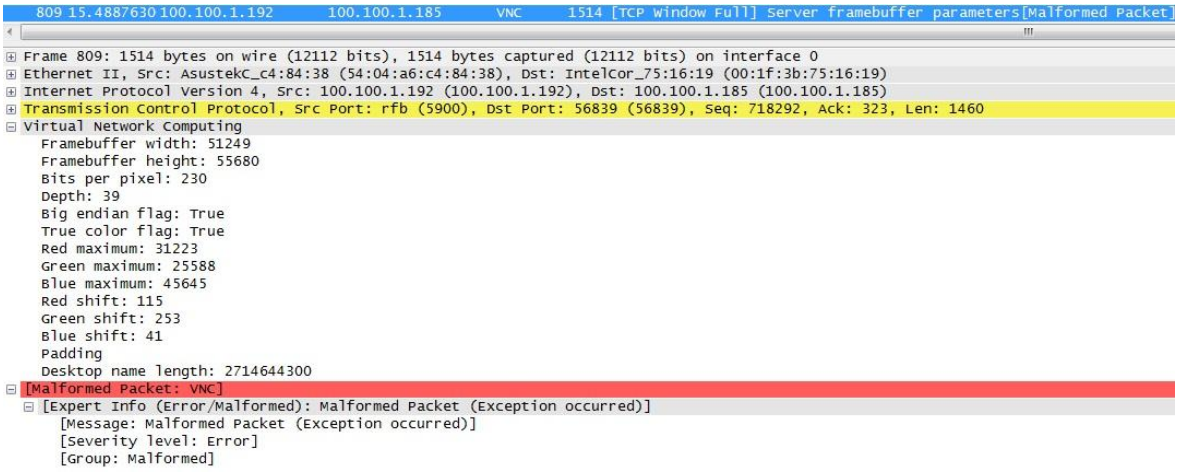
Obrázek 20 – UltraVNC Secure Plugin

UltraVNC oproti původní verzi VNC dále obsahuje:

- nastavitelné DSM šifrování komunikace,
- přenos souborů,
- chatovací nástroj,
- přídavné funkce (odstranění tapety, odstranění Aero u Windows Vista, ...).



Obrázek 21 – UltraVNC Server a Viewer

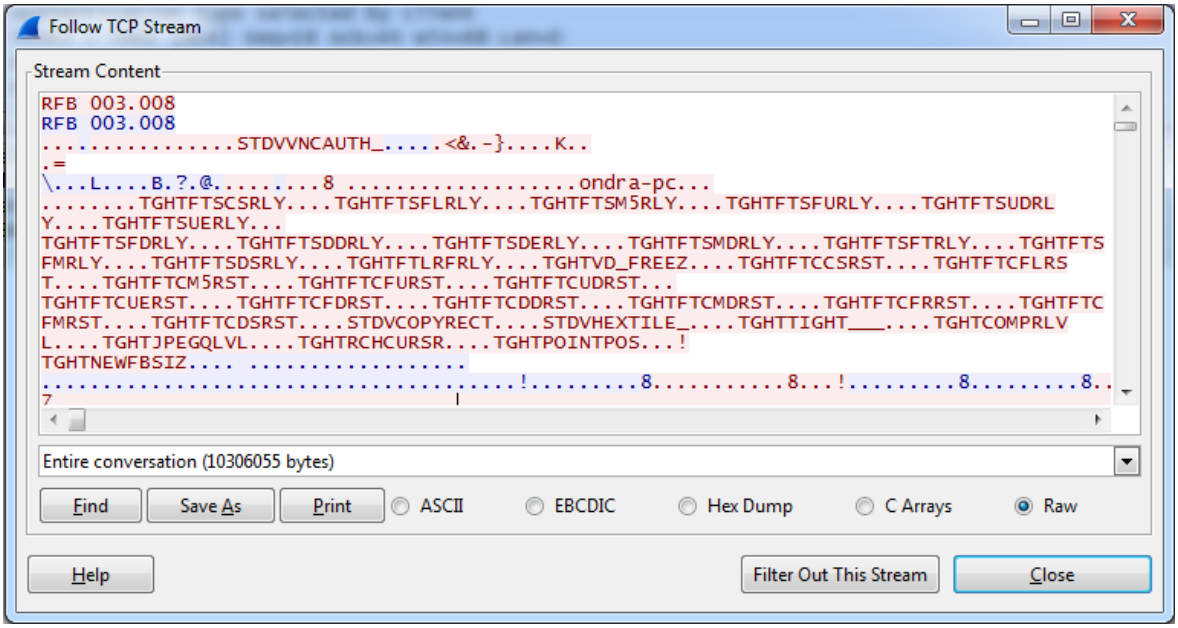


Obrázek 22 – Zachycené parametry framebufferu

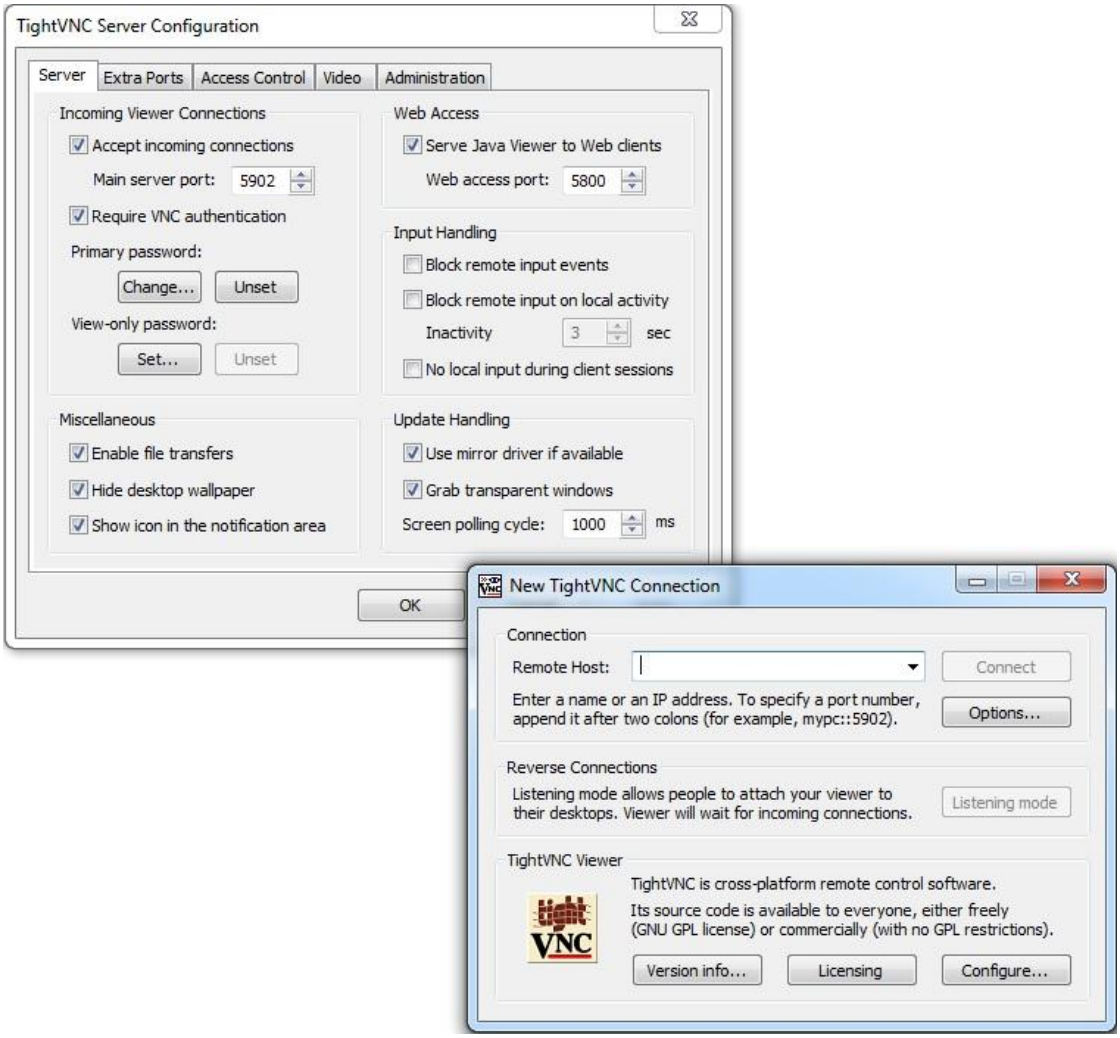
3.5.3 TightVNC

TightVNC stejně jako UltraVNC je bezplatný software s využitím FRB protokolu. Jeho první verze byla zveřejněna roku 2001. Software lze implementovat na operační systémy Windows, Linux a mobilní platformy Windows Mobile, Android a iOS.

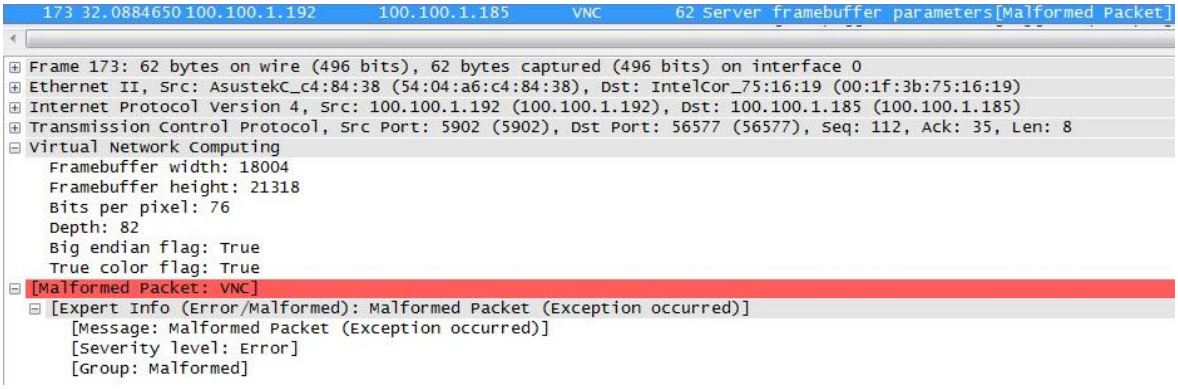
TightVNC poskytuje kromě základních nastavení pouze přenos souborů. Jeho nevýhodou je nedostatečné zabezpečení komunikace pomocí 56bit DES. Tento nedostatek lze odstranit použitím SSH tunelu. (KAPLINSKY, 1999)



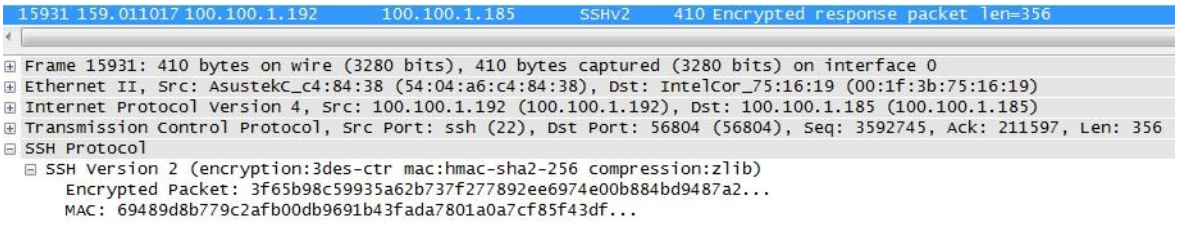
Obrázek 23 – TightVNC TCP Stream



Obrázek 24 – TightVNC Server a Viewer



Obrázek 25 – Nešifrované spojení, zachycený framebuffer



Obrázek 26 – Šifrované spojení pomocí SSH tunelu

3.5.4 Porovnání VNC derivátů

VNC deriváty lze porovnat z pohledu funkce a ceny. Z pohledu funkce je RealVNC nejstabilnější a ve své komerční verzi Enterprise poskytuje i nejvíce služeb společně se zabezpečením přenosu. K RealVNC se počtem funkcí přibližuje pouze UltraVNC.

Z pohledu ceny v bezplatné verzi Free nabízí RealVNC naopak nejméně funkcí a zabezpečená je pouze autentizace. Lepší variantou je TightVNC, který umožňuje navíc přenos souborů, Java klienta i podporu mobilních platforem. Jejich společnou nevýhodou je nešifrovaný nebo slabě šifrovaný přenos. Ten je nutné následně zabezpečit pomocí tunelu. Druhou variantou je UltraVNC, který kromě dalších služeb obsahuje i SecurePlugin. Ten šifruje přenos a odstraňuje potřebu tunelu. Z bezplatných verzí obsahuje nejvíce služeb. Jeho nevýhodou je slabá podpora operačních systému. Mobilní platformy nejsou u UltraVNC podporovány vůbec.

I přes značné výhody mají všechny zmíněné software jednu společnou nevýhodu. Tou je absence přenosu audia. Protokol RFB na přenos audia nebyl zkonstruován. Porovnávané vlastnosti jsou zobrazeny v tabulce Tabulka 3.

Tabulka 3 – Porovnání VNC programů

Vlastnost/ aplikace	RealVNC (Free)	TightVNC	UltraVNC	RealVNC (Enterprise)
Podpora OS	Windows, Linux, Mac OS	Windows, Linux	Windows	Windows, Linux, Mac OS
Zabezpečený přenos	Ano	Ne	Ano	Ano
Přenos souborů	Ne	Ano	Ano	Ano
Tisk	Ne	Ano	Ano	Ano
Chat	Ne	Ano	Ano	Ano
Přenos audia	Ne	Ne	Ne	Ne
Mobilní platformy	Windows Mobile, Android, iOS	Windows Mobile, Android, iOS	Ne	Windows Mobile, Android, iOS

4 Představení RDP

RDP je síťový protokol, vyvinutý firmou Microsoft. Protokol slouží k vzdálenému připojení a správě počítače pomocí grafického rozhraní. Protokol byl poprvé implementován ve Windows NT 4.0 Terminal Server Edition. Protokol je sestaven na základě ITU řady protokolů T.120. Standard T.120 je složen ze sady komunikačních protokolů aplikační vrstvy, které umožňují real-time služby, vícebodové přenosy multimediálních dat a konference. Protokol pracuje na principu architektury klient-server. Kompletní architektura protokolu je zveřejněna na webových stránkách firmy Microsoft.

4.1 Architektura protokolu

4.1.1 RDP spojení

Na počátku RDP spojení je tzv. připojovací sekvence, která slouží k upřesnění společného nastavení po celou dobu spojení mezi klientem a serverem. Připojovací sekvence lze rozdělit na deset odlišných fází.

V první fázi dochází k inicializaci spojení. Klient inicializuje spojení odesláním *Connection Request PDU X.224* na server. Ten odpoví potvrzením spojení pomocí *Connection CornfirmPDU X.224*. Od tohoto okamžiku jsou všechna přenášená data zabalena v *X.224 PDU*.

```
⊞ Frame 92: 73 bytes on wire (584 bits), 73 bytes captured (584 bits) on interface 0
⊞ Ethernet II, Src: IntelCor_75:16:19 (00:1f:3b:75:16:19), Dst: AsustekC_c4:84:38 (54:04:a6:c4:84:38)
⊞ Internet Protocol Version 4, Src: 100.100.1.185 (100.100.1.185), Dst: 100.100.1.192 (100.100.1.192)
⊞ Transmission Control Protocol, Src Port: 52751 (52751), Dst Port: ms-wbt-server (3389), Seq: 1, Ack: 1, Len: 19
⊞ TPKT, Version: 3, Length: 19
⊟ ISO 8073/X.224 COTP Connection-Oriented Transport Protocol
    Length: 14
    PDU Type: CR Connect Request (0x0e)
    Destination reference: 0x0000
    Source reference: 0x0000
    0000 .... = Class: 0
    .... ..0. = Extended formats: False
    .... ...0 = No explicit flow control: False
    Parameter code:   0x01 (Unknown)
    Parameter length: 0
    Parameter value: <not shown>
    Parameter code:   0x08 (ATN extended checksum - 32 bit)
```

Obrázek 27 – RDP X.224 Connect Request

```
93 15.5803600 100.100.1.192    100.100.1.185    COTP    73 CC TPDU src-ref: 0x1234 dst-ref: 0x0000

⊞ Frame 93: 73 bytes on wire (584 bits), 73 bytes captured (584 bits) on interface 0
⊞ Ethernet II, Src: AsustekC_c4:84:38 (54:04:a6:c4:84:38), Dst: IntelCor_75:16:19 (00:1f:3b:75:16:19)
⊞ Internet Protocol Version 4, Src: 100.100.1.192 (100.100.1.192), Dst: 100.100.1.185 (100.100.1.185)
⊞ Transmission Control Protocol, Src Port: ms-wbt-server (3389), Dst Port: 52751 (52751), Seq: 1, Ack: 20, Len: 19
⊟ TPKT, Version: 3, Length: 19
    Version: 3
    Reserved: 0
    Length: 19
⊟ ISO 8073/X.224 COTP Connection-Oriented Transport Protocol
    Length: 14
    PDU Type: CC Connect Confirm (0x0d)
    Destination reference: 0x0000
    Source reference: 0x1234
    0000 .... = Class: 0
    .... ..0. = Extended formats: False
    .... ...0 = No explicit flow control: False
    Parameter code:   0x02 (Unknown)
    Parameter length: 1
```

Obrázek 28 – RDP X.224 Connect Confirm

V druhé fázi dochází k základnímu nastavení komunikace. Základní nastavení mezi klientem a serverem je vyměněno pomocí *MCS Connect Initial PDU* a *MCS Connect Response PDU*. *Connect Initial PDU* a *Connect Response PDU* využívají GCC pakety. Tyto pakety obsahují bloky dat, které byly načteny klientem a serverem.

Třetí fáze slouží k nastavení kanálů. Klient odešle *MCS Erect Domain Request PDU,* následovaný *MCS Attach User Request PDU*. Ten slouží k připojení primární uživatelské identity k MCS doméně. Server odpoví *MCS Attach User Confirm PDU*, který obsahuje identifikační číslo uživatelského kanálu. Klient pokračuje v připojení uživatelského kanálu. Vstupně/výstupní kanál a všechny další statické virtuální kanály žádají o připojení pomocí *MCS Channel Join Request PDU*. Identifikační čísla kanálů jsou získány z dat obsažených v GCC paketu. Server potvrdí připojení každého kanálu pomocí *MCS Channel Join Confirm PDU.* RDP klienti verze 4.0-8.0 odesílají na server Channel Join Request až po potvrzení předchozí žádosti. Klient 8.1 odesílá všechny požadavky v jedné dávce. Tím je minimalizován časový úsek připojení. Od této chvíle jsou všechny data odeslané z klienta na server zabalena v *MCS Send Data Request PDU,* zatímco data odeslaná ze serveru jsou zabalena v *MCS Send data Indication PDU*. Ten je navíc zabalen prostřednictvím *X.224 PDU.*

Čtvrtá fáze spojení slouží k zabezpečení. Jsou-li standardní bezpečnostní mechanismy RDP protokolu zapnuty, klient odešle na server *Security Exchange PDU.* Ten obsahuje šifrované 32B náhodné číslo. Toto číslo je šifrováno pomocí veřejného klíče serveru. Veřejný klíč serveru stejně jako 32B náhodné číslo jsou získány z paketu *GCC Conference Creacte Response.* Klient a Server poté využijí dvě náhodná 32B čísla ke generování klíčů relace. Ty jsou využity k šifrování a ověření integrity následného RDP provozu. Od této chvíle může být RDP komunikace šifrována. *X.224* a *MCS* záhlaví poté označují, zda jsou připojené data šifrovány. Komunikace server-klient nemusí být vždy šifrována, zatímco klient-server komunikace musí být šifrována vždy.

Pátá fáze připojení slouží k výměně bezpečnostních mechanismů. Bezpečností data klienta (např. uživatelské jméno, heslo, automatická volba cookie) jsou odeslána na server pomocí *Client Info PDU.*

Šestá fáze připojení je určena k autodetekci spojení. Cílem je určit charakteristiku sítě, jako je například latence a šířka pásma mezi klientem a serverem. Charakteristika je určena výměnami paketů v předem určeném intervalu.

Sedmá fáze spojení slouží ke kontrole licencí. Cílem je přenést licence ze serveru ke klientu. Klient tyto licence ukládá a při následném spojení odešle klíč na server k ověření. V některých případech klient nemá povolení k uložení licence. Ve skutečnosti pakety vyměňované v této fázi protokolu závisejí na licenčních mechanismech serveru.

Osmá fáze spojení slouží k multitransportu. Po připojení je zajištěno dokončení licenční fáze. Poté může server zahájit multitransportní spojení odesláním *Multistransport Request*

PDU paketu ke klientu. Multitransportní spojení využívá zprávy z RDP-UDP, TLS, DTLS a multitransportních protokolů.

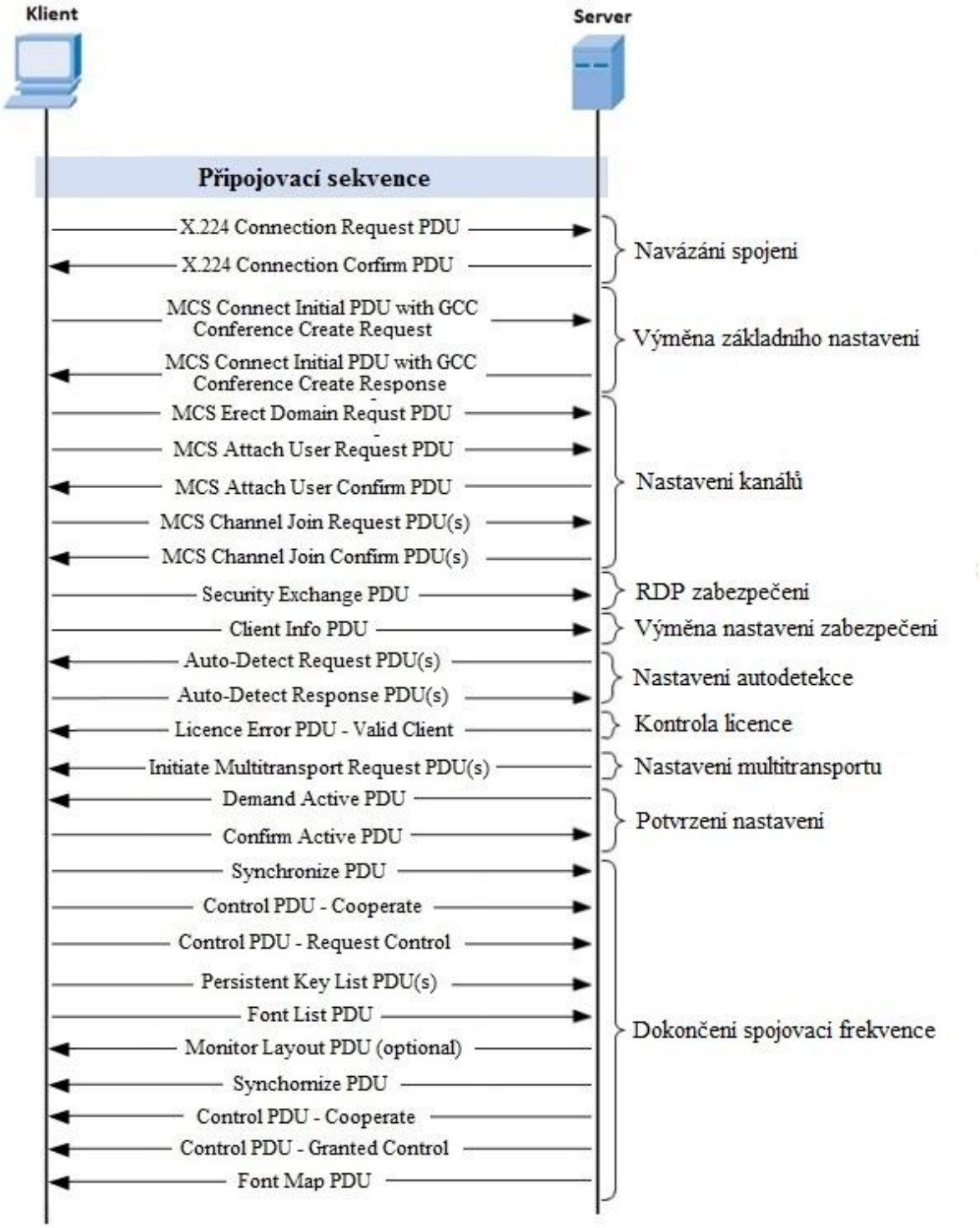
Devátá fáze připojení slouží k potvrzení předešlého nastavení. Server odešle na klienta *Demand Active PDU*, který obsahuje podporované vlastnosti serveru. Ty jsou potvrzeny klientem pomocí *Confirm Active PDU*.

Desátá a poslední fáze spojení je určena k dokončení spojovací frekvence. Klient a server si vymění pakety ke shrnutí detailů spojení. Pakety zaslané z klienta na server v průběhu této fáze nejsou závislé na některém z paketů, zaslaných ze serveru ke klientu. Pakety mohou být zaslány v jedné dávce, ale pouze za předpokladu, že je dodrženo toto pořadí:

- *Client Synchronize PDU* je zaslán po předání *Confirm Active PDU.*
- *Client Control PDU* je zaslán po předání *Client Synchronize PDU.*
- *Client Control (Request Control) PDU* je odeslán po předání *Client Control (Cooperate)* PDU.
- *Persistent Key List PDUs* jsou odeslány po předání *Client Control (Request Control) PDU.*
- *Font List PDU* je odeslán po předání *Persistent Key List PDUs.*
    - V případě že *Persistent Key List PDUs* nebyly odeslány, je odeslán po předání *Client Control (Request Control) PDU.*
- Volitelný *Monitor Layout PDU* není závislý na žádném PDU a je zaslán po *Demand Active PDU.*
- *Server Synchornize PDU* je zaslán jako odpověď na *Confirm Active PDU.*
- *Server Control (Cooperate) PDU* je zaspán po předání *Server Synchronize PDU.*
- *Server Cointrol (Granted Control) PDU* je zaslán jako odpověď na *Client Control (Request Control) PDU.*
- *Font Map PDU* je zaslán jako odpověď na *Font List PDU.*

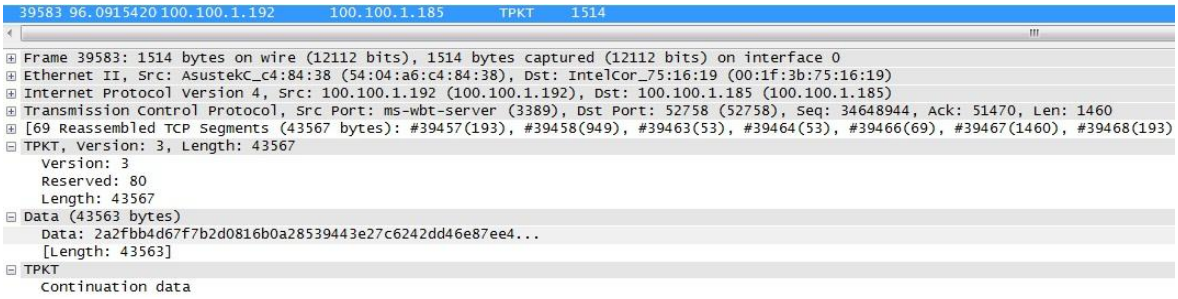
Jakmile klient odešle *Confirm Active PDU*, může začít s komunikací a odesíláním vstupních dat klávesnice a myši na server. Ten po obdržení *Font List PDU* může odesílat grafický výstup klientovi. Kromě vstupních a grafických dat, které mohou být vyměněny mezi klientem a serverem při spojení, obsahuje i informace o řízení a virtuálních kanálech.(CORPORATION, 2013; MICROSOFT, 2007)

Celá připojovací sekvence je zobrazena na obrázku Obrázek 29.

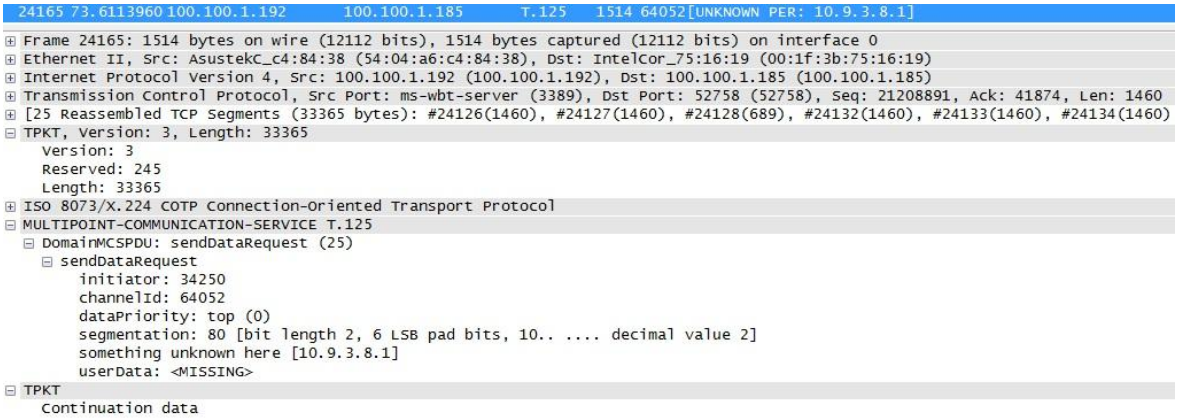


Obrázek 29 – RDP připojovací sekvence (CORPORATION, 2013)

Protokol RDP standardně využívá transportní protokol TPKT. TPKT pracuje na vrcholu TCP. Ke komunikaci se standardně využívá port 3389.



Obrázek 30 – TPKT protokol



Obrázek 31 – Protokol T.125

4.1.2 Statické virtuální kanály

Statické virtuální kanály slouží k přidání funkcí, které nejsou přímo obsaženy v RDP protokolu. Ve vzniklém spojení jeden kanál využívají obrazová data společně se vstupními zařízeními jako je například myš nebo klávesnice. Ostatní kanály jsou volné pro přenos funkcí. Každý virtuální kanál působí jako nezávislý datový tok. Kanály umožňují bezeztrátovou komunikaci mezi klientem a serverem přes hlavní RDP spojení. V rámci spojení může být vytvořeno až 31 virtuálních kanálů. (CORPORATION, 2013)

4.1.3 Přenos a komprese dat

Protokol RDP využívá kompresi dat přenášených pomocí virtuálních kanálů a PDU paketů zaslaných ze serveru ke klientovi. Jedna verze kompresoru (RDP 4.0) je založena na MPPC protokolu a využívá 8kB vyrovnávací paměť. Pokročilejší verze kompresoru (RDP 5.0) je odvozena z předchozí verze 4.0 a využívá 64kB vyrovnávací paměť upravené o Huffmanova kódovací pravidla. RDP dále implementuje RLE ke kompresi bitmapových dat odeslaných ze serveru ke klientovi. Všichni klienti musí být schopni dekomprese bitmapových dat. (CORPORATION, 2013)

4.1.4 Zabezpečení protokolu

Standardní zabezpečení protokolu RDP podporuje čtyři úrovně šifrování:

- Nízká – všechna data odeslaná z klienta na server jsou šifrována maximální ochranou klíče podporovaného klientem.

- Kompatibilní s klientem – všechna data zasílaná mezi klientem a serverem jsou šifrována maximální ochranou klíče podporovaného klientem.
- Vysoká – všechna data zasílaná mezi klientem a serverem jsou šifrována maximální ochranou klíče podporovaného serverem.
- Kompatibilní s FIPS – všechna data zasílaná mezi klientem a serverem jsou chráněna pomocí šifrovací metody FIPS.

Požadovaná úroveň zabezpečení je nastavena na serveru. (CORPORATION, 2013)

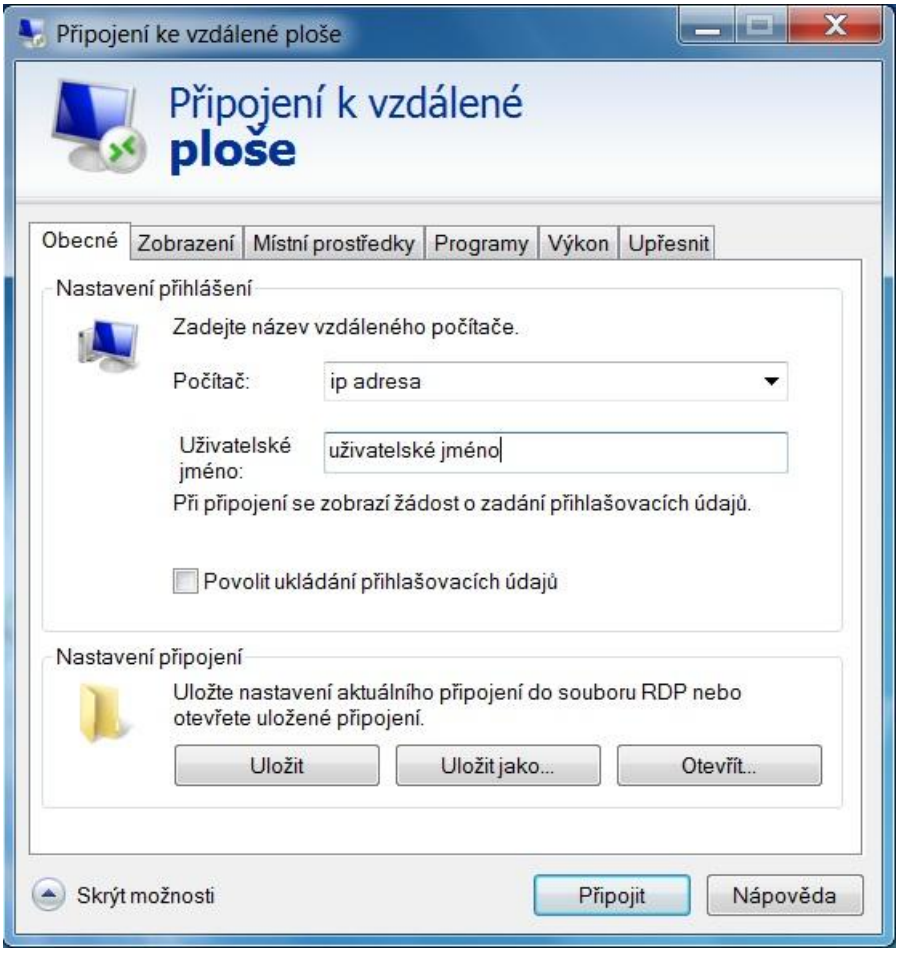
4.2 Implementace protokolu

RDP protokol jakožto produkt společnosti Microsoft je implementován ve většině operačních systému od verze Windows 95 a serverové edice Windows NT. Ačkoli jde o produkt Microsoftu, klienti pro připojení k serveru vzdálené plochy existují na více operačních systémech. Existují i na mobilních platformách jako jsou iOS a Android. Serverové části se nenachází pouze u operačních systémů Microsoft. Existují řešení i pro Unixové systémy.

4.2.1 Vzdálená plocha - Vzdálená pomoc

Microsoft Windows je "jednouživatelský" operační systém. To v praxi znamená, že na jedné pracovní stanici může pracovat pouze jeden uživatel. Výjimku tvoří serverové edice Windows. Ty umožňují připojení a využití jednoho serveru více uživateli.

Vzdálená plocha s výjimkou serverových systémů respektuje Windows jako jednouživatelský systém. To způsobuje odhlášení uživatele na serveru po připojení uživatele z pozice klienta. Uživatel na straně serveru je odhlášen a nevidí aktuální činnost na počítači. Po opětovném přihlášení uživatele na straně serveru je vzniklé připojení přerušeno. Klient obdrží zprávu o přihlášení jiného uživatele. Prostřednictvím vzdálené plochy je možné využít vzdáleného tisku nebo vzdálené schránky. Jako server vzdálené plochy nelze využít jakákoliv edice Windows.



Obrázek 32 – RDP vzdálené plocha

Vzdálené pomoc je opakem vzdálené plochy. Zde klient a server sdílí stejnou obrazovku. To umožňuje komunikaci mezi klientem a serverem v reálném čase. Vzdálená pomoc dále umožňuje sledovací, ale i ovládací mód. V případě ovládacího módu mohou uživatelé pracovat zároveň. Zahájení komunikace má na starost server, který pošle tzv. Pozvání. Vzdálená pomoc podporuje mimo klasické textové komunikace i hlasovou společně s výměnou souborů. Serverem může být počítač s libovolnou edicí systému Windows 7.

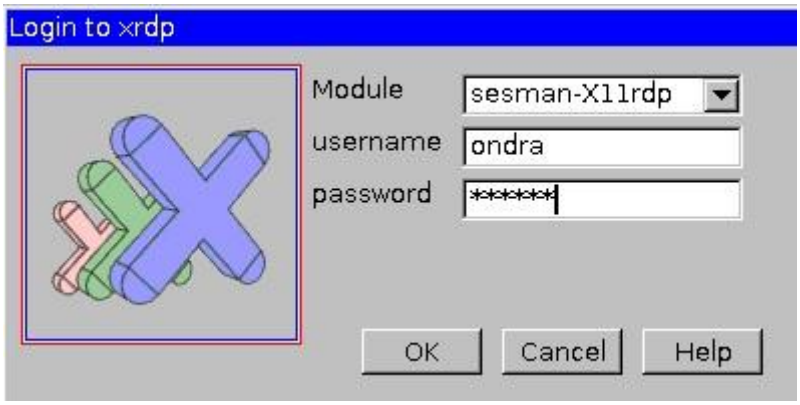
Vzdálená pomoc i Vzdálená plocha je ve výchozím nastavení vypnuta. Připojení je možné jak z lokální tak i veřejné sítě. K připojení z jiné než lokální sítě je nutná veřejná IP adresa nebo správně nastavené porty routeru. Ideálním řešením je využití VPN.

4.2.2 Terminal Services

Terminal services neboli serverová část RDP protokolu je součástí Windows od edice NT 4.0 Terminal Server Edition až po současné systémy Windows Server 2012 R2 a Windows 8.

V linuxovém systému je nejznámější implementací serverové části RDP protokolu xrdp. Xrdp poskytuje plně funkční terminálový server na operačním systém Linux. V praktické části byl využit operační systém Ubuntu 14.04. Ten je schopen navázat spojení s rdesktop,

freerdp a klientem vzdálené plochy společnosti Microsoft. Xrdp je bezplatný software šířený pod licencí GPL.



Obrázek 33 – Xrdp server

4.2.3 RDP klienti

RDP klienti slouží pro připojení k serveru. RDP klient je součástí operačního systému Windows od edice 95 až po dnešní moderní Windows 8.1. Stejně jako serverová část RDP protokolu i klientská část existuje pro více operačních systémů včetně podpory pro mobilní platformy. Mezi podporované operační systémy patří Linux, Unix, Android, OS X, iOS a další operační systémy.

V linuxovém systému systému mezi nejznámější RDP klienty patří FreeRDP. FreeRDP je bezplatná implementace RDP protokolu vydaná pod licencí Apache. FreeRDP vznikl v roce 2009 z rdesktop a implementací nových funkcí. FreeRDP obsahuje klientskou, ale i serverovou část. Implementuje mnoho funkcí z protokolu RDP 8.0 mezi které patří:

- přesměrování zvuku,
- přesměrování tisku,
- přesměrování portů,
- RemoteFX,
- RemoteApp,
- přesměrování USB,
- autentizace na úrovni sítě,
- přesměrování clipboard.

FreeRDP je dostupný pro operační systémy Anroid, Mac OS, Linux, Windows a iOS. V praktické části byl využit klient na operačním systému Ubuntu 14.04.

4.2.4 Verze protokolu

První verzí protokolu RDP je verze 4.0, která byla představena v systému Windows NT 4.0 Server. Edice NT 4.0 využívá technologii Citrix MultiWin. Hlavní myšlenkou této technologie byla podpora pro připojení více uživatelských relací zároveň.

Verze 5.0 byla představena v systému Windows 2000 Server. Obsahuje podporu tisku v lokální síti. Současně je zaměřena na lepší využití šířky pásma.

Verze 5.1 byla představena v systému Windows XP Professional. Obsahuje podporu 24 bitové barvy a zvuku. Klient je k dispozici pro systémy Windows 95, 98, 2000 a Windows NT 4.0.

Verze 5.2 byla představena v systému Windows Server 2003. Zároveň je integrována do systému Windows XP Professional x64. Obsahuje podporu pro místní mapování zdrojů, adresářové relace a možnost připojení v konzolovém módu. Pro ověření autentizace serveru a zašifrování komunikace využívá TLS 1.0.

Verze 6.0 byla představena v systému Windows Vista. Obsahuje podporu pro autentizaci na úrovni sítě a Windows Presentation Foundation aplikace. Dále vylepšuje podporu zobrazení pro více monitorů. Klient této verze je dostupný pro operační systém Macintosh X. Je k dispozici s podporou pro Intel a PowerPC Mac OS verze 10.4.9 a novější.

Verze 6.1 je součástí systému Windows Server 2008 a zároveň Windows Vista Service Pack 1. Tato verze zahrnuje nové funkce představené v systému Windows Server 2008. Mezi tyto funkce patří vzdálené připojení k jednotlivým programům. Hlavní přednost této verze je Easy Print Driver. Ten umožňuje přesměrování tisku. Tím je možné tisknout na straně klienta z aplikací běžících na serveru bez nutnosti instalace tiskového ovladače na server.

Verze 7.0 je součástí systému Windows Server R2 a zároveň Windows 7. Mezi nové funkce této verze patří přesměrování Windows Media Playeru, obousměrné audio, podpora Aero glass společně s rychlejší výměnou bitmapových dat. Tím je stabilizován obraz při vytížení sítě či pomalém přenosu. Tato verze jako první podporuje skutečný přenos pro více monitorů. Většina z výše uvedených funkcí k dispozici pouze pro edice systému Windows 7 Enterprise nebo Ultimate.

Verze 7.1 je součástí Windows 7 Service Pack 1 a Windows Server 2008 R2 Service Pack 1. V této verzi byla přidána podpora RemoteFX. (TERMSERV, 2009)

Verze 8.0 je součástí Windows 8 a Windows Server 2012. Aktualizace operačních systému Windows Server 2008 R2 a Windows 7 umožňuje instalaci klienta RDP 8.0 identického pro Windows 8 a Windows Server 2012. Po aktualizaci lze využít všech ovládacích prvků. Mezi nové funkce verze 8.0 patří:

- RemoteFX media streaming - umožňuje přenos medií po různých typech sítí, včetně WAN sítě.
- Remote FX automatická detekce sítě - detekuje připojení a automaticky nastaví vlastnosti rámce.
- RemoteFX pro WAN - optimalizace pro pomalé či latentní připojení.
- RemoteFX adaptivní grafiky - optimalizuje zobrazení v závislosti na vytížení serveru a klienta, využívá progresivní vykreslování.

- RemoteFX přesměrování USB - umožňuje přesměrování USB i do virtuálních desktopů.
- Výkonnostní čítače. (VÝŠEK, 2013)

V této verzi byla odstraněna funkce Aero glass. Windows Server 2012 podporuje připojení k verzi 6.0 a novější.

Verze 8.1 je součástí systému Windows 8.1 a Windows Server 2012 R2. Pro Windows 7 Service Pack 1 existuje aktualizace klienta RDP 8.1. Aktualizace neimplementuje serverové komponenty verze 8.1. Tato verze opravuje vizuální závady.

Nově představené verze RDP protokolu nejsou vždy kompatibilní se staršími a všemi stávajícími verzemi operačního systému Windows.

5 Porovnání VNC a RDP

V této kapitole jsou porovnány moderní protokoly pro vzdálenou správu VNC a RDP.

5.1 Vlastnosti protokolů

Protokoly VNC a RDP mají shodné, ale i rozdílné vlastnosti. Z tohoto hlediska ze porovnat následující:

- Porty
    - Oba protokoly využívají jiný síťový port. RDP běží standardně na portu 3389, zatímco VNC protokol využívá portu 5900. V obou případech je nutné správně natavit router a firewall.
- Síťová architektura
    - RDP i VCN protokol pracují na principu klient-server. Výjimku tvoří software RealVNC, který využívá architektury peer-to-peer. Z pohledu RDP nelze za klasický klient-server považovat službu Vzdálená plocha v operačním systému Windows.
- Zabezpečení protokolu
    - Oba protokoly využívají společné zabezpečení pomocí autentizace. RDP protokol navíc podporuje čtyři úrovně šifrování, které je určeno na straně serveru. Oproti tomu je VNC protokol v základu nezabezpečený. V polovině případů VNC je doporučeno využít SSH tunel.
- Způsob přenosu dat
    - RDP protokol využívá k přenosu objektové řešení, zatímco VNC protokol využívá bitmapové řešení. Obě řešení jsou rovnocenné v případě dostatečné rychlosti sítě. V opačném případě je objektové řešení rychlejší. Oba zmíněné protokoly využívají k přenosům kompresi dat.
- Nároky na systém

    - V operačním systému Windows zatíží více systém protokol VNC. To je způsobeno instalací další služby. V ostatních operačních systémech jsou nároky velice podobné.
- Způsob ovládání
    - Způsob ovládání může být v obou případech odlišný. V případě využití protokolu VNC jsou přenášeny snímky obrazovky. Uživatel jak na straně serveru, tak i na straně klientu vidí shodné snímky. V opačném případě při využití RDP protokolu a Vzdálené plochy nelze sledovat aktuální práci na serveru. (JANIŠ)

5.2 Implementace protokolů

Implementace protokolů se liší v závislosti na operačním systému.

5.2.1 Operační systém Linux

Protokoly VNC i RDP je možné implementovat na operační systém Linux. Vzdálená správa se obvykle provádí přes protokol X11. Ten byl implementován pro grafické aplikace. Většina VNC derivátů obsahuje přímou podporu operačního systému Linux. Případný problém s kompatibilitou systému z pozice klienta řeší Java viewer.

Protokol RDP jako produkt Microsoftu je v operačním systému zastoupen také. RDP server lze zajistit instalací xrdp. Klientskou část RDP obstará například FreeRDP nebo Remmina.

5.2.2 Operační systém Windows

Protokol RDP je součástí operačního systému Windows od edice Windows NT a Windows 95 prostřednictvím služby Vzdálená plocha. Protokol je u většiny edicí součástí jak v podobě klienta tak i serveru.

Protokol VNC osahuje přímou podporu pro operační systém Windows jak v podobě klienta tak i serveru. Oproti RDP není primárně zastoupen a musí být doinstalován.

5.3 Využití protokolů v praxi

Z porovnávaných VNC derivátů je nejstabilnější a zároveň nabízí nejvíce funkcí RealVNC Enterprise. Ten je v praxi možné využít v:

- interní a externí Help desk,
- IT oddělení,
- dálkové demonstrace,
- eLearning,
- využití pro individuální účely.

RealVNC nabízí široké spektrum využití vzdálené správy pro komerční i nekomerční účely. Hlavní nevýhodou tohoto software je absence přenosu zvuku.

Možné využití RDP protokolu je podobné, avšak není stejné. RDP protokol je vhodné využít v lokální síti nebo vytvořené virtuální privátní síti v rámci systému MS Windows, se kterým dobře interaguje. Vzdálená plocha protokolu RDP podporuje funkce, které protokol VNC neumožňuje. Mezi tyto funkce patří podpora zobrazení více monitorů, přesměrování USB portů, využití virtuálních kanálů a několikrát zmiňovaný přenos audia.

Implementace protokolů RDP i VNC mají dvě výhody oproti modernímu programu pro vzdálenou správu TeamViewer. První výhoda vzniká při využití vzdálené správy v lokální síti. RDP a VNC oproti TeamVieweru nemusí komunikovat přes Internet a své servery. Druhou výhodou těchto protokolů jsou lepší vlastnosti pro administrátora. Implementace protokolu RDP stejně jako VNC mohou zablokovat instanci cizího softwaru. V případě TeamVieweru a spuštěné verze QuickSupport je zablokování vzdáleného přístupu velice komplikované.

Závěr

Tato bakalářská práce byla zaměřena na podrobnou analýzu dostupných protokolů pro vzdálenou správu a přístup. Celá komunikace byla zachycena pomocí síťového nástroje Wireshark, který je schopen zachytit kompletní příchozí i odchodí pakety na zkoumaném síťovém rozhraní.

V první kapitole práce byly popsány základní pojmy, spojené se vzdálenou správou. Byl zde nastíněn proces zachycení a analýzy paketů. Dále byly podrobně vysvětleny tunely pro zabezpečení komunikace vzdálené správy mimo lokální síť. Byl zde krátce popsán referenční ISO/OSI model a jeho vrstvy spojené s protokoly RDP, RFB a aplikacemi VNC.

Druhá kapitola práce byla zaměřena na představení VNC. Byla zde popsána historie, vlastnosti a zabezpečení původního protokolu VNC. Dále zde byla podrobně popsána architektura protokolu RFB. Byly zde zmíněny moderní protokoly VNC a jeho varianty. Na závěr této kapitoly byla provedena implementace, zachycení provozu a porovnání nabízených služeb zmíněných protokolů VNC.

Třetí kapitola této práce byla zaměřena na představení protokolu RDP. Byla zde podrobně popsána architektura protokolu společně s verzemi RDP protokolu. Dále zde byly představeny RDP klienti a Terminal Services pro různé operační systémy. Na závěr této kapitoly byla provedena implementace protokolu na dva operační systémy a zachycena jejich komunikace.

Čtvrtá a poslední kapitola této práce byla zaměřena na porovnání protokolu, které slouží ke vzdálené správě. Bylo zde poukázáno na zásadní rozdíly mezi protokoly VNC a RDP. Na závěr této kapitoly byly popsány možnosti nasazení zkoumaných protokolů v praxi.

Literatura

CORPORATION, MICROSOFT. 2005. Configuring authentication and encryption: Terminal Services. [Online] 21. Leden 2005. http://technet.microsoft.com/en-us/library/cc782610.aspx.

—. 2013. Remote Desktop Protocol: Basic Connectivity and Grapgics Remoting. [Online] 18. 1 2013. [Citace: 23. 4 2014.] http://download.microsoft.com/download/9/5/E/95EF66AF-9026-4BB0-A41D-A4F81802D92C/[MS-RDPBCGR].pdf.

HORÁK, Jaroslav a Milan, KERŠLÁGER. 2006. *Počítačové sítě pro začínající správce.* Brno : Computer Press, 2006. ISBN 80-251-0892-9.

JANIŠ, Roman. Porovnání VNC vs. RDP - vzdálená plocha. [Online] [Citace: 17. 4 2014.] http://janis.site88.net/index.php/windows/118-porovnani-vnc-vs-rdp-vzdalena-plocha.html.

KAPLINSKY, Constantin. 1999. TightVNC: VNC-Compatible Free Remote Control / Remote Desktop Software. [Online] TightVNC, 1999. [Citace: 15. 4 2014.] http://www.tightvnc.com/.

KOLEKTIV. 2003. *CCNA 3 and 4 Vyd.1.* Indianapolis : Cisco Press, 2003. ISBN 15-871-3113-7.

LABORATORIES, AT&T. 1999. VNC - Virtual Network Computing from AT&T Laboratories Cambridge. [Online] AT&T Laboratories Cambridge, 1999. [Citace: 14. 4 2014.] http://www.hep.phy.cam.ac.uk/vnc_docs/index.html.

MICROSOFT. Protokoly tunelového propojení VPN. [Online] Microsoft. [Citace: 9. 4 2014.] http://technet.microsoft.com/cs-cz/library/cc771298(v=ws.10).aspx.

—. 2007. Understanding the Remote Desktop Protocol (RDP). [Online] Microsoft, 27. 3 2007. [Citace: 16. 4 2014.] http://support.microsoft.com/kb/186607/en.

RealVNC. 2002. *VNC remote access for desktops.* [Online] 2002. [Citace: 15. 4 2014.] http://realvnc.com/products/vnc/.

RICHARDSON, Tristan. 2010. The RFB Protocol: Version 3.8. [Online] RealVNC Ltd, 26. 11 2010. [Citace: 12. 4 2014.] http://www.realvnc.com/docs/rfbproto.pdf.

Sanders, Chris. 2012. *Analýza sítí a řešení problémů v programu Wireshark.* Brno : Computer Press, 2012. ISBN 978-80-251-3718-5.

STALLINGS, William. 2010. *Protocol Basics: Secure Shell Protocol - The Internet Protocol Journal, Volume 12, No.4 - Cisco Systems.* [Online] 2010. [Citace: 10. 4 2014.]

http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_12-4/124_ssh.html#reference1 .

ŠIMONOVÁ, Stanislava. 2004. *Informační sítě: doplněk elektronické studijní opory.* Pardubice : Univerzita Pardubice, 2004. ISBN 80-719-4648-6.

TERMSERV. 2009. Remote Desktop Connection 7 for Windows 7, Windows XP & Windows Vista. [Online] 21. 8 2009. [Citace: 18. 4 2014.] http://blogs.msdn.com/b/rds/archive/2009/08/21/remote-desktop-connection-7-for-windows-7-windows-xp-windows-vista.aspx#9902608.

UltraVNC. 2008. UltraVNC VNC Remote Support Software Desktop Control Free Opensource. [Online] 2008. [Citace: 4. 17 2014.] http://www.uvnc.com/.

VAŠEK, Jiří. 2009. VNC a Vzdálená plocha - kouzlo vzdáleného přístupu. [Online] 11. 2 2009. [Citace: 14. 4 2014.] http://pctuning.tyden.cz/software/jak-zkrotit-internet/12639-vnc_a_vzdalena_plocha-kouzlo_vzdaleneho_pristupu?start=5.

VÝŠEK, Ondřej. 2013. RDP 8.0 pro Windows 7 a Windows Server 2008 R2. *Optimalizované IT.* [Online] 2013. http://www.optimalizovane-it.cz/windows-7/rdp-8.0-pro-windows-7-a-windows-server-2008-r2.html.

WHITE, Curt M. 2007. *Data communacations and computer networks: a business user's approach.* Boston : Thomson Course Technology, 2007. ISBN 14-188-3610-9.