UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Jure Mohar

Primerjava protokolov za prikaz oddaljenega virtualnega namizja

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE

STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

Mentor: izr. prof. dr. Miha Mraz

Ljubljana, 2011

Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Tržaška 25
1001 Ljubljana, Slovenija
telefon: 01 476 84 11
01 476 83 87
faks: 01 426 46 47
01 476 87 11
www.fri.uni-lj.si
e-mail: dekanat@fri.uni-lj.si

Št. naloge: 00156/2011
Datum: 05.09.2011

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **JURE MOHAR**

Naslov: **PRIMERJAVA PROTOKOLOV ZA PRIKAZ ODDALJENEGA VIRTUALNEGA NAMIZJA**

EVALUATION OF REMOTE DISPLAY PROTOCOLS FOR VIRTUAL DESKTOP INFRASTRUCTURE

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

Kandidat naj v svojem delu izvede analizo protokolov za prenos in prikaz oddaljenega virtualnega namizja. Pri tem naj svojo pozornost usmeri na protokole proizvajalcev Microsoft in VMware. V nadaljevanju dela naj kandidat vzpostavi dve alternativni rešitvi za prenos in prikaz oddaljenega virtualnega namizja na osnovi Hyper-V in VMware View infrastruktur. Rešitvi naj med seboj primerja z vidika zmogljivosti delovanja pod različnimi pogoji.

Mentor:

prof. dr. Miha Mraz

Fakulteta za računalništvo in informatiko
2
UNIVERZA V LJUBLJANI

Dekan:

prof. dr. Nikolaj Zimic

I Z J A V A O A V T O R S T V U

diplomskega dela

Spodaj podpisani/-a Jure Mohar,

z vpisno številko 63040322,

sem avtor diplomskega dela z naslovom:

Primerjava protokolov za prikaz oddaljenega virtualnega namizja

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal/-a samostojno pod mentorstvom izr. prof. dr. Mihe Mraza

- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela

- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki »Dela FRI«.

V Ljubljani, dne 15.12.2011 Podpis avtorja/-ice: _____

Zahvaljujem se svojemu mentorju izr. prof. dr. Mihu Mrazu za strokovno pomoč in potrpežljivost. Iskreno se zahvaljujem svoji družini za zaupanje in moralno podporo, ki so mi jo nudili tekom celega študija. Posebna zahvala gre tudi sošolcem, za vso motivacijo in skupno učenje, brez katerega bi težko prilezel do konca študija.

Diplomsko nalogo posvečam svoji dragi Simoni, brez katere te diplomske ne bi bilo.

KAZALO VSEBINE

KAZALO SLIK

KAZALO TABEL

SEZNAM KRATIC

AD	*Active Directory*	aktivni imenik na OS Microsoft Server
AES	*Advanced Encryption Standard*	napredni standard za šifriranje
API	*application programming interface*	programski vmesnik
CMD	*Command Prompt*	ukazna vrstica
DNS	*Domain Name Server*	sistem domenskih imen
FQDN	*Fully Qualified Domain Name*	popolnoma kvalificirano domensko ime
GDI	*Graphics Device Interface*	vmesnik grafičnih naprav
HDD	*Hard Disk Drive*	trdi disk
ICA	*Independent Computing Architecture*	/
LAN	*Local Area Network*	lokalno omrežje
MMR	*Multi-Media Redirection*	multimedijska preusmeritev
NAT	*Network Address Translation*	prevajanje omrežnih naslovov
OS	*Operating System*	operacijski sistem
P2P	*Peer-to-Peer*	protokol »vsak z vsakim«
PNRP	*Peer Name Resolution Protocol*	Microsoftov protokol »vsak z vsakim«
POC	*Proof Of Concept*	postavitev testnega okolja
RDC	*Remote Desktop Connection*	povezava z oddaljenim namizjem
RDP	*Remote Desktop Protocol*	protokol oddaljenega namizja
RFB	*Remote Frame-Buffer*	protokol za oddaljen slikovni pomnilnik
RPC	*Remote Procedure Call*	klic za oddaljeni postopek
RTT	*Round Trip Time*	čas obhoda
SNMP	*Simple Network Management Protocol*	preprosti protokol za upravljanje omrežja
SSD	*Solid State Drive*	negibljiv disk
SSH	*Secure Shell*	varnostna lupina
SSL	*Secure Socket Layer*	sloj varnih vtičnic
TCP	*Transport Control Protocol*	protokol za krmiljenje prenosa
TS	*Terminal Services*	terminalske storitve
TTL	*Time-To-Line*	življenjska doba
UDP	*User Data Protocol*	protokol za uporabniške podatke
VDI -	*Virtual Desktop Infrastructure*	virtualizacija računalniških namizij
VM	*Virtual Machine*	virtualna naprava
VPN	*Virtual Private Network*	navidezno zasebno omrežje
WAN	*Wide* Area *Network*	prostrano omrežje

POVZETEK

V diplomskem delu smo primerjali protokole za prikaz oddaljenega virtualnega namizja. Protokola nad katerima smo izvedli bolj podrobno zmogljivostno analizo sta bila RDP (Remote Desktop Protocol) in PCoIP (PC-over-IP). RDP je verjetno najbolj razširjen in poznan protokol za dostop do oddaljenega namizja. Kot njegov neposredni konkurent za potrebe VDI okolja je bil razvit protokol PCoIP. V tem diplomskem delu nas je zanimalo predvsem, kako se ta dva protokola obneseta na različnih povezavah, koliko računalniških virov porabita in kakšna je uporabniška izkušnja. Testirali smo protokole RDP 7, RDP z RemoteFX ter PCoIP (View 5). Za potrebe zmogljivostne analize smo postavili dve testni VDI okolji, eno na infrastrukturi Microsoft Hyper-V in drugo na infrastrukturi VMware View. Rezultati analize so pokazali, da se oba protokola v LAN okolju obneseta odlično, vendar pozitivno izstopa RDP z RemoteFX. V WAN okolju pa je predvsem z vidika uporabniške izkušnje lovoriko dobil protokol PCoIP.

Ključne besede: RDP, PCoIP, RemoteFX, VDI, virtualizacija računalniških namizij, oddaljeno namizje, protokoli za prikaz oddaljenega namizja

ABSTRACT

In this thesis, we evaluated remote display protocols for virtual desktop infrastructure. Protocols, over which we carried out a more detailed comparative analysis, were RDP (Remote Desktop Protocol) and PCoIP (PC-over-IP). RDP is probably the most widespread and well-known protocol for accessing a remote desktop, whereas PCoIP protocol was developed as a direct competitor to RDP for the VDI environment. In this thesis, we present through comparative analysis, how these two protocols work in different contexts and how much resources they consume. In addition, we investigate and evaluate user experience. Protocols we have tested are RDP 7, RDP with RemoteFX and PCoIP (View 5). For our performance analysis, we have set up two VDI environments, first one on the Microsoft Hyper-V infrastructure and the second one on VMware View infrastructure. The results of the analysis have shown that both protocols worked very well in LAN environment, yet RDP with RemoteFX worked a bit better. In WAN environment, mainly because of good user experience, the winning protocol was PCoIP.

Keywords: RDP, PCoIP, RemoteFX, VDI, virtual desktop infrastructure, remote desktop, remote display protocols

1 UVOD

Virtualizacija računalniških virov je ena izmed najbolj prodornih tehnologij, ki je v zadnjih letih dodobra spremenila podobo podatkovnih centrov. Koncept virtualizacije se iz strežniškega okolja vse bolj seli tudi na ostala področja, kot so omrežna infrastruktura, pomnilniški sistem in virtualizacija računalniških namizij. Slednje je področje s katerim se srečujem na dnevni ravni, zato se dobro zavedam pomembnosti protokolov za prikaz oddaljenega virtualnega namizja.

1.1 Virtualizacija računalniških namizij

Virtualizacija računalniških namizij (angl. *VDI*) je trenutno ena izmed najhitreje razvijajočih se tehnologij. Obljublja zmanjšanje stroškov lastništva osebnih računalnikov, boljši izkoristek računalniških virov, manjšo električno porabo, večjo varnost in lažje vzdrževanje. Koncept virtualizacije računalniških virov je v strežniškem prostoru že dobro ukoreninjen, sedaj pa se seli tudi v prostor računalniških namizij. VDI v kombinaciji z lahkimi odjemalci ponuja cenejšo in enostavnejšo alternativo klasičnim osebnim računalnikom, zato ni čudno, da se vse več podjetji odloča za tovrstni pristop. Posledično narašča tudi število ponudnikov. Izraz VDI je skovalo podjetje VMware, ki se je prvo zavedalo prednosti, ki jih lahko tak pristop ponuja. Kljub začetnemu nasprotovanju so mu kmalu sledili tudi ostali proizvajalci strojne in programske opreme kot so Microsoft, Citrix, Oracle, RedHat, HP, MokaFive, Kaviza, itd. Pri tehnologiji VDI imamo na uporabnikovi strani odjemalca z zaslonom. Slednji je lahko PC, prenosnik, tablični računalnik ali lahki odjemalec, na strani strežnika pa imamo uporabnikov operacijski sistem in njegove podatke, ki so varno shranjeni v podatkovnem centru. Ključni komponenti in ob enem krivca zakaj se tak pristop ni prijel že desetletja prej, sta protokol za prikaz oddaljenega namizja ter pasovna širina povezave.

1.2 Motivacija

Če želimo uporabljati VDI moramo v prvi vrsti zagotoviti povezljivost med odjemalcem in gostiteljem, prenos slikovnega medpomnilnika pa zagotovijo protokoli za prikaz oddaljenega namizja. Slednji bodo v tem diplomske delu tudi opisani in analizirani. Zmogljivostno analizo bomo opravili nad protokoloma RDP in PCoIP. RDP je lastniški protokol podjetja Microsoft in je verjetno najbolj razširjen protokol za dostop do oddaljenega namizja, saj je vključen v Microsoftove izdelke že od leta 1998. PCoIP je lastniški protokol podjetja VMware in Teradici, razvit za potrebe VDI okolja in kljub temu, da je relativno nov, vseeno predstavlja močno konkurenco protokolu RDP. Omenjena protokola sem izbral iz dveh razlogov. Prvi razlog je aktualnost, saj sta bila oba protokola pred kratkim posodobljena (RDP7 in View5).

Drugi razlog pa je *FlipIT* – storitev računalništva v oblakih, ki jo ponuja podjetje NIL, kjer delam. *FlipIT* ponuja končnim strankam celovito IT rešitev v oblakih in ker je VDI sestavni del te rešitve, se na dnevni ravni srečujem z omenjenima protokoloma.

Želeli smo analizirati oba protokola pod različnimi pogoji ter brez nepredvidljivih vplivov, zato smo morali postaviti dve VDI okolji. Na infrastrukturi VMware *View* bi sicer lahko testirali tako PCoIP kot tudi RDP protokol, vendar smo želeli preizkusiti tudi RDP protokol s tehnologijo *RemoteFX*, ki je primarno namenjena VDI okolju in deluje samo preko Microsoftove infrastrukture *Hyper-V*. Med analizo nas je predvsem zanimalo, kako se ta dva protokola obneseta na različnih povezavah, koliko računalniških virov porabita in kakšna je uporabniška izkušnja. Različne pogoje na povezavi smo ustvarili z WAN emulatorjem *WAN-Bridge*, meritve pa samo opravili s programskima monitorjema *Performance Monitor* ter *Fraps*. Rezultati meritev so grafično prikazani na slikah 12 – 21.

2 ODDALJENO NAMIZJE

Za potrebe pričujočega diplomskega dela moramo ločiti dva zelo povezana pojma, ki pa zajemata drugačne tehnologije. To sta oddaljen dostop in oddaljeno namizje. V svetu informatike se izraz oddaljen dostop (angl. *remote access*) nanaša na zmožnost dostopa do oddaljenega računalnika oz. naprave, preko oddaljenega ali lokalnega omrežja. Glede na način dostopa oz. pravice dostopa ločimo tri večje kategorije dostopov:

a) dostop do oddaljenih datotek,
b) dostop za potrebe administracije,
c) dostop do oddaljenega namizja (angl. *remote desktop*).

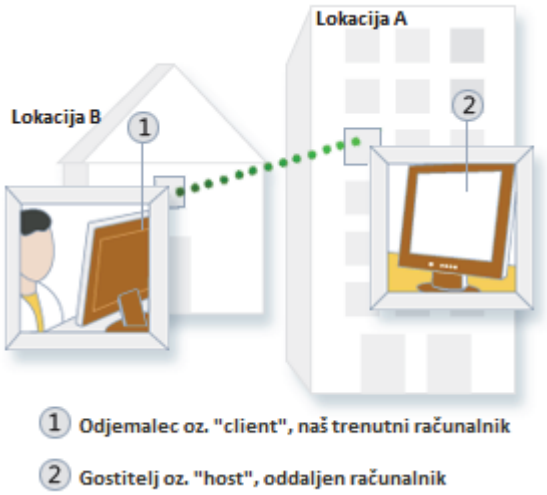
Slednji kategoriji se bomo posvetili bolj podrobno v naslednjih poglavjih, zato naj najprej omenimo ostali dve:

a) Dostop do oddaljenih datotek nam omogočajo razni protokoli, kot so NFS, SMB (CIFS), FTP in še mnogi drugi. Njihov namen je omogočiti dostop odjemalcu (angl. *client*) do datotek, ki se nahajajo na oddaljenem gostitelju (angl. *host*).
b) Dostop za potrebe administracije je kategorija, ki jo lahko sicer uporabljamo tudi v sklopu ostalih dveh, vendar se bomo v diplomskem delu osredotočili na funkcionalnosti, ki jih ostali dve ne ponujata. Mrežne naprave lahko preko omrežja administriramo preko različnih protokolov in orodij. Omenili bomo le nekaj najpogostejših, katerim pa je skupna funkcionalnost in oddaljena konfiguracija oz. dostop. Najbolj uporabljani protokoli so Telnet, SSH, SNMP ter različni protokoli za vzpostavitev virtualnih privatnih omrežij (VPN). Omenimo naj še nekaj orodij za uporabo teh protokolov, kot so ukazna vrstica (CMD), program Putty, HyperTerminal ter mnogo drugih.

2.1 Dostop do oddaljenega namizja

Oddaljeno namizje pomeni prikaz namizja, ki je zagnan na gostitelju (angl. *host*) na lokaciji A ter prikazan na zaslonu odjemalca (angl. *client*) na lokaciji B (slika 1). V smeri od gostitelja do odjemalca se prenaša samo slika, v obratni smeri pa samo ukazi preko vhodnih naprav kot sta miška in tipkovnica. Torej imamo poleg vpogleda v namizje oddaljenega računalnika tudi kontrolo nad njim, enako kot če bi fizično sedeli pred njim, ob predpostavki, da imamo za tako dejanje ustrezne pravice. Zaslonska slika na odjemalčevi strani je kopija gostiteljeve, ki se spreminja na določen interval, ali ko pride do sprememb zaradi interakcije z aplikacijami, ki so bile sprožene na gostiteljevi ali odjemalčevi strani. Gostitelj v obeh primerih izvede

aktivnost, kot bi bila zahteva izvedena lokalno. V primeru oddaljenega namizja nimamo dostopa le do datotek, temveč do celotnega sistema z vsemi pripadajočimi napravami (diski, tiskalniki, itd.), ki so priklopljene na gostitelja. Prav tako lahko dostopamo do omrežja, v katerem se le-ta nahaja. Kasneje bomo omenili tudi zmožnost preslikav odjemalčevih naprav v gostiteljevo sejo, kar je zelo praktično predvsem v primeru tiskanja.



Slika 1: Oddaljeno namizje.

Ideja oddaljenega namizja je prisotna že mnogo let. Začetki segajo že v leto 1984, ko so na inštitutu za tehnologije v Massachusetts (MIT) razvili X Window System, pozneje poimenovan kar X11 protokol, ki je v izboljšani različici v uporabi še danes. Namen protokola je bil omogočiti oddaljen grafični vmesnik, ki bi bil neodvisen od arhitekture sistema in bi omogočal deljenje procesorske moči in aplikacij na gostitelju med več uporabniki oz. odjemalci.
Oddaljeno namizje pa je v pravem pomenu besede prišlo z izidom Microsoft Windows NT 4.0 Terminal Server-ja leta 1998. Microsoft je takrat predstavil terminalne storitve (angl. *terminal services*), ki so ob pomoči terminalnega odjemalca (angl. *terminal services client*) omogočale uporabnikom dostop do aplikacij in podatkov na oddaljenem računalniku preko omrežja z uporabo protokola RDP (angl. *remote desktop protocol*). Microsoft je razvoj oddaljenega dostopa pripeljal tako daleč, da je slika na odjemalčevem zaslonu postala identična sliki na gostiteljevem zaslonu in tako posledično, ob dovolj hitri povezavi, skoraj ne uspemo razlikovati ali se prikazuje oddaljeno namizje ali lokalno.

V zgodovini oddaljenih dostopov sta prevladovala protokola ICA (Citrix) in RDP (Microsoft), ki sta slikovni medpomnilnik (angl. *frame buffer*) osvežila približno enkrat na vsakih 100 milisekund, kar je zadostovalo za prikaz enostavnih GDI (angl. *graphics device interface)*

aplikacij. Nikakor pa to ni bilo primerno za grafično intenzivne aplikacije, kot so Aero[1], 3D aplikacije ali aplikacije, ki zahtevajo zvočno in video sinhronizacijo. S prihodom virtualizacije računalniških namizij (VDI), ki se jo oglašuje kot alternativo klasičnim osebnim računalnikom, je potreba po hitrejšem osveževanju in odzivnosti storitev oddaljenega dostopa postala še večja [13].

2.2 Uporaba oddaljenega namizja

Uporaba oddaljenega namizja sega od enostavne uporabe za dostop do domačega ali službenega namizja, do oddaljene pomoči končnim uporabnikom, pa vse do zahtevnih administratorskih opravil. Ker so potrebe različne, je na voljo mnogo izvedb oddaljenega namizja od odprtokodnih do plačljivih, vsaka pa ima svoje prednosti in slabosti. Glede na implementacijo protokola razlikujemo tiste, ki omogočajo vstop v obstoječo sejo, kar pomeni da so odjemalčeve aktivnosti simultano vidne na strani gostitelja npr. VNC, in tiste, ki gostitelju zaprejo sejo in onemogočijo prikaz zaslonske slike ter izvajanje ukazov npr. RDC. Slednje bomo težje uporabljali za oddaljeno pomoč končnim uporabnikom, saj nam ne bodo uspeli pokazati kaj je narobe, in mi njim ne, kako napako rešiti.
Najbolj pa nas bo v tem diplomskem delu zanimala uporaba oddaljenega namizja za potrebe virtualizacije računalniških namizij (VDI), saj je to področje danes zelo aktualno, tako med IT strokovnjaki, kot med proizvajalci VDI rešitev, med katerimi ni videti konca rivalstvu.

2.2.1 Oddaljena pomoč

Oddaljen dostop do namizja za potrebe tehnične pomoči je zelo praktična rešitev, ki prihrani mnogo časa in nepotrebne logistike. Danes lahko v le nekaj sekundah prevzamemo kontrolo nad uporabnikovim računalnikom, ki nas je poklical za pomoč. Zbirka aplikacij, ki ponujajo tovrstno rešitev, je ogromna in presega meje tega diplomskega dela, zato bomo omenili le nekaj najpogostejših. Za nekaj izmed naštetih orodij bomo nekoliko kasneje tudi opisali princip delovanja.

Najbolj znano orodje za oddaljeno deljeno namizje je VNC (angl. *virtual network computing*), ki je verjetno zaradi odprtokodne narave tudi tako priljubljen in razširjen, saj podpira vrsto različnih operacijskih sistemov. Zadnje čase je postal zelo priljubljen Teamviewer, ki je za nekomercialno uporabo brezplačen in veliko bolj odziven kot VNC preko slabših povezav (npr. WAN). Zelo uspešen je tudi produkt ISL Light slovenskega podjetja XLAB, ki ponuja eno izmed najvarnejših povezav ter kopico dodatnih funkcionalnosti, vendar pa je plačljiv in

[1] grafični uporabniški vmesnik v operacijskih sistemi Windows Vista in 7, ki omogoča prosojnost oken, »živi« predogled, animacije ozadja in ikon, itd.

posledično bolj osredotočen na podjetja. GoToMyPC in Logmein sta prav tako zelo priljubljeni rešitvi, vendar sta bolj kot oddaljeni pomoči namenjeni oddaljenemu dostopu in kontroli nad našim računalnikom. V zadnjem času se pogosto pojavljajo tudi rešitve, ki temeljijo na HTML5 protokolu. Slednji je vsebovan v večini novih brskalnikov, zato posledično ne potrebujemo namestitve dodatnih programov. Programi, ki uporabljajo tak pristop, so ThinRDP, ThinVNC in Chrome Remote Desktop, Spark View, Ericom AccessNow, itd. Slednji je sicer plačljiv, vendar ga odlikuje predvsem odzivnost in funkcionalnost, saj naj bi nadomestil namenske programe, kot so RDC in View Client na odjemalcih, kjer le-teh ne moremo namestiti zaradi premalo pravic ali nezdružljivosti, npr. Chromebook ali računalniki v spletnih kavarnah. Od vseh naštetih HTML5 odjemalcev sta za oddaljeno pomoč primerna le ThinVNC in Chrome Remote Desktop, ker ne uporabljata RDP protokola in tako ne prekineta gostiteljeve seje, ko se povežemo nanj. Bolj podrobno bomo delovanje teh protokolov in orodij opisali nekoliko kasneje.

V sklop aplikacij oddaljene pomoči spada še en produkt, ki je že vključen v vse operacijske sisteme Windows od XP različice naprej. Oddaljena pomoč Windows (angl. *windows remote assistance*) omogoča začasen prevzem kontrole nad oddaljenim namizjem preko lokalnega omrežja ali interneta. Za razliko od »Povezave z oddaljenim namizjem« oz. RDC, se gostiteljeva seja ne prekine, ostane aktivna in odjemalcu oz. uporabniku, ki nudi pomoč, dovoli, da se pridruži v obstoječo sejo.

2.2.2 Virtualizacija računalniških namizij

Virtualizacija računalniških virov je ena najbolj prodornih tehnologij, ki je v zadnjih letih dodobra spremenila podobo podatkovnih centrov, čeprav začetki te tehnologije segajo daleč v leto 1964, ko je IBM razvil prvi operacijski sistem CP-40, ki je podpiral virtualna okolja. Koncept se danes iz strežniškega okolja vse bolj seli na druga področja, kot so omrežna infrastruktura in pomnilniški sistemi. Najbolj odmevna pa je v zadnjem času virtualizacija računalniških namizij (angl. *virtual desktop infrastructure* - VDI), ki s stališča vzdrževanja in celotnih stroškov lastništva obljublja cenejšo, varnejšo in enostavnejšo alternativo običajnim osebnim računalnikom, saj želi ločiti programski uporabniški del od strojnega. VDI želi uporabniški del (operacijski sistem, programe in namizje), ki je bil do sedaj nameščen na lokalni delovni postaji, prestaviti na varno v podatkovni center. Posledično, fizično delovno postajo lahko nadomestimo z lahkim odjemalcem (angl. *thin client*). Z virtualizacijo namizja se v celoti odstrani odvisnost fizične naprave od celotnega uporabniškega operacijskega okolja. Končnemu uporabniku tako ostane le preprost odjemalec, kot vmesnik za zaslon, miška in tipkovnica, s katerima lahko upravlja oddaljeno namizje v podatkovnem centru. Odjemalec je lahko »zero« odjemalec, lahki odjemalec, PC, tablični računalnik ali pametni

telefon. Ker se celotno procesiranje odvija na strani gostitelja, je odjemalec lahko zelo enostaven, poceni ali celo star odslužen PC. Ni pomembno kateri OS teče na odjemalcu, na »zero« odjemalcih pa celo ni operacijskega sistema, niti centralno procesorske enote, trdega diska ali delovnega pomnilnika. Vse delo opravi posebno prilagojen čip. Delo, ki ga opravlja odjemalec, ni zahtevno. Zadolžen je le za prikaz slike oddaljenega namizja na lokalno priklopljen zaslon ter pošiljanje ukazov, dobljenih iz tipkovnice in miške, nazaj proti gostitelju. Prednost idejne zasnove teh odjemalcev je tudi možnost takojšnje zamenjave v primeru okvare ali kraje, saj na lokalnem računalniku oz. odjemalcu ni nobenih podatkov in ga lahko v trenutku zamenjamo z drugim. Končni uporabnik se mora le prijaviti v oddaljeno namizje in tako lahko nadaljuje delo tam, kjer ga je končal oz. kjer je bil prekinjen.

VDI je le eden izmed konceptov virtualizacije namizij (angl. *desktop virtualization*), ki jih lahko delimo glede na to, kje se izvaja virtualni operacijski sistem; lokalno na fizični napravi (angl. *client-hosted*) ali na oddaljenem strežniku (angl. *server-hosted*). V obeh konceptih govorimo o virtualnih namizjih, ki gostijo na hipervisorju, le da je ta v prvem primeru na lokalnem računalniku, v drugem pa na strežniku. Ker VDI gostuje svoja virtualna namizja na strežniškem hipervisorju, spada v kategorijo »Server-Hosted Virtual Desktops« (SHVD).
V nadaljevanju virtualizacijo računalniški namizij delimo še na dve kategoriji glede na način dostave virtualnega namizja končnemu uporabniku:

- statično virtualno namizje (angl. persistent, stateful); vsak uporabnik dobi svoje unikatno namizje, ki si ga lahko prilagaja in shranjuje podatke,
- dinamično virtualno namizje (angl. *non-persistent, stateless*); obstaja eno osnovno namizje iz katerega se naredi kopija (angl. *linked clone*), ko se uporabnik prijavi, obenem pa se porabniški podatki, ki so shranjeni ločeno, združijo s kopijo ob prijavi.

Veliko je že bilo in še bo povedanega o tem, v kateri model računalništva spada VDI. Vsi se bolj ali manj strinjamo z že prej omenjeno kategorijo SHVD, popolnoma drugačna pa je situacija, ko govorimo o t.i. »Server-Based Computing« (SBC) modelu računalništva. Koncept VDI je zelo podoben že desetletja znanemu terminalskemu načinu dela (angl. *terminal services*), vendar pa je zasnovan na drugačnih temeljih ter posledično za drugačne potrebe. Ključna razlika med SBC in VDI je, da se pri SBC več uporabnikov poveže na en deljen TS strežnik ali Citrix Desktop, medtem ko se pri VDI en uporabnik poveže samo s svojim namizjem [10].

Če povzamemo:

- VDI omogoča uporabniku preko protokola za prikaz oddaljenega namizja dostop do svojega virtualnega namizja, ki gostuje na oddaljenem strežniku.

- TS (Microsoftovo poimenovanje »Session-based remote desktops«) omogoča uporabniku preko protokola za prikaz oddaljenega namizja dostop do namizja na večuporabniškem strežniku, vendar le kot izolirano sejo brez večjih pravic.
- SBC je tehnologija, kjer se aplikacije izvajajo na oddaljenem strežnika, odjemalcu pa so nato preko protokola za prikaz oddaljenega namizja poslane samo posodobitve zaslona, kar pa v osnovi ni nič drugega kot omogočata VDI in TS.

Podobnosti se kažejo tudi v ključnih značilnostih, ki jih navajajo tako zagovorniki TS kot tudi zagovorniki VDI, so centralno upravljanje (angl. *central management*), dostopnost (angl. access), zmogljivost (angl. performance) in varnost (angl. *security*) [9]. Vse naštete prednosti so tudi prednosti »server-based computing« modela.

Z vidika končnega uporabnika je naloga obeh načinov (TS in VDI) zagotoviti dostop do oddaljenega namizja, ki gostuje na oddaljenem strežniku. Da lahko to nalogo izpolnita, se morata zanesti na ključno komponento v tem procesu, in sicer na protokol za prikaz oddaljenega namizja (RDP, PCoIP, ICA,…). V nadaljevanju diplomskega dela bomo največ pozornosti namenili prav protokolom za oddaljena namizja.

2.2.3 Primerjava virtualizacije računalniških namizij in terminalskih storitev

V nadaljevanju bomo primerjali značilnosti virtualizacije računalniških namizij (VDI) in terminalskih storitev (TS).
Pri tehologiji VDI aktivnosti enega uporabnika ne vplivajo na delovanje drugih uporabnikov, pri čemer je vsak uporabnik omejen na sistemska sredstva (vire) v okviru svoje virtualne naprave. V primeru, da pride do napake na eni izmed virtualnih naprav ali njeni programski opremi, ostale virtualne naprave delujejo nemoteno. Za razliko od VDI pri terminalskih storitvah (TS) sistem postane neodziven za vse ostale, če se pojavi napaka na eni izmed sej. Prednost TS pristopa je v tem, da ni potrebe po upravljanju množice samostojnih virtualnih namizij, saj imamo na strežniku le eno instanco OS z več sejami (en uporabnik – ena seja).
Ena izmed značilnosti VDI je, da lahko individualne virtualne naprave ločeno ponovno zaženemo, ugasnemo, ali damo v stanje mirovanja ter jih tako prestavimo na drug strežnik. Prav tako je uporabnikom omogočena popolna kontrola nad svojim navideznim namizjem, ki si ga lahko prilagodijo glede na svoje potrebe. Tudi aplikacije je možno namestiti neposredno v Windows okolje, s čimer se izognemo zapletenim postopkom priprave za delovanje v okolju, za katerega aplikacija dejansko ni bila namenjena. Z razliko od VDI pristop TS ponuja aplikacije, ki so nameščene samo na strežniku, kar pomeni lažje posodabljanje in upravljanje.

Utrjevanje namizja (angl. *desktop hardening*[2]) je pri VDI sicer priporočljivo, ni pa obvezno za razliko od terminalnih storitev (TS) in Citrix XenApp. Prednost TS so manjše strojne potrebe na strežniku, saj se večino bremena prenese na odjemalca.
Pri VDI imamo s funkcijo »Offline Desktop« možnost uporabe namizja tudi kadar ni povezave do omrežja (kopija virtualne naprave se prenese na lokalni računalnik, ki ob ponovni vzpostavitvi lokalne omrežja posodobi le spremembe). Nastavimo lahko »TTL« vrednost, ki omeji delovanje v »offline« načinu le za določeno obdobje.

Tabela 1: Primerjava TS (RDS) in VDI.

	TS (RDS)	VDI
Zrelost tehnologije	Preizkušena in zrela	Nastajajoča in v razvoju
Nadgradljivost	Več uporabnikov / strežnik	Manj uporabnikov / strežnik
Izolacija / Varnost / Personalizacija (osebna prilagoditev)	• Izolacija na ravni seje • Deljen OS med uporabniki • Zagnano kot standardni uporab. • Nižja osebna prilagoditev	• Izolacija na ravni VM • Namenski OS na uporabnika • Lahko zagnano kot administrat. • Višja osebna prilagoditev
Združljivost aplikacij	Windows Server OS	Windows OS (desktop)

Tako VDI kot TS sta ključni komponenti pri virtualizaciji namizij, saj vsaka izmed komponent zadovolji določene računalniške potrebe in scenarije. Za nezahtevne uporabnike (klicni center), ki opravljajo delo le na določenih preverjenih aplikacijah, bo TS še vedno povsem uspešen. Po drugi strani pa bodo zahtevni uporabniki morali poseči po VDI rešitvi, saj jim ta pristop omogoča lasten nadzor nad svojim namizjem.

2.3 Protokoli

V koncept oddaljenega namizja spada ključna vmesna komponenta, ki sliko ustvarjeno na gostitelju po svojih najboljših močeh prenese in upodobi na strani odjemalca. Ta ključna komponenta je protokol za prikaz oddaljenega namizja. V nadaljevanju bomo preučili nekaj različnih protokolov, poudarek pa bo predvsem na RDP in PCoIP protokolih.

2.3.1 Protokol X Window System

Protokol X Window System imenovan tudi X ali X11 je bil razvit leta 1986 na *Institute of Technology (MIT)* v Massachusetts-u. Kasneje je postal de facto standard za okenski sistem

[2] Proces zagotavljanja večje varnosti sistema, z zmanjševanjem nepotrebnih funkcionalnosti, storitev, uporabniških računov, nepotrebne programske opreme. Proces deluje po principu: »namenski sistem je bolj varen od večnamenskega«.

na vseh vodilnih Unix delovnih postajah. Za razliko od predhodnih zaslonskih protokolov, ki so bili namenjeni prikazovanju na fizično priklopljene zaslone, je bil X11 izdelan za potrebe prikazovanja preko omrežnih povezav. Arhitektura protokola temelji na modelu odjemalec/strežnik (angl. *client/server*) in je popolnoma neodvisna od operacijskega sistema, kar protokolu omogoča veliko fleksibilnost. Model odjemalec/strežnik se nekoliko razlikuje od standardnega, ki smo ga vajeni. Ponavadi več odjemalcev dostopa do strežnika, medtem ko ima pri X11 vsaka lokalna delovna postaja svojo X strežnik aplikacijo in lahko dostopa do oddaljene naprave na kateri teče X odjemalec. Najlažje si je predstavljati strežnik kot uporabnikov terminal in odjemalca kot aplikacijo. X zagotovi aplikaciji zaslon in I/O storitve in ima zato funkcijo strežnika. Ker aplikacija uporablja te storitve, ima funkcijo odjemalca.
Naloga X strežnika je, da zahteve sprejete od X odjemalca pošlje gonilniku naprave. Ta jih nato predela v strojno kodo razumljivo za grafično kartico. Zahteve, ki jih pošilja odjemalec, so zelo enostavne, kot v primerih »nariši črto od točke do točke«, »nariši pravokotnik od točke do točke«, itd. Sam izgled namizja oz. aplikacij na odjemalcu se lahko nekoliko razlikuje od gostitelja, saj protokol X temelji na ukazih za primitivne oblike izrisa in ne uporablja enostavnega kopiranja slikovnega medpomnilnika. Tak pristop dovoljuje pospešeno izvajanje tako 2D, kot tudi 3D operacij na oddaljenem X strežniku, vendar se tu pojavi težava pasovne širine in zakasnitev. Dolžnost X strežnika je tudi sprejem ukazov iz tipkovnice in miške ter prenos le-teh X odjemalcu, da jih ta lahko izvede. Naloga X odjemalca je zgraditi okno, gumbe, meni, naslovno vrstico, itd., in jih nato preko zbirke funkcij XLib poslati X strežniku, ki jih prikaže na zaslonu. Vendar X ne določa specifikacij za izgled in obliko teh gradnikov. Za to poskrbijo aplikacije »window manager«, »GUI widget toolkits« in »desktop environments«. Tu tiči tudi razlog, zakaj se lahko izgled istih aplikacij od računalnika do računalnika razlikuje in imamo prosto izbiro, kakšen grafični uporabniški vmesnik bi želeli imeti. To omogoča npr. Linux za razliko od Microsoft Windows, kjer smo omejeni le na enega. Za lažje razumevanje se moramo zavedati, da ni potrebe da sta X odjemalec in X strežnik na isti delovni postaji, saj X11 protokol za komunikacijo med X odjemalcem in X strežnikom uporablja TCP/IP sklad, kar v praksi pomeni, da se komunikacija lahko vzpostavi preko lokalnega omrežja ali interneta. Preko interneta lahko varno komunicirata tudi preko tunelske povezave s šifriranjem seje. X11 uradno ne podpira zvoka.
Praktični primeri uporabe so administracija oddaljene naprave preko grafičnega vmesnika, uporaba aplikacije za skupinsko delo večjega števila terminalskih uporabnikov, poganjanje zahtevnih simulacij na oddaljenem računalniku in prikaz rezultatov na lokalnem računalniku, poganjanje grafičnega programa na več računalnikih na enkrat in nadzor le-teh z enim samim zaslonom, miško in tipkovnico.
X Window uporabljajo predvsem operacijski sistemi Unix, najdemo pa ga tudi v operacijskem sistemu HP OpenVMS, v različnih MAC OS X. V MAC OS X (v10.5 Leopard)

je vključen X.org (X11R7.2). Microsoft Windows privzeto ne podpira X standarda, vendar pa ga lahko podpira z raznimi implementacijami, kot so Cygwin/X, Xming, WeirdX, itd.

2.3.2 Protokol RFB

RFB (angl. *remote frame buffer*) je enostaven protokol za oddaljen dostop do grafičnega vmesnika. Originalno je bil razvit v Olivetti Research Laboratory (ORL), kot tehnologija za prikaz oddaljenega zaslona na enostavnem lahkem odjemalcu (angl. *videotile*) z ATM povezavo. Protokol je postal bolj uporaben, ko se je razvila odprtokodna aplikacija VNC, s katero so se objavile tudi specifikacije protokola RFB, in od takrat je protokol brezplačno na voljo komurkoli. Leta 2002 je bil ORL zaprt, ključni razvijalci protokola RFB in aplikacije VNC pa so se združili pod imenom RealVNC, Ltd., ki imajo blagovni znamki tudi v lasti. Protokol je še vedno brezplačno na voljo na njihovi internetni strani.
RFB je resnično protokol za lahke odjemalce, saj so že pri načrtovanju poudarili, da naj bi imel čim manjše zahteve od odjemalca. Na ta način bi lahko odjemalci delovali na širokem spektru strojne opreme. Protokol naredi odjemalce tako rekoč brez stanj (angl. *stateless*). V praksi to pomeni, da se stanje uporabniškega vmesnika, v primeru da odjemalec prekine povezavo s strežnikom in se nato poveže nazaj na isti strežnik, ohrani.
Protokol RFB deluje na nivoju slikovnega medpomnilnika (angl. *frame buffer*) in je zato primeren za vse sisteme oken in aplikacij, vključno z X11, Windows in Macintosh. Najbolj znan program, ki uporablja RFP protokol, je že omenjeni VNC. Kljub svojim prednostim pa ima tak način tudi slabosti, saj je pogosto manj efektiven kot rešitve, ki ponujajo boljše razumevanje grafičnih primitivov, kot sta protokola X11 ali RDP. Ta dva protokola pošiljata neposredno grafične primitive ali ukaze v enostavni obliki (npr. »odpri okno«), za razliko od RFP protokola, ki pošilja le neobdelane slikovne pike. Sicer tudi RFB uporablja grafični primitiv, vendar le pravokotnik. Protokol na ta način vzame slikovne pike v pravokotniku v danem x,y položaju in ga pošlje odjemalcu. V takšni osnovni obliki je izris dokaj neučinkovit, saj porabi veliko pasovne širine. Iz tega razloga obstaja mnogo izboljšav predvsem na nivoju kodiranja. Te izboljšave nam omogočajo, da s kompromisom med različnimi parametri (pasovna širina omrežja, hitrost risanja na odjemalcu, hitrost obdelave na strežniku, kvaliteta prikaza) omogočimo boljšo uporabniško izkušnjo končnim uporabnikom.
Zaporedje teh pravokotnikov predstavlja posodobitev slikovnega medpomnilnika. Posodobitev je izrecno zahtevana s strani odjemalca, kar pomeni, da je posodobitev iz strežnika poslana odjemalcu le kot odgovor na zahtevo. To daje protokolu možnost prilagoditve. Dejstvo je, da počasnejša bosta odjemalec in povezava, nižja bo raven posodobitev. Pri tipičnih aplikacijah, kjer se spremembe dogajajo na istem območju slikovnega medpomnilnika, se posodobitve zgodijo takoj ena za drugo. Pri slabših povezavah in počasnih odjemalcih pa se prehodna stanja zanemarijo, saj s tem ustvarimo manj

omrežnega prometa in manj obremenimo odjemalca z izrisovanjem. Žal pa ima ta pristop tudi slabo stran, saj se slikovni medpomnilnik ne izrisuje povezano in deluje po principu diapozitivov. Če torej premikamo okno iz točke A v točko B, ne bomo uspeli videti poteka premika, pač pa samo kako je bilo okno v danem trenutku v točki A, naslednji trenutek pa v točki B. Tak način nikakor ne more biti primeren za aplikacije, kjer se spreminja veliko slikovnih pik na enkrat (npr. video).
Kot smo že omenili, obstaja več različic in načinov kodiranja. V začetni interakciji se med odjemalcem in strežnikom vključi pogajanje o formatu in kodiranju slikovnih pik, ki bodo poslane. Ta pogajanja so narejena tako, da je delo odjemalca čim bolj enostavno, saj mora biti na koncu pogajanj strežnik sposoben poslati slikovne pike v formatu, ki jih podpira odjemalec. V primeru, kjer odjemalec podpira več oblik kodiranj, lahko strežnik izbere tistega, ki je njemu lažji za obdelavo.

Format slikovnih pik se nanaša na predstavo posameznih barv. Najpogostejši formati so 24 in 16 bitni (»*true colour*«) ter 8 bitni (»*colour map*«). Kodiranje predstavlja način, kako bo pravokotnik slikovnih pik poslan preko žice. Vsak paket s pravokotnikom slikovnih pik ima v glavi zapisan X,Y položaj na zaslonu, ter širino in višino in tip kodiranja. V paketu nato sledi podatek kodiran s specifičnim tipom. Seznam različnih tipov kodiranj vzdržuje RealVNC Ltd in so prikazana v tabeli 2.

Tabela 2: Različni tipi kodiranj za protokol RFB.

Unikatna ID številka	**Tip kodiranja**
0x00000000	Raw
0x00000001	CopyRect
0x00000002	RRE (Rising Rectangle)
0x00000004	CoRRE (Compact Rising Rectangle)
0x00000005	Hextile
0x00000006	Zlib
0x00000007	Tight
0x00000008	ZlibHex
0x00000009	Ultra
0x00000010	ZRLE
0x00000011	ZYWRLE
0xFFFF0001	CacheEnable
0xFFFF0006	XOREnable
0xFFFF8000	ServerState (UltraVNC)
0xFFFF8001	EnableKeepAlive(UltraVNC)

Unikatna ID številka	Tip kodiranja
0xFFFF8002	FTProtocolVersion (File Transfer Protocol - UltraVNC)
0xFFFFFF00 - 0xFFFFFF09	CompressLevel (Tight encoding)
0xFFFFFF10	XCursor
0xFFFFFF11	RichCursor
0xFFFFFF18	PointerPos
0xFFFFFF20	LastRect
0xFFFFFF21	NewFBSize
0xFFFFFFE0 - 0xFFFFFFE9	QualityLevel(Tight encoding)

2.3.3 Protokol Citrix ICA

Independent Computing Architecture (ICA) protokol je razvilo podjetje Citrix System za potrebe poganjanja Microsoft Windows aplikacij na oddaljenih strežnikih. Prav tako kot X11 protokol, je tudi ICA protokol neodvisen od operacijskega sistema in podpira poleg Microsoft Windows še številne Unix platforme, Mac OS, iOS, Android, itd. Aplikacije za X11 protokol je bilo potrebno posebej razvijati, medtem ko Citrix ICA omogoča uporabo vseh aplikacij, ki so bile razvite za operacijski sistem Microsoft Windows. Protokola X11 in ICA predstavljata pravo implementacijo lahkih odjemalcev, saj se izvajanje aplikacijske logike odvija 100 odstotno na strežniku. Kljub temu, da je cilj obeh protokolov isti (zagnati aplikacijo na enem računalniku in rezultat prikazati na drugem), sta si protokola močno različna. ICA na strani odjemalca oz. oddaljenega računalnika ne zahteva nobenega poznavanja grafičnih primitivov in ne izvaja upodabljanja, kot se to dogaja pri X11 protokolu. Slednji je prav zato tudi veliko bolj potraten, če gledamo z vidika pasovne širine. ICA podpira senčenje seje (angl. *session shadowing*), administratorju pa je omogočeno pridružiti se v obstoječo uporabniško sejo. ICA popolnoma loči komponento zaslona in komponento aplikacije. V praksi to pomeni, da lahko prekinemo sejo s terminalom, aplikacijo pa pustimo zagnano. Ko se bomo naslednjič prijavili v sejo, bo ta aplikacija še vedno odprta in v takšnem stanju kot smo jo pustili.

Prva javna različica ICA protokola se je pojavila v produktu WinView za operacijske sisteme OS/2, temelječ na večuporabniškem jedru MultiWin. Po zatonu OS/2 je Citrix objavil produkt WinFrame, ki je bila kombinacija Windows NT 3.51 in MultiWin tehnologije. WinFrame strežniki so izvajali Windows aplikacije, do katerih so uporabniki lahko hkrati dostopali z različnimi odjemalci prek omrežnih povezav. Ključni element WinFrame tehnologije je bil prenosni protokol, ki je razdelil aplikacijsko logiko in uporabniški vmesnik. Preko omrežja so potovali le signali tipkovnice, miške in podatki za obnovitev zaslona. ICA protokol je uporabnikom dal občutek, da delajo z lokalno nameščeno aplikacijo, čeprav se je le-ta izvajala na oddaljenem strežniku.

ICA je visoko optimiziran za prenašanje grafičnih zaslonskih slik, tako da za normalno delo zadostuje že 28,8-kilobitna modemska povezava s strežnikom. To je fleksibilen in razširljiv protokol nastavljen tako, da se prilagaja različnim zmogljivostim odjemalca. Med rokovanjem ICA odjemalec posreduje strežniku informacije o resoluciji ekrana, globini barv, velikosti pomnilnika in ostale podatke, na podlagi katerih se protokol ustrezno prilagodi. Takšna komunikacija omogoča veliko število različnih odjemalcev, od enostavnega terminala do visoko zmogljive delovne postaje ter tabličnih računalnikov.
Za protokol ICA je odjemalec naprava za prikaz, strežnik pa gostitelj aplikacije. Ko uporabnik pošlje vhodne ukaze preko miške in tipkovnice strežniku (po ICA navideznem kanalu za vhodne ukaze), jih ta izvede in upodobi. Strežnik nato te grafike kompresira in jih po drugem navideznem kanalu pošlje odjemalcu. Ko ima odjemalec enkrat osnoven pogled, mu strežnik pošilja samo še posodobitve zaslona in s tem prihrani mnogo pasovne širine. ICA receiver na strani strežnika lahko sprejme tako Windows ICA odjemalce kot tudi Unix, Linux, MAC, DOS ICA odjemalce.
Produktov, ki uporabljajo prenosni ICA protokol, je veliko. Poleg že omenjenega WinFrame, ki ga je nasledil MetaFrame in nato Presentation Server (le-ta se danes imenuje Citrix XenApp), je tu še Citrix XenDesktop, AnywhereTS, Goverlan Remote Control, Remote Desktop Manager, itd.
Ključni izzivi prenosnega protokola so zakasnitve in hitrost, prikaz grafično intenzivnih aplikacij ter slaba pasovna širina povezave. V takih primerih je potreba po kompresiji in optimizaciji neizogibna, v primeru da želimo odjemalcu zagotoviti primerno uporabniško izkušnjo. Lahko imamo opravka z odjemalcem z različno platformo in premalo lokalnih računalniških virov. V takem primeru moramo poslati preko omrežja kar bitno sliko. Glede na zmožnost odjemalca lahko tudi razbremenimo strežnik. Primer so multimedijske vsebine, ki se izvajajo na samem odjemalcu. Protokol ICA privzeto teče preko TCP vrat 1494, ali pa ovit v CGP protokol preko vrat 2598. ICA podpira več kanalov na sejni plasti ISO/OSI modela za potrebe USB preusmeritev in drugih medijskih komponent.
Naj omenim še HDX (High Definition User Experience), ki je velikokrat omenjen v povezavi z ICA protokolom, vendar nikakor ni nova verzija protokola, kot se ga včasih napačno interpretira. Tehnologija HDX je skupek različnih protokolov in tehnologij za zagotavljanje uporabniške izkušnje visoke ločljivosti. Sicer HDX bazira na ICA protokolu, vendar vsebuje še mnogo drugih tehnologij, ki s skupnimi močmi končnemu uporabniku ponujajo uporabo Flash tehnologij, spletne kamere, VoIP, avdio, 3D grafike,itd.

2.3.4 Protokol PCoIP

PCoIP grafični protokol je bil razvit s strani podjetja Teradici za potrebe prikazovanja zahtevnih grafičnih aplikacij na daljavo. Med zahtevne grafične aplikacije štejemo video

animacije, 3D in CAD aplikacije, slike za medicinske potrebe, itd. Da bi zadovoljili velike zahteve vseh naštetih aplikacij, so pri omenjenem podjetju ustvarili 1:1 rešitev osnovano popolnoma na strojni opremi. Prilagojen PCoIP čip v gostitelju je opravljal kodiranje in čip v odjemalcu je izvajal dekodiranje. Ker pa je ta sicer zelo uspešna in robustna implementacija protokola le 1:1 rešitev, je tudi zelo draga in neprimerna za večje število uporabnikov. Prav iz tega razloga sta VMware in Teradici združila moči in razvila programsko različico tega protokola.

PCoIP je strežniško centraliziran protokol, kar pomeni, da se vsa zahtevna opravila opravijo na močnih strežnikih. PCoIP protokol se popolnoma zanaša na upodabljanje [3] (angl. *rendering*) vseh slikovnih pik [4] (angl. *pixels*) na strani gostitelja, z izjemo uporabe multimedijske preusmeritve oz. MMR. Lahko povzamemo, da so vsi GDI primati in ostali grafični zahtevki upodobljeni s strani gostiteljeve centralne procesne enote (CPU). V primeru da gre za programsko implementacijo oz. da imata odjemalec in gostitelj strojno implementacijo, potem upodabljanje izvrši grafična procesna enota (GPU). Večina sistemov za oddaljeno namizje temelji na odjemalčevem upodabljanju (angl. *client-rendering*), kar pomeni da so grafični ukazi poslani odjemalcu, le-ta pa jih nato izvede oz. upodobi na zaslonu. Problem takšnega pristopa ni le potreba po zmogljivem »lahkem odjemalcu«, pač pa tudi v protokolu, ki mora biti stalno posodobljen za potrebe novih kodekov in novih funkcionalnosti. Primer tega je Microsoft, ki je izdal Aero vmesnik v Visti, večina protokolov za oddaljeno namizje pa tega ni podpirala.

Nasprotno PCoIP deluje na nivoju slikovnih pik, saj vzame celoten upodobljen zaslon, kodira slikovne pike, način upodabljanja pa ga ne zanima. Zato tak način podpira vse trenutne in bodoče grafične protokole in kodeke, dodatno pa nam da na izbiro veliko število končnih odjemalcev, katerih zmogljivost skorajda ne igra nobene vloge. Za odjemalce so zelo primerni namenski lahki odjemalci ali »zero« odjemalci, ki so cenovno bolj ugodni in vzdržljivi od navadnih računalnikov, predvsem pa porabijo zelo malo električne energije.

GDI primitive PCoIP protokol uspešno upodablja s porabo manj kot 2 % centralne procesne enote v primeru pisarniških aplikacij. Procesorsko zelo obremenjujoče pa je upodabljanje pri dekodiranju videa in pri 3D aplikacijah, vendar PCoIP dobro izkorišča večjedrne procesorje.

Ključne prednosti upodabljanja na strani gostitelja so kompatibilnost, varnost in prilagajanje protokola glede na prepustnost povezave v realnem času. Kompatibilnost z vsemi trenutnimi in bodočimi aplikacijami ter video kodeki je zagotovljena, ker le-te niso odvisne od odjemalčeve naprave. Tudi za varnost je poskrbljeno, saj podatki ne zapustijo podatkovnega centra, pač pa se preko omrežja prenašajo samo kriptirane slikovne pike. S prenosom stisnjene bitne slike se lahko protokol prilagodi v realnem času z upoštevanjem razpoložljive pasovne

[3] pretvorba visokonivojskega opisa v grafično obliko

[4] najmanjša logična enota bitne slike v pomnilniku, ki ima določeno svetlost in barvo

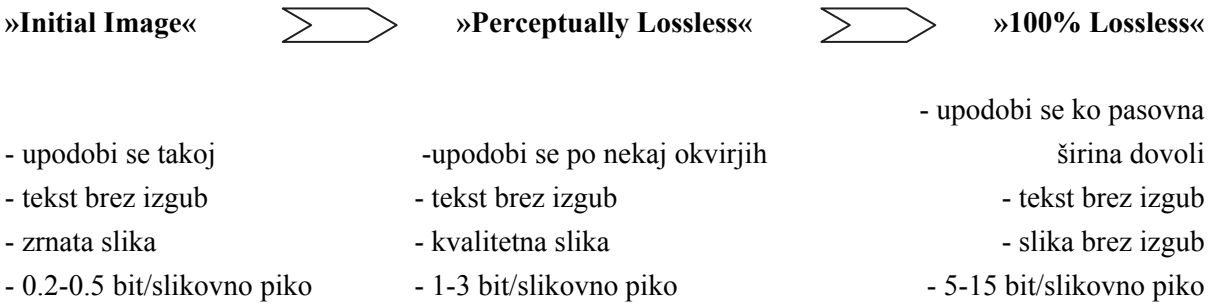
širine in zakasnitev (angl. *latency*). Na WAN povezavah z manj pasovne širine in večjimi zakasnitvami bo slika manj ostra in bo porabila 0.2-0.5 bitov/slikovno piko. Pri malo boljših povezavah bo slika kvalitetna in bo porabila 1-3 bitov/slikovno piko, medtem ko bomo za sliko brez izgub porabili 5-15 bitov/slikovno piko. PCoIP protokol ne porabi najmanj pasovne širine, Citrix pa to s pridom uporablja v marketinške namene, da pokaže svoj ICA protokol v boljši luči. Le-ta porablja skoraj fiksno pasovno širino, ne glede na hitrost povezave. Fiksna poraba pasovne širine se v slabih razmerah izkaže za dobro, vendar ni razloga zakaj ne bi na dobrih povezavah želel izkoristiti pasovne širine. Dober protokol bo porabil toliko pasovne širine, kolikor jo bo lahko izkoristil. Več podatkov bo preteklo prek žice, manj bo obremenitev za gostitelja in odjemalca (manj kompresije) in boljša bo uporabniška izkušnja. Zakaj bi torej omejevali protokol tam, kjer za to ni potrebe? Tega dejstva se PCoIP protokol dobro zaveda in izkoristi ponujeno, saj se zna prilagoditi glede na prepustnost povezave v realnem času.

2.3.4.1 Delovanje PCoIP protokola

Programska implementacija PCoIP protokola uporablja TCP in UDP vrat 50002. TCP se uporablja za vzpostavitev seje in kontrolo, medtem ko je UDP namenjen optimalnemu delovanju in prenosu pretočne vsebine. Podatki so med prenosom kriptirani s 128 bitnim AES šifrirnim algoritmom.

Na uporabnikovem zaslonu navadno najdemo različne vsebine vključno z ikonami, video vsebino, slike, grafe, tekst,itd. Vsaka izmed naštetih vsebin ima posebne značilnosti, ki določajo, kako so zaznane s strani uporabnika. Zaradi te raznolikosti bi bilo naivno meniti, da lahko z enim samim kodekom (npr. MPEG) uspešno kompresiramo celotno uporabniško namizje. MPEG je resnično zadovoljiv samo za pretočne video vsebine, če pa bi z njim predstavili tekst ali interaktivne aplikacije, pa bi bila uporabniška izkušnja (zaradi slabe kvalitete teksta in slabe odzivnosti) bržkone nezadovoljiva. Veliko protokolov za oddaljeno namizje uporablja samo 1 ali 2 kodeka za kompresijo celotne vsebine oddaljenega namizja, kar pa ne more zadostovati za dobro uporabniško izkušnjo. Neefektivna bi na primer bila uporaba kompresije z izgubo (angl. *lossy*) za celoten zaslon in tudi za tekstovno vsebino, saj bi končni uporabnik videl tekst kot neoster in zamegljen. Zato PCoIP posebej za tekstovno vsebino uporablja učinkovito brezizgubno kompresijo in tako zmanjša pasovno širino in porabo CPE. Močan poudarek na kompresiji teksta je logičen odgovor, če pogledamo vsebino zaslona povprečnega uporabnika (npr. tekst na internetni straneh, v PDF, elektronskih sporočilih, dokumentih, itd.). Protokoli za oddaljeno namizje porabijo za »cleartype / anti-alias« pisavo približno 24% pasovne širine. Naloga dobrega protokola je aktivno spremljanje vsebine, ki jo uporabnik pregleduje in dinamično izbira algoritme za kompresijo [8].

PCoIP protokol vzame neobdelane slikovne pike iz slikovnega medpomnilnika, jih kategorizira v različne slikovne tipe (angl. *image decomposition*) in jih nato kompresira z uporabo najbolj primernega kodeka za vsak slikovni tip posebej. Optimizacija in razkroj na slikovne tipe (grafike, teksti, ikone, fotografije, video, itd.) se uporabljata zaradi čim bolj učinkovitega prenosa in čim manjše porabe pasovne širine. Po optimizaciji se slikovne pike še kriptirajo in pošljejo preko LAN ali WAN do odjemalca, kjer jih PCoIP kodeki dekodirajo in postavijo v t.i. *lossless* stanje – stanje brez izgub oz. enako kot je bilo pred prenosom. Na najvišjem nivoju je ta proces identičen ne glede na to, ali gre za programsko ali strojno implementacijo na gostitelju. Privzeto PCoIP na strani odjemalca res ustvari identično sliko t.i. »*lossless image*«, kot je na strani gostitelja, vendar ne takoj. Ko se zaslon na gostitelju posodobi, PCoIP takoj pošlje odjemalcu začetno sliko (angl. *initial image*), kateri nato sledijo posodobitve, dokler ni dosežena visoka kakovost slike z izgubami (angl. *high quality lossy picture*) t.i. »*perceptually lossless image*«. In šele nato, če bo zaslon ostal nespremenjen, bo PCoIP v ozadju izpopolnjeval sliko dokler ne bo identična oz. brez izgub – »*lossless*«. Ta postopek se imenuje »*built to lossless* (BTL)« in privzeto traja cca 30 sekund.

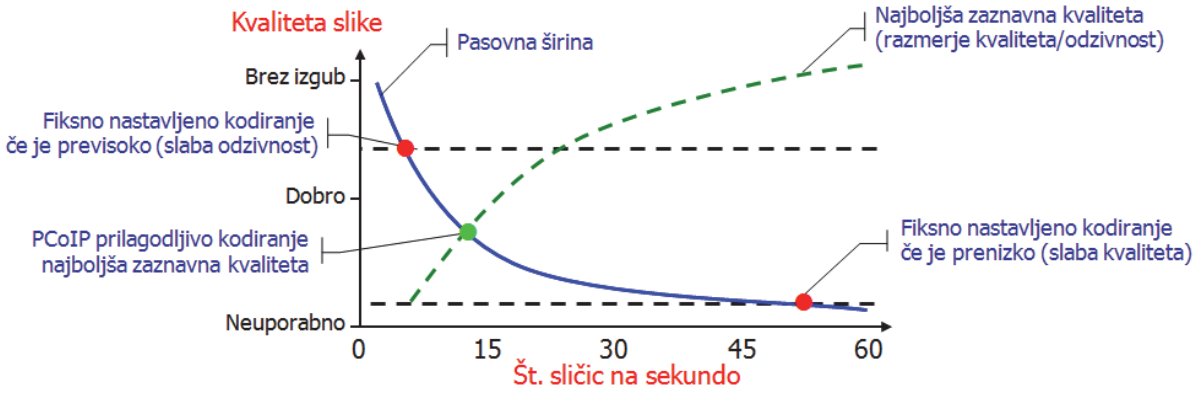


Slika 2: Postopek izgradnje slike brez izgub.

Ker nekateri uporabniki potrebujejo bolj kot kvalitetno slike odzivnost (slika 3), lahko pri PCoIP protokolu funkcijo BTL izklopimo in tako zmanjšamo porabo pasovne širine. Z vklopljenim BTL naj bi nezahteven uporabnik porabil 60Kb/s, z izklopljenim pa 48Kb/s, vendar so to le okvirne številke.

Vsak protokol za oddaljeno namizje sprejema kompromise med kakovostjo slike in odzivnostjo oz. številom sličic na sekundo FPS (angl. *frame rate*). Ne glede na pasovno širino, pa naj bo še tako slaba, imamo lahko visoko kakovost slike, vendar na račun malega števila sličic. Posledično to pomeni zelo slabo interakcijo z uporabnikom. V obratnem primeru več sličic na sekundo predstavlja boljšo odzivnost, vendar na račun kvalitete slike (slika 2). Pogosto protokoli omogočajo, da uporabnik sam nastavi želeno kakovost in odzivnost, takšne fiksne nastavitve pa obveljajo do nadaljnjega, ne glede na pasovno širino. Takšen pristop je lahko dokaj rizičen, ker napačne vrednosti lahko vodijo do zelo slabe kvalitete ali pa do slabe odzivnosti. PCoIP ne uporablja vnaprej določenih nastavitev temveč

optimalen kompromis med kvaliteto in odzivnostjo, ki se prilagaja glede na trenutno pasovno širino in zahtevnost grafike in ki dinamično prilagaja kodiranje, da zagotovi najboljšo možno uporabniško izkušnjo. Torej če se sprosti nekaj dodatne pasovne širine, bo PCoIP protokol samodejno izboljšal kvaliteto slike in število sličic na sekundo.



Slika 3: Kompromis med odzivnostjo in kvaliteto slike.

Ker gre pri PCoIP protokolu izključno za upodabljanje na gostiteljevi strani, odjemalci niso nič drugega kot enostavni dekodirniki (angl. *simple decompression or decoder engines*). Vzporednico bi lahko potegnili z delovanjem digitalne televizije, kjer se slikovne pike kompresirajo z uporabo MPEG ali H.264 kodeki, ki se nato dekompresirajo na odjemalčevi strani. Razlika med PCoIP kodeki in digitalnimi TV kodeki je, da so slednji neefektivni za prikazovanje tipičnih računalniških namizij, kot so tekst in računalniška grafika. Z upodabljanjem na strani gostitelja, PCoIP odjemalci le dekodirajo kompresirane slikovne pike z uporabo PCoIP kodekov in ne upodabljajo le-teh.

PCoIP protokol poskuša čim manj obremenjevati centralno procesorsko enoto pri osveževanju zaslona in zato velik del časa miruje. Za primer poglejmo uporabnika, ki odpre internetno stran, kjer se izvede nekaj aktivnosti, posodobi se zaslon, nato pa ostane statičen, dokler se uporabnik ne premakne na drugo stran.

PCoIP protokol za čim bolj učinkovito posodabljanje in čim manjšo porabo pasovne širine uporablja tudi predpomnjenje (angl. *caching*) na strani odjemalca. V mnogih primerih se le majhen del zaslona spremeni. Prav zato bi bilo neefektivno in potratno pošiljati posodobitev celotnega zaslona. Zaradi tega razloga PCoIP opravlja prostorsko filtriranje in pošlje le del zaslona, ki se je spremenil. Poleg te rešitve bi bilo smiselno tudi določene dele zaslona začasno shraniti na strani odjemalca. V primeru, da minimiziramo neko aplikacijo, ki smo jo imeli na zaslonu (upodabljanje se je že zgodilo, prenos slike je že bil opravljen), in se kasneje odločimo to isto aplikacijo maksimirati, se bo morala posodobitev zaslona ponovno prenesti,

pa čeprav gre za identične slikovne pike. Podobno se dogaja pri premikanju oken. To potratno ponovno posodabljanje PCoIP protokol elegantno reši s predpomnjenjem na strani odjemalca. Posledica je zmanjšana količina prenesenih podatkov in izboljšana uporabniška izkušnja. Privzeto je ta funkcionalnost vklopljena in se jo lahko nastavlja, vendar ne deluje na mobilnih naprava in »zero« odjemalcih.
PCoIP ponuja še eno zelo uporabno funkcionalnost »PCoIP Session Statistic«, ki omogoča pregled statistike prenosa omrežnih podatkov, avdio, slikovnih in USB podatkov za vsako virtualno namizje posebej.

2.3.4.2 »Zero« odjemalci proti programskim odjemalcem

Glavna razlika med PCoIP »zero« odjemalci in PCoIP programskimi odjemalci je vzdrževanje in varnost. »Zero« odjemalci nimajo operacijskega sistema ali brskalnika, ki bi moral biti stalno posodobljen in ne potrebujejo protivirusne zaščite ali protivohunskega programja. Vse, kar potrebujejo, je čip s kodeki za dekodiranje slikovnih pik. »Zero« odjemalci imajo sicer na voljo posodobitve, vendar niso nujno potrebne, ker gre le za dodajanje novih funkcionalnosti, za katere se stranka sama odloči, ali jih potrebuje. Teradici in VMware sta se obvezala za združljivost za nazaj za vse »zero« odjemalce s strojno programsko opremo 3.0 in novejše.
Mnogokrat je uporabniška izkušnja na »zero« odjemalcih boljša, kajti na programskih odjemalcih težko zagotovimo enake pogoje zaradi deljenja sistemskih sredstev med ostalo programsko opremo, ki ni View Client.
PCoIP protokol je sicer identičen tako za »zero« kot tudi za programske odjemalce, vseeno pa je nekaj razlik pri optimizaciji. Kadar se »zero« odjemalci povezujejo na strojno implementacijo gostitelja, se kategorizacija slikovnih pik (angl. *image decomposition*) izvaja na nivoju posamezne pike. Pri programski implementaciji ne glede ali gre za odjemalca ali za gostitelja, pa se kategorizacija slikovnih pik izvaja na nivoju blokov, kar posledično zmanjša porabo centralno procesorske enote, zahteva pa nekoliko večjo pasovno širino.
Poleg zmogljivosti so še druge razlike med omenjenimi odjemalci. Za podprte tipe videa lahko programski odjemalci uporabijo MMR (angl. *multimedia redirection)* za dekodiranje videa na odjemalčevi strani namesto na strani gostitelja, medtem ko »zero« odjemalci, ki temeljijo na Tera1 Silicon čipu, tega ne podpirajo. Prav zato mora biti celoten video dekodiran na strani gostitelja.
Naslednji večji razliki sta podpora USB naprav in virtualno tiskanje, t.i. ThinPrint. »Zero« odjemalci ne podpirajo vseh USB naprav, razen kadar so v navezi s PCoIP strojnim odjemalcem. Prav tako pa ne podpirajo virtualnega tiskanja, ki je omejeno le na programske odjemalce ter katerega funkcionalnost pride do izraza, kadar imamo na računalniku nameščen operacijski sistem s pripadajočimi gonilniki za tiskalnik. Preko programskega odjemalca

VMware View Client, ki je prav tako nameščen na računalniku, nato dostopamo do virtualne naprave v katero se samodejno preslika lokalni tiskalnik, ki je nameščen na fizični delovni postaji. Za to ne potrebujemo dodatnih gonilnikov, saj to izvede storitev ThinPrint, ki se izvaja na virtualni napravi. V primeru povezave »zero« odjemalca na strojno PCoIP implementacijo gostitelja so podprte vse USB naprave. Vendar je tak način dokaj nepriročen in drag, ker je to 1:1 rešitev, saj potrebujemo na strani gostitelja PCoIP strojno kartico, na drugi strani pa »zero« odjemalca. Poleg vsega pa lahko hkrati deluje samo ena seja. Pregled prednosti in slabosti, tako ene kot druge vrste implementacij odjemalca protokola PCoIP, si lahko ogledamo v tabeli 3.

Tabela 3: Prednosti in slabosti strojnih in programskih implementacij PCoIP odjemalca.

Strojna implementacija odjemalca	Programska implementacija odjemalca (VMware View Client)
+ najvišja možna zmogljivost in odzivnost (v navezi s strojnim gostiteljem) + odstrani potrebo po vzdrževanju delovnih postaj (brez CPE, OS, gonilnikov, anti virusov) + podpira vse USB naprave (v navezi s strojnim gostiteljem) + nedovzeten za viruse, spyware, vdore + Samsung monitor z PCoIP čipom	+ podpira standardno x86 arhitekturo strojne opreme (PC, prenosnik, lahki odjemalec) + oddaljen dostop preko prenosnikov + dostop preko internet s programskim VPN odjemalcem + predpomnjenje na strani odjemalca (Client Side Caching)
- ne podpira vseh USB naprav (v navezi s programskim gostiteljem) - ne podpira ThinPrint - potreba po namenski strojni opremi - za dostop preko interneta potrebuje VPN usmerjevalnik - predpomnjenje na strani odjemalca (Client Side Caching)	- dostop samo preko VMware View - x86 odjemalci potrebujejo vzdrževanje - odzivnost oddaljenega namizja odvisna tudi od strojne opreme odjemalca (CPE, pomnilnik,..)

2.3.4.3 Primerjave strojne in programske implementacije PCoIP gostitelja

Tako kot pri odjemalcih, imamo tudi pri gostiteljih programsko in strojno implementacijo protokola PCoIP. Kot smo že omenili, je bil PCoIP sprva zasnovan kot strojna rešitev za najzahtevnejša opravila. Ker gre pri strojni implementaciji za namenski čip brez x86 procesorja (ki kodira, prenaša, dekodira), se izognemo potrebi po operacijskem sistemu, anti-virusih programih, gonilnikih, dodatni strojni opremi (trdi disk, delovni spomin, itd.). Vendar

pa trenutna različica strojne implementacije ponuja samo 1:1 rešitev, kar pomeni, da lahko samo en uporabnik hkrati dela na gostitelju z eno PCoIP strojno kartico. Več strojnih kartic, izmed katerih vsaka zahteva PCI-express režo, lahko ponudi deljenje iste oddaljene seje.
VMware View predstavlja programsko implementacijo protokola PCoIP tako za kodiranje na strani gostitelja oz. strežnika, kot tudi za dekodiranje na strani odjemalca. Aplikacija VMware View podpira standardno x86 arhitekturo strojne opreme, tako da za upravljanje VMware virtualnih namizij preko PCoIP protokola ne potrebujemo nobene posebne strojne opreme. VMware View podpira virtualizacijo namizij. V praksi to pomeni, da en sam strežnik poganja več sej za več neodvisnih uporabnikov, ki pa lahko dostopajo tako preko strojne, kot tudi programske implementacije PCoIP protokola. Kljub večji fleksibilnosti pa je programska rešitev veliko bolj procesorsko obremenjujoča. Pregled prednosti in slabosti tako ene kot druge vrste implementacij gostitelja protokola PCoIP so predstavljene v tabeli 4.

Tabela 4: Prednosti in slabosti strojnih in programskih implementacij PCoIP gostitelja.

Strojna implementacija gostitelja	**Programska implementacija gostitelja (VMware View)**
+ najvišja možna zmogljivost in odzivnost (v navezi s strojnim odjemalcem) + podpira najzahtevnejše aplikacije (video, CAD) + nobene dodatne obremenitve CPE na strežniku + zelo varno – ni direktnega dostopa do OS gostitelja, samo preko namenskih PCoIP Ethernet vrat	+ podprta standardna strojna oprema, ki podpira VMware + fleksibilnost in prihranki zaradi virtualnih namizij
- samo 1:1 rešitev (1 kartica v gostitelju : 1 odjemalec) - potreba po namenski PCoIP strojni opremi - potreba po DVI grafični kartici in prosta PCI-Express reža	- primarno razvita za potrebe oddaljenih pisarniških aplikacij - kodiranje obremenjuje CPE na gostitelju oz. strežniku

PCoIP protokol je zelo konsistenten in se ne ozira na to, ali kodiranje in dekodiranje opravlja PCoIP strojni čip ali pa aplikacija VMware View. To ponuja vrsto različnih rešitev, od popolnoma strojnih ali programskih, do hibridnih. Slednje so kombinacija obeh in zato verjetno najbolj primerne za produkcijsko okolje.

2.3.5 Protokol RDP

RDP (angl. *remote desktop protocol*) je Microsoftov protokol, ki uporabniku omogoči grafični vmesnik do oddaljenega računalnika. RDP je razširitev protokolov družine ITU T.120, ki so v uporabi še danes in so bili razviti za potrebe multimedijskih konferenc.

RDP je večkanalni protokol, ki omogoča ločen prenos komunikacijskih podatkov, predstavitvenih podatkov in šifriranih ukazov miške in tipkovnice. Privzeta vrata na katerih posluša so TCP 3389. RDP podpira do 64.000 ločenih kanalov, ki jih s pridom izkoriščajo storitve, kot so Print Redirection, Clipboard Mapping, itd.

Z izidom Windows Server 2008 in Windows Vista operacijskim sistemom, je RDP še razširil svoje funkcionalnosti, kot so izboljšan izgled, Desktop Window Manager (DWM) za virtualizacijo, 32-bitna barvna globina, ClearType tekst in izboljšana podpora za preusmeritev naprav. Velikokrat prihaja do nesporazumov, kateri operacijski sistem uporablja določeno verzijo protokola. Označbe orodja oz. odjemalca so velikokrat zavajajoče, zato bomo to najlažje predstavili v tabelah 5 in 6.

Tabela 5: Različice odjemalcev RDC.

Odjemalec RDC	Različica lupine
Windows Server 2000 SP4	5.00.2195.6674
Windows XP	5.1.2600.0
Windows XP SP1	5.1.2600.1106
Windows XP SP2	5.1.2600.2180
Windows Server 2003	5.2.3790.0
Windows RDC 6	6.0.6000.16386
RDC 6.1	6.0.6001
RDC 7.0	6.1.7600
RDC 7.1	6.1.7601

Tabela 6: Različice protokola RDP.

Gostitelj RDP	Različica protokola RDP
Windows Server 2000	RDP 5.0
Windows XP	RDP 5.1
Windows Server 2003	RDP 5.2
Windows Vista	RDP 6.0
Windows Server 2008	RDP 6.1
Windows Server 2008 R2	RDP 7 (RDP 7.1)
Windows 7	RDP 7 (RDP 7.1)

Pomembno je vedeti, da so v zgornjih tabelah navedene privzete različice protokola in odjemalcev, ki pa jih je možno nadgraditi. Tako lahko na operacijske sisteme Microsoft XP SP3 in Vista namestimo odjemalca RDC 7 in s tem tudi pridobimo določene funkcionalnosti protokola RDP 7, ob predpostavki, da se povezujemo na gostitelja, ki podpira RDP 7.

2.3.5.1 Delovanje protokola

RDP svoj video gonilnik namesti v Windows grafični sistem enako kot pravi grafični gonilnik, le da je ta, namesto za pravo fizično grafično kartico, gonilnik za virtualno grafično kartico. Namesto pošiljanja grafičnega izrisa na fizični GPE, RDP naredi inteligentno odločitev, kako zakodirati te ukaze v RDP in jih poslati po žici. To lahko sega od kodiranja bitne slike do kodiranja manjšega ukaza, kot je nariši črto od točke do točke. RDP paketki so nato poslani preko omrežja odjemalcu, kjer le-ta odpakirane podatke interpretira v ustrezne Win32 GDI API klice in jih izriše na zaslon. Na vhodni poti so kliki miške in tipkovnice preusmerjeni od odjemalca do strežnika. Na strežniku RDP uporabi svoj gonilnik za miško in tipkovnico, ki je namenjen interpretaciji ukazov.
Vsaka različica RDP protokola uporablja močno šifriranje, ki je privzeto vklopljeno. Prejšnje različice so uporabljale RSA Security RC4 ključ, ki je bil razvit za potrebe učinkovitega šifriranja majhnih količin podatkov ter za varno komuniciranje preko omrežji. Trenutne različice uporabljajo še močnejše šifriranje in možnost preverjanja pristnosti strežnika. To je mogoče, ker je RDP zgrajen na vrhu varnostnega mehanizma CredSSP, ki uporablja Kerberos ali TLS za preverjanje pristnosti. Podobno je tudi SSL protokol uporabljen za šifriranje prometa do in od zaščitenih internetnih strani.
Poleg stiskanja in predpomnjenja bitnih slik, simbolov oz. pismenk in fragmentov (več simbolov skupaj) v RAM, RDP 5 dodaja še statično predpomnjenje na disku, ki je lahko na voljo tudi vsem naslednjim sejam drugih uporabnikov. Ta predpomnilnik lahko zagotovi boljšo uporabniško izkušnjo, predvsem na počasnejših povezavah, še posebej pri aplikacijah z velikimi bitnimi slikami.

Izraba pasovne širine

Preko RDP protokola mnogo naprav pošilja različne podatke med odjemalcem in strežnikom. Vsaka naprava porabi nekaj pasovne širine za pošiljanje svojih podatkov, kot so video, podatki v odložišču, printerji, itd. Te naprave tekmujejo med seboj za pasovno širino, kar pa ni najbolj primerna rešitev, saj po isti poti potujejo tudi grafični podatki za prikaz oddaljenega namizja na zaslonu. Posledično zaradi tega trpi kvaliteta in odzivnost oddaljenega namizja. Primer tega je tiskanje velikega dokumenta, za katerega bomo porabili dobršen del pasovne širine, pri slabih povezavah pa bomo prikaz oddaljenega namizja spremljali po »diapozitivih«. V Windows Server 2008 so to izboljšali z uvedbo preprostega sistema, ki zagotovi določen

odstotek pasovne širine samo grafičnemu upodabljanju, preostanek pa nameni virtualnim kanalom (preusmerjene naprave). Privzeta rezervirana vrednost je 70% za grafične podatke in 30% za ostale podatke.

Predpomnjenje bitnih slik

RDC odjemalec podpira predpomnjenje v delovnem spominu in statično predpomnjenje na trdem disku. Obe vrsti predpomnjenja shranjujeta bitne slike iz strežnika na odjemalčevem računalniku v delovnem spominu in disku v primeru ponovne uporabe in s tem ustvarjata prihranek na pasovni širini. Obe vrsti predpomnjenja skupaj zagotavljata večji predpomnilnik. Predpomnjenje naj bi v večini primerov prihranilo cca 25% pasovne širine. Če želimo, lahko velikost predpomnilnika poljubno nastavimo preko registra – HCU\SOFTWARE\Microsoft\Terminal Server Client\BitmapPersistCacheSize in v RDP datoteki pod argumentom »bitmapcachesize:i:1500«, kar bo nastavilo velikost na 1500 kB.
Če torej na odjemalcu ni nikakršnega problema s prostorom, je priporočljivo vklopiti čim večji predpomnilnik za zmanjšanje pasovne širine.

Desktop Composition

Windows Vista Desktop Composition je bistveno spremenil način izrisovanja oken na zaslonu. Namesto direktnega izrisovanja, je izris oken preusmerjen na »off-screen« površino v video pomnilnik. Desktop Window Manager (DWM) nato upodobi okna in jih predstavi na zaslonu. Poleg tega DWM omogoča vizualne efekte, Flip 3D, prosojno okno, itd.
RDP je bil razširjen za podporo Desktop Composition na daljavo. To lahko doseže s prenosom DWM ukazov iz strežnika na RDP odjemalca. Odjemalec interpretira te ukaze in upodobi namizje. Skupek teh DWM ukazov vodi do povečanja porabe pasovne širine. Desktop Composition na daljavo je na voljo le med operacijskimi sistemi navedenimi v tabeli 7.

Tabela 7: Uporaba Desktop Composition glede na OS.

Odjemalec \ Gostitelj	Win Vista	Win 2008	Win 7 / Win 2008R2
Win 2008	DA	NE	NE
Win Vista	DA	NE	NE
Win 7 / Win 2008R2	NE	NE	DA

Barvna globina in ClearType tekst

S prihodom Windows Vista smo pridobili nov grafični vmesnik s transparentnimi okni in Aero funkcijami. Ker so pri Microsoft-u želeli podobno uporabniško izkušnjo pripeljati tudi v oddaljena namizja, so v RDP najprej dodali 32-bitno barvno globino. Ker pa se je na testih izkazalo, da je 32 bitna predstavitev bolj učinkovita, so začeli 24 bitno opuščati. Iz tega

razloga imamo danes na voljo 16 bit High Color in 32 bit Hightest Quality. V primeru da izberemo 24 bit, bo RDP privzeto uporabil 16 bitno barvno globino.

ClearType je tehnologija za prikazovanje glajenih pisav, ki izboljšajo branost teksta na LCD zaslonih. Ker Windows Vista, Office 2007 ter novejše izvedbe uporabljajo ClearType in ob izklopu te funkcije izgled ni ravno privlačen, so RDP verziji 6.1 dodali tudi opcijo ClearType. Privzeto je ta funkcija izklopljena, ker so črke predstavljene kot pismenke, ki se predpomnijo na strani odjemalca in tako prihranijo pasovno širino (npr. znak A v pisavi Arial). S funkcijo ClearType vklopljeno pa je pisava poslana kot bitna slika, posledično je prikazana pravilno, vendar na račun veliko večje porabe pasovne širine.

2.3.5.2 Tehnologija RemoteFX

Tehnologija RemoteFX je bila predstavljena s prihodom Windows Server 2008R2 SP1 z namenom izboljšati vizualno izkušnjo RDP protokola. Microsoft obljublja uporabniško izkušnjo visoke ločljivosti, več hkratnih zaslonov, ločljivost posameznega zaslona do 1900x1200, AERO vmesnik, 3-D ter USB preusmeritev pod pogojem, da je operacijski sistem na virtualnem namizju Windows 7 SP1. RemoteFX deluje le, če to virtualno namizje poteka na Microsoftovem hipervisorju Hyper-V 2008R2 SP1. Ker pa RemoteFX izvaja zahtevno kodiranje na strani gostitelja, moramo v strežnik vgraditi zmogljivo grafično kartico (primer tega je zaslon, ki ima ločljivost 1900x1200 in porabi 220MB spomina). Zaradi velike porabe pasovne širine predstavlja edino možnost LAN rešitev [12].
RemoteFX lahko tudi uporabimo na Remote Desktop Session Host (Terminal Services) brez dodatne grafične kartice, vendar v tem primeru ne dobimo enake uporabniške izkušnje in funkcionalnosti (npr. USB preusmeritev). Poleg tega dodatno zelo obremenimo centralno procesorsko enoto strežnika.

2.4 Programska orodja

Aplikacij, ki omogočajo oddaljen dostop, je mnogo, razlikujejo pa se predvsem po protokolu na katerem sloni njihova implementacija ter po funkcionalnostih, ki jih omogočajo.
V tem delu diplomske naloge bomo predstavili nekaj dobro uveljavljenih programskih orodij za oddaljen dostop, njihove funkcionalnosti ter protokole, ki jih uporabljajo.

2.4.1 VNC odprtokodna rešitev

Virtual Network Computing (VNC) je odprtokodna rešitev za grafično deljenje namizja, ki uporablja RFB protokol za oddaljen nadzor računalnika. Deluje neodvisno od platforme in zato ne predstavlja nikakršnih težav v primeru da imata odjemalec in gostitelj različna

operacijska sistema. VNC omogoča povezavo več odjemalcev na istega gostitelja, kar je lahko koristno predvsem za predstavitve. Originalna izvorna koda je pod odprtokodno licenco, prav tako pa tudi mnogo njenih novejših izpeljank, zato se ne gre čuditi številu različic tega programa. Originalni razvijalci VNC so bili iz Olivetti & Oracle Research Labs (Cambridge), ki so kasneje pripadali družbi AT&T in tako zapustili ta projekt. Original AT&T VNC verzija ni več v množični uporabi, saj imamo danes na voljo mnogo različic s pomembnimi izboljšavami. Zelo pogoste različice so RealVNC, TightVNC, TigerVNC in trenutno najbolj razširjena UltraVNC. Vse različice so še vedno kompatibilne s starejšimi verzijami, vsaj kar zadeva osnovne funkcije programa VNC. Različice ponujajo različne funkcionalnosti. Nekatere so optimizirane za Microsoft Windows, druge ponujajo prenos datotek. Zanimivo je tudi dejstvo, da lahko z eno različico programa na gostiteljevi strani in drugo različico na strani odjemalca vseeno dostopamo do osnovnih funkcionalnosti. Kljub temu da sta VNC in RFB kodi brezplačno dostopni, sta registrirani blagovni znamki podjetja RealVNC Ltd.

VNC je odličen in enostaven program za oddaljen nadzor, ki pa mu žal primanjkuje nekaj funkcij, ki bi prišle prav za večje organizacije, predvsem z vidika beleženja in varnosti. VNC prav tako ni namenjen za zaposlene, ki delajo na daljavo ali za lahke odjemalce, ker ne predstavlja podpore za lokalno tiskanje iz oddaljenih aplikacij. Ker je bil VNC originalno razvit za potrebe lokalnega omrežja LAN, se pri povezavi preko WAN omrežja pojavljajo vprašanja glede varnosti. V primeru uporabe VNC preko javnega omrežja je močno priporočljiva souporaba VPN tunela ali zaščita celotnega VNC prometa s kriptografskim protokolom SSL. Priporočljiva je tudi uporaba gesla daljšega od 8 znakov. Kljub temu, imamo pri nekaterih različicah programa VNC omejitev znakov na maksimalno dolžino 8 in se tako vse kar je daljše, zanemari.

Celoten sistem VNC je sestavljen iz VNC strežnika (angl. *server*) na strani gostitelja ter VNC gledalca (angl. *viewer*) na strani odjemalca. Za grafični prikaz skrbi že omenjeni RFB protokol. Delovna postaja, kjer je pognan VNC strežnik, ne potrebuje fizičnega zaslona, ker za prikaz skrbi Xvnc. Xvnc je Unix VNC strežnik, ki temelji na standardnem X strežniku. Za aplikacije pognane na delovni postaji Xvnc predstavlja X strežnik, ki prikaže okno na zaslonu. Oddaljenemu VNC gledalcu pa Xvnc predstavlja VNC strežnik.

Če povzamemo, aplikacija se prikaže na Xvnc, kot bi se prikazala na normalnem X zaslonu, vendar se namesto na fizično priklopljenem zaslonu, prikaže VNC gledalcu. Kot zanimivost dodajmo, da na sistemih z operacijskim sistemom Unix/Linux, ki dovoljujejo več vzporednih X11 sej, VNC lahko vzpostavi povezavo z obstoječo sejo ali pa ustvari novo. Na sistemih z operacijskim sistemom Microsoft Windows pa lahko VNC povezavo vzpostavimo vedno le v obstoječo sejo.

VNC privzeto uporablja TCP vrata 5900, ki jih je potrebno odpreti na požarni pregradi pri gostitelju. Kadar le-teh ni možno odpreti, lahko VNC strežnik postavimo v način poslušanja

(angl. *listening mode*), kjer se poveže na odjemalca oz. gledalca preko vrat 5500, katera mora odjemalec pustiti odprta. Pri uporabi programa VNC preko interneta se poleg vprašanja varnosti pojavi še nekaj težav. Zaradi dokaj požrešnega protokola potrebujemo dovolj pasovne širine na obeh straneh povezave, poleg tega pa potrebujemo kar nekaj tehničnega znanja za nastavitev požarnih zidov, NAT translacij in usmerjevalnikov za uspešno vzpostavitev z oddaljenim namizjem.

2.4.2 Aplikacija ISL Light

Program ISL Light je zelo uspešna rešitev za oddaljeno tehnično pomoč, ki jo je izdelalo slovensko podjetje XLAB. Protokol, ki ga aplikacija uporablja za prikazovanje namizja, je lastniško zaščiten s strani podjetja in ni javno objavljen. Odlikuje ga predvsem vrsta funkcionalnosti, ki jih mnogi konkurenti ne premorejo, kot so zelo optimizirano delovanje tudi v slabših pasovnih širinah ter eden izmed najbolj varnih prenosov, saj uporablja 3-nivojsko šifriranje. Poleg že omenjenega oddaljenega namizja, ki ga lahko pogledamo ali pokažemo, ponuja ISL Light tudi možnost tekstovnega pogovora med stranko in svetovalcem. Skupen pogled je možno omejiti le na določeno aplikacijo ali del namizja. ISL Light omogoča tudi prenos datotek v obe smeri, povezavo ob ponovnem zagonu oddaljenega računalnika in risalno desko, s katero lahko svetovalec riše po strankinem zaslonu. Med stranko in svetovalcem lahko vzpostavimo varno VoIP povezavo, ki jo je možno popestriti z uporabo videa. ISL Light ponuja možnost oddaljenega tiskanja, snemanja sej in prehajanje preko požarnih zidov, zaradi česar ni potrebe po dodatnih konfiguracijah požarnih naprav. ISL Light je vsebovan v eni sami izvršni datoteki, ki se ob zagonu samodejno namesti in nastavi ter ne potrebuje ponovnega zagona.

Seje ISL Light se vzpostavijo preko številnih strežnikov porazdeljenih po vsem svetu, ki se združujejo v obsežno platformo za spletno komunikacijo. Omrežje strežnikov temelji na porazdeljeni bazi podatkov ter vsebuje mehanizme za uravnoteženo geografsko porazdelitev sej. Delovanje omrežja je na ta način neodvisno od centralnega strežnika.

ISL Light prihaja v dveh različicah, gostujoči ali strežniški. Gostujoča storitev temelji na tehnologiji računalništva v oblaku oz. poslovnem modelu SaaS (angl. *software as a service*) in zagotavlja neprestano dostopnost in stabilnost storitve. Predstavlja enostavno rešitev, ki je primerna za večino podjetij in posameznikov, saj ne zahteva nikakršnih namestitev, potreben je le en klik. Gostuje na zmogljivi globalni mreži strežnikov ISL Online Network.

Strežniška storitev je na voljo za podjetja, ki zahtevajo popolno neodvisnost in nadzor nad svojim sistemom. V tem primeru se vse seje vzpostavijo preko strežnika v strankinem podatkovnem centru. Večja podjetja ali organizacije lahko sistem namestijo na več strežnikih ter jih povežejo v korporativno omrežje za spletno komunikacijo. V strežniškem modelu je

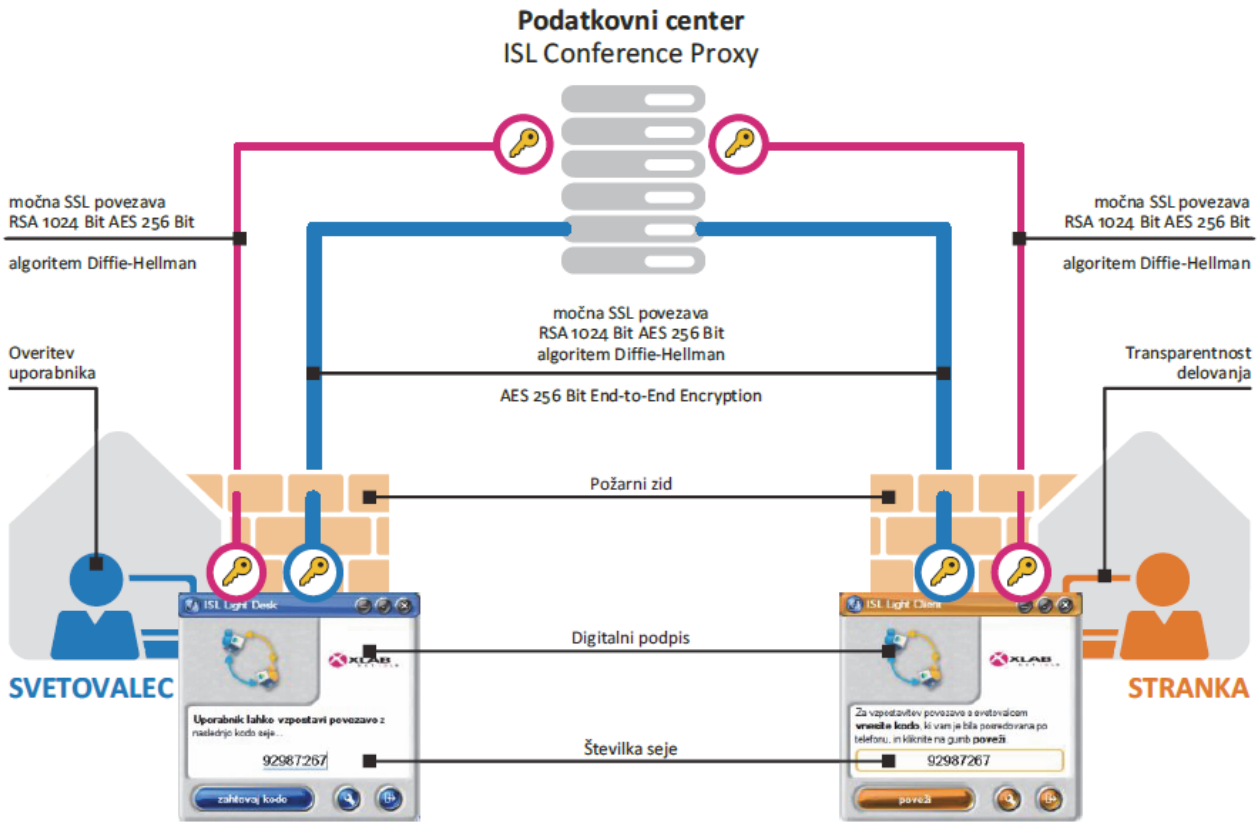
učinkovitost in dostopnost ISL Light storitve odvisna od stranke same oz. njene IT infrastrukture.

Na voljo imamo dve možnosti ISL Light delovanja:

- Preden prične podporno sejo na daljavo, se mora svetovalec prijaviti s svojim uporabniškim imenom in geslom. Ko se prijavi, pridobi številko seje in jo posreduje stranki. Številka seje je enkratna in se lahko uporabi le enkrat. Stranka na spletni strani vnese številko seje in klikne »Poveži«. Tako lahko svetovalec prične nuditi podporo na daljavo.

- Svetovalec prenese aplikacijo ISL Light Desk in jo zažene. Prijavi se s svojim uporabniškim imenom in geslom ter prične podporno sejo na daljavo. Številka seje je enkratna in se lahko uporabi le enkrat. Stranka prenese aplikacijo ISL Light Client in jo zažene. Stranka vnese številko seje in klikne na gumb »Poveži«.

Vir: www.islonline.com



Slika 4: Varnostna shema aplikacije ISL Light.

Obe aplikaciji ISL Light Desk in ISL Light Client sta potrjeni z digitalnim podpisom Verisign. Seje so zavarovane z AES 256 bitnim šifriranjem med končnima točkama (glej sliko 4). Za izmenjavo simetričnih AES 256 bitnih ključev skrbi kriptosistem z javnimi / zasebnimi 1024 bitnimi RSA ključi in algoritem Diffie-Hellman (standard za SSL). Številka seje postane po končani povezavi neuporabna. Stranka mora potrditi vsako sejo ISL Light, prav tako se vsaka funkcija potrdi ločeno in se jo lahko takoj izklopi.

2.4.3 Aplikacija Teamviewer

Teamviewer je zelo priljubljen program za oddaljeno kontrolo, deljenje namizja in prenašanje datotek med računalniki. Poznan je predvsem po svoji brezplačni naravi za nekomercialne namene ter dobri odzivnosti tudi na slabših medmrežnih povezavah. Protokol, ki ga aplikacija uporablja za prikazovanje namizja, je last podjetja in ni javno objavljen. Za uporabo aplikacije Teamviewer zaženemo izvršljivo datoteko, ki je ni potrebno namestiti in lahko teče brez administratorskih pravic. Ustvariti je potrebno še kodo in geslo (geslo si lahko po želji tudi prednastavimo), ki ju posredujemo uporabniku, kateremu želimo omogočiti dostop do našega namizja. Le-ta nato podatke vnese v lokalno zagnan TeamViewer in tako dobi nadzor nad našim namizjem.
TeamViewer uporablja RSA ključ za izmenjavo in AES 256-bitno kodiranje seje. Vendar ker uporablja le uporabniško kodo in geslo, je le-ta občutljiv na razne napade, kot je »ribarjenje« (angl. *phishing*[5]). V privzeti konfiguraciji TeamViewer uporablja enega od strežnikov teamviewer.com, ki skrbi za povezavo in usmerjanje prometa med lokalnimi odjemalci in oddaljenim gostiteljem. Na ta način poteka ves promet skozi ta strežnik. TeamViewer premore mnogo uporabnih funkcij med katerimi je potrebno izpostaviti avdio in video prenos, tekstovni pogovor in konferenčni klic. TeamViewer VPN je še ena izmed zelo zanimivih funkcij, ki omogoča vzpostavitev VPN povezave s partnerjem.

2.4.4 Aplikacija Citrix Receiver

Citrix Receiver je prenovljeni univerzalni odjemalec za dostop do virtualnih namizij (XenDesktop) ali aplikacij (XenApp) preko večine naprav in iz poljubne lokacije. Končnim uporabnikom omogoča večjo fleksibilnost, IT administratorjem pa poenostavi delo, saj jim ni potrebno več skrbeti za končne naprave, ne glede ali gre za PC, MAC ali iPhone, za službeno ali domačo napravo. Zgrajen je na varnem kriptiranem HDX (ICA) protokolu, ki določa specifikacije za prenos podatkov med strežnikom in odjemalci in deluje popolnoma neodvisno od platforme. Potrebna je le enkratna namestitev, saj so vse nadaljnje posodobitve nameščene

[5] socialni inženiring z uporabo elektronskih komunikacij

samodejno iz strežnika. Delovanje je podobno kot pri komponenti Flash za brskalnik, ki se lahko samodejno posodablja, ko se na strežniku objavijo posodobitve.
Zaradi hitrega razvoja različnih naprav, s katerimi lahko dostopamo do podatkov v podjetju ali doma, se IT oddelki srečujejo z vedno večjimi težavami, ostali zaposleni pa z vedno večjimi omejitvami. S pomočjo Citrix Receiver odjemalca, ki ga lahko namestimo na naprave z operacijskim sistemom Windows, Linux, MAC OS, iOS, Android, BackBerry, itd., lahko poenostavimo administracijo in zadovoljimo potrebe zaposlenih. Ker Citrix Receiver deluje le kot lahki odjemalec za prikaz oddaljenega namizja ali aplikacije, ki potekajo varno v podatkovnem centru, nas končne naprave niti ne zanimajo preveč, saj so lokalni podatki izolirani od podatkov v podatkovnem centru. Prednost Citrix Receiver odjemalca je prilagojenost za vsako vrsto naprav posebej. Tako se na primer pri uporabi tabličnega računalnika dotiki prevedejo tako, da se pravilno odzovejo na virtualnem namizju ali aplikaciji. Konkretni primer predstavljata iPad in Galaxy Tab, ki poganjata vsak svoj operacijski sistem, in preko katerih lahko dostopamo do oddaljenega virtualnega namizja. Protokol Citrix HDX hkrati obljublja odlično uporabniško izkušnjo s podporo za Aero vmesnik, Flash, VoIP in ostale multimedijske vsebine. Protokol HDX poleg kriptirane povezave omogoča eno najboljših optimizacij protokola za prenos preko slabših internetnih povezav.

2.4.5 Oddaljena pomoč Windows

Oddaljena pomoč Windows (angl. w*indows remote assistance*) omogoča začasen prevzem kontrole nad oddaljenim namizjem preko lokalnega omrežja ali interneta. Za razliko od aplikacije RDC se gostiteljeva seja ne prekine, ampak ostane aktivna in tako uporabniku, ki nudi pomoč, dovoli da se pridruži v obstoječo sejo. Oddaljena pomoč (angl. *remote assistanc*e) je vključena v vseh operacijskih sistemih Windows od XP različice naprej. Z operacijskim sistemom Windows Vista je oddaljena pomoč postala samostojna aplikacija, ki ne uporablja več centra za pomoč in podporo (angl. *help and support center*) ali aplikacije Windows Messenger, pač pa temelji na programskem vmesniku Windows Desktop Sharing.
Windows Desktop Sharing za razliko od Terminal Services, ki ustvari novo sejo za vsako RDP povezavo, lahko gosti oddaljeno RDP sejo v že obstoječi lokalni seji trenutno prijavljenega uporabnika. Pri tem lahko uporabimo deljenje celotnega namizja, le del namizja, ali pa določeno aplikacijo. Podprto je tudi deljenje več-zaslonskega namizja. Windows Desktop Sharing sestavljata dve ključni komponenti, ki sta RDPSession in RDPViewer. Prva omogoči deljenje seje, druga pa ogled. Več RDPViewer komponent lahko dostopa do ene same seje RDPSession. Tak način s pridom uporablja Windows Meeting Space. Uporabniki RPDViewer komponent so lahko pasivni ali aktivni gledalci. Slednji imajo možnost uporabe oddaljene aplikacije v realnem času.

Sejo z oddaljenim računalnikom lahko začasno zaustavimo, saj imamo na voljo tekstovni pogovor in diagnostiko. Oddaljeno pomoč je bila predelana tako, da danes porabi manj pasovne širine predvsem pri slabih povezavah. Prav tako sedaj podpira tudi NAT prehajanje, zaradi česar se lahko povežeta tudi uporabnika, ki se nahajata za NAT napravo. V operacijskem sistemu Windows 7 oddaljena pomoč Windows temelji na RDP 7 in pri preprostem povezovanju (angl. *easy connect*) uporablja PNRP protokol. S preprostim povezovanjem uporabnik samo posreduje 12 mestno kodo administratorju in računalnika bosta vzpostavila P2P povezavo preko lokalnega omrežja ali interneta. Ta 12 mestna koda je pravzaprav transformiran »PNRP Peer Name«, ki se objavi v PNRP oblaku, kjer so tudi ostali PNRP uporabniki oz. njihova »Peer« imena. V tem oblaku se nato interaktivno poišče ime, ki odgovarja zahtevanim kriterijem, in povezava se vzpostavi preko IPv6 omrežja.

2.4.6 Povezava z oddaljenim namizjem (RDC)

Verjetno najpogosteje uporabljen program za oddaljeno namizje je RDC (angl. *remote desktop connection*), bolj znan pod imenom Remote Desktop, še prej pa kot Microsoft Terminal Services Client (mstsc). Aplikacija uporablja storitev Remote Desktop Services za dostop do oddaljenega računalnika preko omrežja z uporabo protokola Remote Desktop Protocol (RDP). RDC prikaže oddaljeno namizje, kot bi bilo zagnano lokalno. Z različico 7.0 je uporabniška izkušnja še bolj izpopolnjena, ker omogoča 32 bitne barve, senčenje, 3D funkcijo Aero, prilagajanje resolucije zaslona ter podporo do 16 hkratnih zaslonov. Vključena je tudi preusmeritev Windows Media Player, ki poskrbi, da se zvok in slika predvajata direktno iz odjemalčeve naprave, kar zmanjša obremenitev gostitelja ter izboljša uporabniško izkušnjo. RDC omogoča tudi preusmeritev tiskalnikov, diskov, mišk, tipkovnic ter »vstavi in poženi« (angl. *plug and play*) naprav, kot so spletne kamere in optični bralniki. Na ta način so lahko vse te naprave uporabljene v oddaljenih aplikacijah. Preusmeritev dvosmernega avdia je prav tako podprta, torej je zvok, ki je ustvarjen na strani oddaljenega računalnika oz. aplikacije, predvajan na strani odjemalca. Prav tako pa je tudi zvok lahko ustvarjen preko mikrofona na strani odjemalca. Dokaj nova funkcija je Easy Print, ki omogoča uporabnikom tiskanje iz virtualnega namizja direktno na tiskalnike, priklopljene na njihove fizične računalnike, brez da bi bilo potrebno namestiti gonilnike v virtualno okolje. Ta funkcija je omejena na virtualna namizja z operacijskim sistemom Windows 7, od odjemalcev pa se zahteva le RDC 6.1 ali novejši. S kombinacijo tipk CTRL+ALT+END lahko ugasnemo ali ponovno zaženemo oddaljen računalnik. Za avtorizacijo lahko poleg uporabniških imen in gesel uporabimo tudi pametne kartice.

»RemoteApp« je poseben način Remote Desktop Services, ki je na voljo le z RDC različico 6.1 in novejšimi. Le-ta omogoča integracijo oddaljene seje v operacijski sistem odjemalca. V praksi to pomeni, da se aplikacija, ki je zagnana na oddaljenem računalniku, obnaša kot vsaka

druga lokalna aplikacija. Tako je lahko zagnanih več aplikacij v sklopu ene seje, vsaka v svojem oknu. Tudi pri takem načinu je možno normalno preusmeriti lokalne računalniške vire.

2.4.7 Aplikacija Vmware View Client

Vmware svoje orodje za oddaljen dostop do centralno gostujočih virtualnih namizji imenuje View Client. Podpira vrsto operacijskih sistemov in naprav, kot so Windows, Linux, MAC OS, iOS, Android, iPad, PC, tablični računalniki, lahki odjemalci, zero odjemalci, itd. Za uspešno vzpostavitev sta potrebni dve komponenti, View Client (na strani odjemalca) in View Agent (na strani virtualnega namizja).

View Agent je storitev (angl. *service*), ki mora biti nameščena na vseh virtualnih namizjih, fizičnih delovnih postajah ali na strežnikih s terminalnimi storitvami do katerih bomo želeli dostopati preko VMware View odjemalca. Na virtualnih namizjih View Agent komunicira z odjemalčevo aplikacijo View Client za potrebe nadzora povezave, virtualnega tiskanja ThinPrint, View Persona Management in dostopa do lokalno priključenih USB naprav.

Ko odjemalec na svoji napravi zažene aplikacijo View Client in vnese potrebne podatke o strežniku in avtorizaciji, se mu pojavi lista vseh virtualnih namizij, do katerih je opravičen. Avtorizacija lahko zahteva poverilnice iz aktivnega imenika (AD), pametne kartice ali RSA SecurID žetona. Če nam administrator dovoljuje lahko pred vzpostavitvijo izberemo vrsto protokola RDP ali PCoIP, število zaslonov in velikost zaslona. Po uspešno vzpostavljeni povezavi dobimo nadzor nad oddaljenim navideznim namizjem. V aplikaciji View Client lahko izberemo katere lokalne USB naprave želimo preusmerit v virtualno namizje, če pa nam je dovoljeno, lahko tudi ponovno zaženemo oddaljeno namizje. Če imamo na virtualnem namizju vklopljeno storitev ThinPrint, se nam bodo vsi odjemalčevi lokalni tiskalniki preslikali v virtualno namizje.

View Client lahko namestimo z ali brez lokalnega načina (angl. *view client with local mode*). To je zelo uporabna funkcija predvsem za ljudi z nestalno povezavo do službenega omrežja ali za mobilne uporabnike, ki veliko potujejo. Lokalni način omogoča, da se virtualna namizja iz podatkovnega centra prenesejo na lokalni računalnik ali prenosnik in takšnega uporabljajo tudi kadar so brez povezave. Ko se »offline« uporabniki prijavijo nazaj v omrežje podjetja, je potrebno narediti sinhronizacijo. View Transfer Server je strežnik odgovoren za uporabnike, ki uporabljajo »view client with local mode«. Njegova dolžnost je sinhronizacija med virtualnim namizjem v podatkovnem centru in lokalno nameščenim namizjem na odjemalcu.

2.4.8 HTML5 odjemalci za oddaljeno namizje

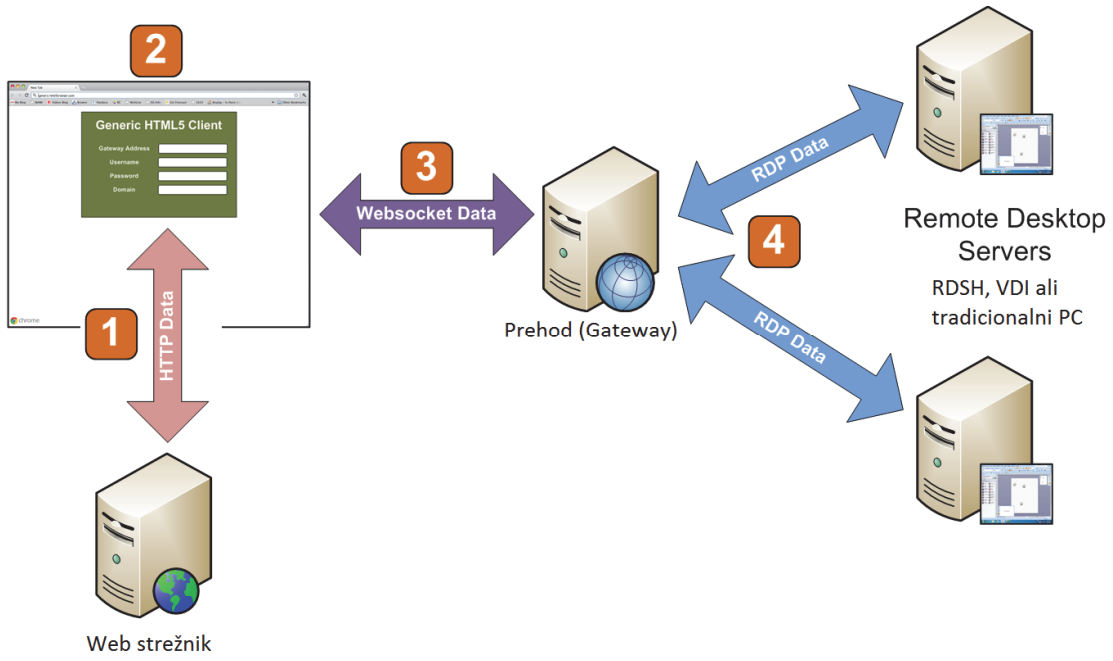
Za uspešen prikaz oddaljenega namizja v brskalnikih, ki podpirajo HTML5, obstajata dve ključni tehnologiji spletne vtičnice, »*websockets*« ter platno »*canvas*«.

»Websockets« je protokol oz. programski vmesnik (API), ki je vgrajen v vseh novejših brskalnikih in omogoča neprekinjen prenos podatkov preko ene TCP vtičnice. Za razliko od HTTP protokola ne pričakuje odgovora za vsako zahtevo. Pri slednjem več zahtev pomeni več povezav, kar postane precej kompleksno in neučinkovito za aplikacije v realnem času. »Websocket« ta postopek spremeni tako, da odpre kanal med odjemalcem in strežnikom, ki ostane odprt med zahtevami. Njegova glavna pomanjkljivost je podpora le za tekstovne podatke, ne pa tudi binarne, ki so ključnega pomena za oddaljeno namizje.

»Canvas« je ustvarilo podjetje Apple že leta 2004, kasneje pa je postal avtohton HTML5 element. »Canvas« omogoča nadzor vsake slikovne pike z uporabo »javascript« ter tako omogoča brskalnikom dinamično prikazovanje 2D grafike. Ta pristop s pridom uporabljajo razne animacije ali igre v brskalnikih, ki ne uporabljajo »Flash« pristopa (primer tega je igra Angry Birds Chrome).

Za oddaljeno namizje odjemalec (v večini primerov »javascript« aplikacija) uporabi podatke, ki prihajajo preko »websockets«, in jih izrisuje na zaslon preko platna oz. »canvas«. Binarni podatki, ki so potrebni za protokol oddaljenega namizja so dostavljeni preko prehoda (angl. *gateway*), ki ga različni proizvajalci poimenujejo vsak po svoje. Podjetje Ericom ga imenuje AccessNow Server, Spark View svojega Spark Gateway, ThinRDP pa ThinRDP Windows Server. V vseh primerih ti prehodi vzpostavijo RDP sejo z oddaljenim gostiteljem in jo prevedejo oz. zakodirajo iz binarnega v tekstovni zapis z uporabo »websockets«. Ti tekstovni podatki so nato poslani na brskalnik, kjer jih odjemalec z uporabo platna oz. »canvas« interpretira in izriše na zaslon.

Poenostavljen prikaz delovanja HTML5 odjemalcev je prikazan na sliki 5, kjer so označeni štirje osnovni koraki. V prvem koraku uporabnik odpre internetno stran s HTML5 odjemalcem. V drugem koraku uporabnik vnese uporabniške podatke, vključno z naslovom prehoda. V tretjem koraku se vzpostavi povezava s prehodom preko »websocket«. V četrtem koraku se prehod poveže z RDP strežnikom, zakodira RDP binarne podatke v tekstovne in jih pošlje nazaj odjemalcu. Odjemalec te podatke odkodira z uporabo »canvas« in jih izriše na zaslon.



Slika 5: Prikaz delovanja HTML5 odjemalcev za oddaljeno namizje.

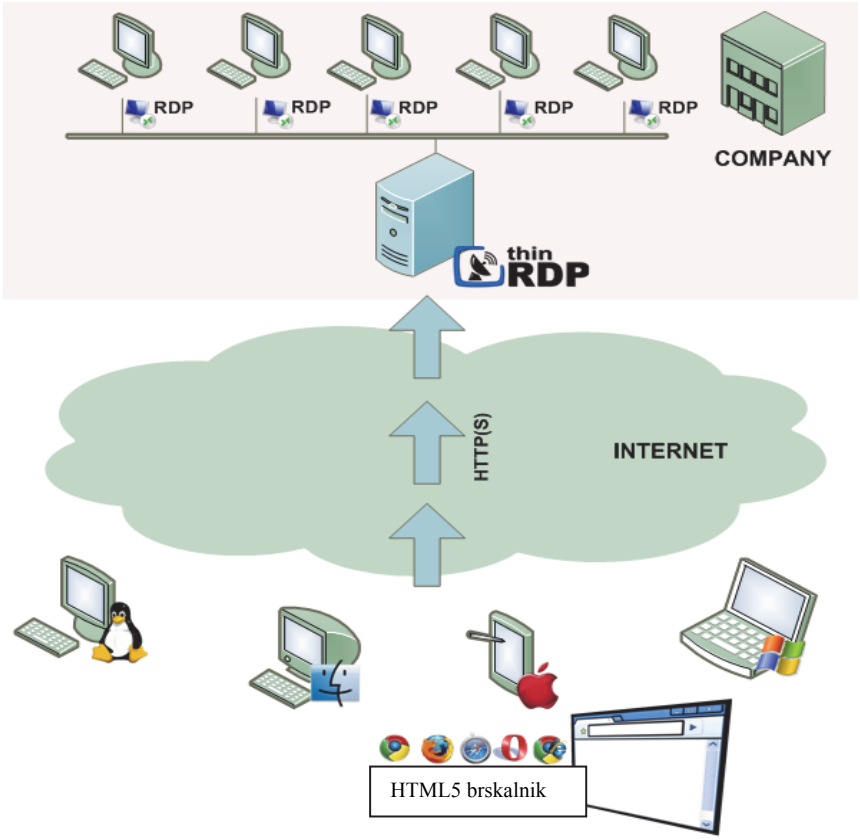
2.4.8.1 Odjemalec Ericom AccessNow

Ericom AccessNow je prvi HTML5 odjemalec, narejen za potrebe oddaljenega namizja vključno s »terminal services« in VDI. Odjemalec je napisan izključno v HTML5 kodi, kar pomeni, da ga lahko zaženemo direktno iz brskalnikov, kot so Chrome, Safari, Firefox in ostali novejši brskalniki, ki podpirajo HTML5. Za delovanje odjemalca ne potrebujemo dodatnih namestitev ali dodatkov, kot so Flash, Java, ActiveX ali Silverlight. Izjema je le Internet Explorer, pri katerem moramo za pravilno delovanje namestiti vtičnik Chrome Frame. Ker ne potrebujemo nobene dodatne namestitve, lahko odjemalca zaženemo iz katerekoli naprave, ki podpira HTML5 brskalnike, vključno z tabličnim računalnikom iPad, mobilnimi napravami iPhone in Android. Postopek vzpostavitve je enak že prej prikazanem na sliki 5. Preko spletnega strežnika (ki je lahko nameščen na ciljnem računalniku ali pa se uporabi strežnik podjetja Ericom, ki je dostopen na app.ericomaccessnow.com), zaženemo odjemalca. Ta se poveže z AccessNow strežnikom, ki je nameščen na gostitelju. AccessNow strežnik prevaja websocket v RDP format in obratno, prenos pa poteka preko varne SSL povezave. Ena izmed zelo uporabnih funkcij AccessNow je, da lahko pred samo povezavo podamo pot do točno določene aplikacije, ki se nam bo nato prikazala v brskalniku namesto celotnega namizja. Omenimo naj še dejstvo, da v omrežju, kjer bi želeli dostop do več računalnikov oz. gostiteljev RDP seje, ni nujno potrebna namestitev AccessNow strežnika na vsakemu izmed njih. Zadostuje že namestitev na enem samem računalniku, preko katerega se nato vzpostavljajo RDP seje do vseh ostalih. Zelo podobno arhitekturo kaže slika 6, le da je namesto ThinRDP strežnika nameščen AccessNow strežnik.

2.4.8.2 Odjemalec ThinRDP

ThinRDP, prav tako kot Ericom AccessNow, podpira večino naprav, ki podpirajo HTML5 brskalnike, in je namenjen varnemu dostopu do računalnika preko omrežja s samo uporabo brskalnika. ThinRDP za prikaz v brskalniku uporablja izključno samo spletne tehnologije, kot so Ajay, WebSockets in JavaScript. Dostopamo lahko tako do virtualnih namizij, kot tudi do terminalskih storitev. Prednost ThinRDP je zelo enostavna namestitev, saj ne potrebuje nobenega dodatnega spletnega strežnika. Vse je namreč že vključeno v eni namestitveni datoteki ThinRDP for Microsoft® Remote Desktop Services. ThinRDP arhitekturo sestavljata dve komponenti, ThniRDP Web Client, ki ni nič drugega kot HTML5 odjemalec ter ThinRDP Windows Server. Slednja komponenta je nekakšne vrste varni spletni strežnik, ki uporablja že omenjene »websockets« in deluje kot prehod med HTML5 odjemalcem in oddaljenim RDP gostiteljem. ThinRDP omogoča varen dostop preko enega IP naslova in vrat do kateregakoli računalnika v lokalnem omrežju, kar je razvidno iz slike 6. Če torej želimo dostopati do računalnikov, ki so v lokalnem omrežju, zadostuje namestitev ThinRDP for Microsoft® Remote Desktop Services že na enem samem računalniku in en javni IP naslov.

Vir: www.cybelesoft.com/thinrdp/

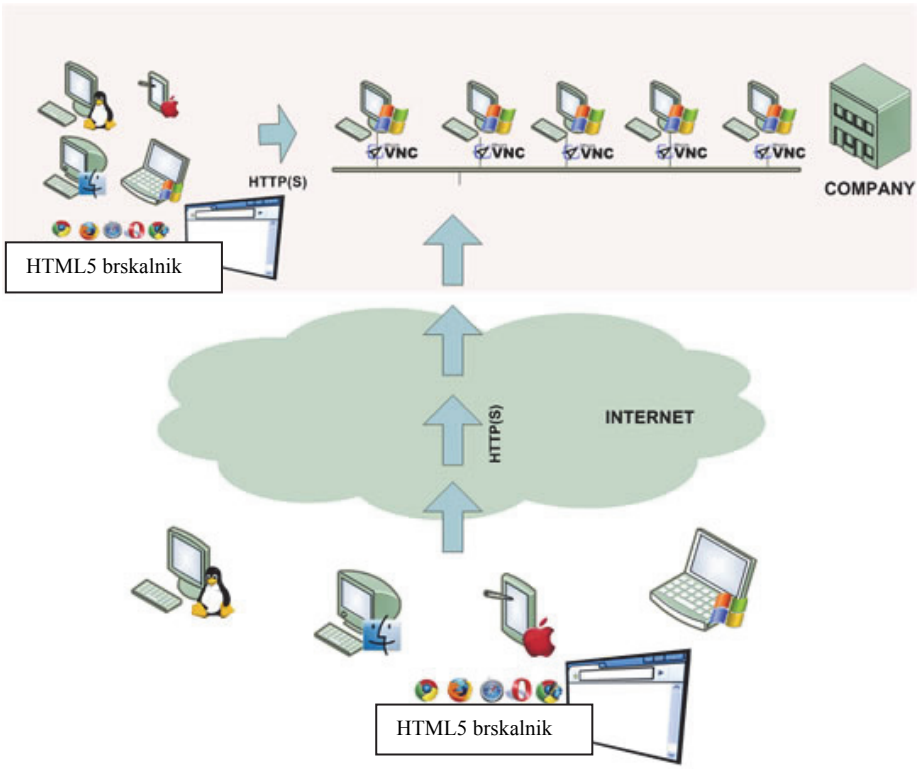


Slika 6: Arhitektura ThinRDP.

2.4.8.3 Odjemalec ThinVNC

ThinVNC za razliko od ThinRDP in AccessNow ne uporablja protokola RDP in je zato bolj primeren za oddaljeno pomoč. Gostitelj namreč ostane v svoji seji ter tako ohrani svoje namizje, ki se pokaže tudi na odjemalčevi strani. Prav tako ne uporablja standardnega VNC protokola (RFB) in namesto tega izkorišča HTML5 platno oz. »canvas« za prikaz oddaljenega namizja z uporabo spletnih standardov AJAX, JSON in HTML5 »websockets«. Kot vsi podobni programi je njegova arhitektura sestavljena iz dveh delov, HTML5 JavaScript odjemalca ter strežniškega dela ThinVNC. Spletni odjemalec se poveže na ThinVNC strežnik z uporabo AJAX in »websockets« tehnologij preko SSL povezave, če je na voljo, in tako prikaže oddaljeno namizje. Pri ThinVNC strežniški del teče na vsakem gostitelju (kot prikazuje slika 7), ki kasneje služi kot strežnik za vzpostavljanje RDP sej. Ta lastnost ga razlikuje od ThinRDP in AccessNow, kjer za strežniški del zadostuje že en sam računalnik. Ko je povezava vzpostavljena, lahko poleg kontrole nad oddaljenim namizjem izmenjujemo tudi datoteke na zelo enostaven način. Priročna funkcija ThinVNC je tudi Manage Presentation, ki omogoča prikaz namizja več uporabnikov hkrati. Vse kar morajo uporabniki storiti je, da v brskalnik vnesejo gostiteljev URL, vpišejo uporabniško ime in geslo (ki so ju prehodno dobili po e-mail sporočilu), in že lahko spremljajo dogajanje na gostiteljevem namizju.

Vir: www.cybelesoft.com/thinvnc/



Slika 7: Arhitektura ThinVNC.

2.4.8.4 Odjemalec Chrome Remote Desktop

Ne dolgo nazaj je Google splavil še en produkt za oddaljeno namizje imenovan Chrome Remote Desktop, ki predstavlja razširitev za brskalnik Chrome. To je prvi produkt, ki uporablja novo tehnologijo imenovano »chromoting«, in katerega beseda je sestavljenka besed Chrome in Remoting. Program je namenjen predvsem oddaljeni pomoči, saj nam omogoča, da z enkratnim geslom pridobimo ali oddamo nadzor nad namizje. Uporabnik, ki želi deliti svoje namizje, odpre brskalnik Chrome, izbere oddaljeno namizje in klikne »Dajte ta računalnik v skupno rabo«. Zgenerira se enkratna 12 mestna koda, ki je nato posredovana osebi, s katero želi deliti namizje. Povabljena oseba v oddaljeno namizje Chrome vnese posredovano 12 mestno kodo in že ima nadzor nad oddaljenim namizjem. Vse kar potrebujemo za delovanje je brskalnik Chrome in nameščena razširitev Chrome Remote Desktop.
Protokol deluje na podlagi več Google tehnologij. Spodnji sloj je P2P povezava, vzpostavljena z »libjingle [6] «. Ker potrebujemo zanesljivo povezavo, se uporablja implementacija TCP imenovana PseudoTcp. Pri tej implementaciji gre sicer za neke vrste zanesljiv TCP prenos, vendar preko nezanesljive povezave kot je UDP. Na vrhu vsega teče SSL povezava, medtem ko je za kodiranje in učinkovito strukturiranje podatkov zadolžen protokol »protobuf«, ki ga Google uporablja pri skoraj vseh internih RPC protokolih in datotečnih formatih. Grafika se kodira z uporabo odprtega video VP8 formata.
Če na kratko povzamemo, je Chrome Remote Desktop varna, zanesljiva »peer-to-peer« povezava, kjer se na strani gostitelja dogaja kodiranje v VP8 formatu, medtem ko je na strani odjemalca le enostaven predvajalnik VP8 formata.

2.5 Problematika protokolov za oddaljen dostop

Kljub temu, da so protokoli za prikaz oddaljenega namizja v uporabi že desetletja in jih proizvajalci stalno izboljšujejo, se bodo vedno pojavljali določeni problemi in omejitve, ki jih sprva protokoli ne bodo uspeli odpraviti. Sčasoma se pojavijo popravki ali izboljšave, vendar se tehnologija razvija naprej in tako lahko zopet naletimo na kakšno nepodprto napravo. V tem delu diplomskega dela bomo omenili nekaj najpogostejših težav, ki se pojavljajo pri uporabi protokolov za oddaljen dostop.

[6] Zbirka odprtokodne C++ kode za izgradnjo »peer-to-peer« aplikacij, več uporabniških glasovnih klepetov, video konferenc in predvajanje glasbe v živo. Povezava se lahko vzpostavi skozi NAT, požarne zidove, »relay« in »proxy« strežnike.

2.5.1 Pasovna širina in zakasnitve

S posvojitvijo VDI tehnologije v večjih podjetjih se je žal izkazalo, da le-ta še ni popolnoma zrela in ni izpolnila pričakovanj tako iz uporabniškega, kot tudi prodajnega vidika. Razlog tiči v razširljivosti (angl. *scalability*), vendar ne v tradicionalnem pomenu, kjer se izraz nanaša na izdelavo večjih, bolj zmogljivih sistemov. Pri VDI tehnologiji je problem razširljivosti v oddaljenih lokacijah, podružnicah in mobilnih delavcih. Ta horizontalna rast uvaja veliko spremenljivk, ki so ključnega pomena za preživetje VDI tehnologije, saj vplivajo tudi na zmogljivost in odzivnost uporabniške izkušnje.

Pri VDI tehnologiji komponente kot so zasnova omrežja in strežnikov, zmogljivost hipervisorja, gostota virtualnih naprav, protokoli za oddaljeno prikazovanje in pasovna širina, vplivajo na njeno zmogljivost. Administratorji lahko z lokalno postavitvijo pridobijo nekaj nadzora in možnosti prilagajanja teh komponent za čim boljšo uskladitev in zmogljivost VDI. Povsem nekaj drugega je, ko VDI zapusti LAN okolje in se premakne v WAN okolje, kjer administratorji nimajo več pristojnosti.

Nepredvidljiva zmogljivost WAN povezave, visoke zakasnitve in omejen pretok so ključni elementi, na katere je treba biti pozoren, kadar želimo dostaviti virtualna namizja preko WAN povezave. Prav na tem področju pa lahko protokoli za oddaljeno prikazovanje pokažejo svojo učinkovitost in zasluge. Vsak protokol sprejema kompromise med kakovostjo slike in odzivnostjo glede na prepustnost povezave (slika 2), vendar vsak na svoj način.

RDP protokol ni optimiziran za WAN povezave oz. povezave z visokimi zakasnitvami, predvsem ko govorimo o osveževanju zaslona za potrebe video vsebine. Kljub temu, da zavzema 70% pasovne širini za potrebe prenosa grafičnih podatkov in 30% za virtualne kanale (npr. za USB podatke), je še vedno zelo občutljiv na velike zakasnitve. To dejstvo je privedlo do razvoja tehnologij različnih proizvajalcev, z namenom pohitriti delovanje RDP protokola. Le-ti so rešitve videli v predpomnjenju, ignoriranju podvojenih podatkov (angl. *de-duplication*) in bolj efektivnem šifriranju [5].

Rešitev pospeševanja se lahko pojavi kot posrednik povezav (angl. *connection broker*), ali WAN optimizacija. Če torej želimo uporabljati RDP preko slabših WAN povezav, je običajno potrebno poseči po pospeševalcih, ki so rešitve drugih podjetji.

PCoIP protokol deluje nekoliko drugače in ima že vgrajene tehnologije, ki izboljšajo uporabniško izkušnjo preko slabih WAN povezav. PCoIP je zasnovan za prepoznavo tipa vsebine in glede na tip določi najbolj primeren kodek za stiskanje. PCoIP uporablja progresivno sestavljanje zaslona na odjemalčevi strani. V prvi fazi je na oddaljen zaslon poslana zelo kompresirana prosojna slika, ki pa je dostavljena v zelo kratkem času, ne glede na pasovno širino. Nato se slika progresivno izboljšuje glede na prepustnost povezave. Na koncu je slika vedno identična izvirni, brez izgub (s predpostavko da zaslon miruje nekaj časa). Razne pospešitve drugih proizvajalcev na PCoIP protokol ne vplivajo.

2.5.2 Lokalno tiskanje

Tiskanje na lokalni tiskalnik, priklopljen direktno na odjemalca (PC ali lahki odjemalec), je nedolgo nazaj predstavljalo dokaj veliko težavo v povezavi s tiskanjem iz oddaljenega namizja, predvsem v virtualnih namizjih. Če smo imeli USB tiskalnik, smo rešitev našli v USB preusmeritvi, vendar smo v tem primeru morali imeti nameščene gonilnike, tako na fizični napravi kot tudi v virtualnem okolju. V primeru dveh različnih operacijskih sistemov smo potrebovali dva različna gonilnika, kar je na nek način vseeno predstavljalo rešitev, medtem ko smo bili v primeru LPT tiskalnika prepuščeni sami sebi. Danes nam proizvajalci ponujajo različne rešitve, ki nam olajšajo delo s tiskalniki.

Microsoft nam ponuja rešitev Easy Print, ki je »proxy« za vsako interakcijo s tiskalnikom. Vse, kar je v povezavi s tiskalnikom, se preusmeri na odjemalčev tiskalnik, brez potrebe po namestitvi dodatnih gonilnikov v virtualnem okolju ali strežniku TS. Ta rešitev omogoča preusmeritve kateregakoli tiskalnika (LPT, COM, USB), ki je lokalno priklopljen na odjemalca ali na gostitelja. Prednost takega načina je tudi manjša poraba pasovne širine, saj se pri standardnem TS tiskanju velikost povečuje glede na število kopij. Obratno pa se pri rešitvi Easy Print prenese samo ena kopija, ki se stiska tolikokrat, kolikor je bilo postavljenih zastavic.

Po drugi strani nam VMware ponuja rešitev ThinPrint, ki je dodana View tehnologiji. Rešitev je zelo podobna že prej omenjeni. Ko namestimo View Client-a na odjemalca, zraven dobimo ».print Client«, zraven View Agent-a pa dobimo na virtualnem namizju ».print Engine«. To sta ključni komponenti rešitve ThinPrint. Komponenta ».print Engine« je sestavljena iz 3 delov, ThinPrint AutoConnect, ki komunicira s komponento ».print Client« in naredi sejo za tiskalnike, ThinPrint Output Gateway, ki je generičen gonilnik za tiskalnike za 32 in 64 bitne sisteme, in ThinPrint Port Monitor, ki je odgovoren za kompresijo in prenos podatkov in ki lahko uporabi tako RDP, kot tudi PCoIP protokol [11].

Komponenta ».print Client« priskrbi informacije o lokalnih tiskalnikih in njihovih nastavitvah ter jih posreduje storitvi ThinPrint AutoConnect. Ko uporabnik zahteva tiskanje, bo aplikacija podala zahtevo ThinPrint Output Gatway gonilniku, ta pa bo zgeneriral EnhancedMetaFile podatke. Podatki bodo nato obdelani in preneseni s storitvijo ThinPrint Port Monitor do odjemalca oz. komponente ».print Client«, kjer jih bo ta postavila v lokalni sistem tiskanja.

2.5.3 USB preusmeritev

Težave predstavlja predvsem hitrost prenosa podatkov iz USB spominskih ključev. Standard USB 2.0 omogoča prenose do 480Mb/s (60MB/s), standard USB 3.0 omogoča prenose do 5Gb/s (640MB/s). Lokalno omrežje omogoča hitrosti do 100Mb/s ali 1Gb/s, medtem ko prostrano WAN omrežje navadno omogoča hitrosti 1-10Mb/s. Že iz teh podatkov nam

postane jasno, da tudi pri idealnih pogojih ne bi morali zagotoviti prave hitrosti prenosa podatkov iz USB naprav do oddaljenega namizja.
Pri pametnih čitalcih kartic je pomembna hitra odzivnost oz. čim manjši čas obhoda (angl. *RTT*). Čitalci so USB naprave, kjer je obhodni čas manjši od 1ms in zato pričakujejo odgovor v podobnem času. Zaradi varnostne politike lahko čitalci ob prevelikem obhodnem času podatke tudi zavrnejo, in tukaj nastane problem. Prav tako pa problem predstavlja tudi neprepoznavnost nekaterih USB naprav.

2.5.4 Podpora za več monitorjev

Protokol PCoIP podpira več hkratnih monitorjev, ki delujejo vsak zase ne glede na verzijo operacijskih sistemov. RDP je do verzije 7.0 podpiral samo razpon (angl. *spanning*) monitorjev, v praksi pa je to pomenilo eno namizje, razpotegnjeno čez dva ekrana, ki nista delovala kot samostojna monitorja vsak z svojo resolucijo. V tabeli 8 je prikazan pregled različnih verzij orodja RDC ter njihova podpora za uporabo več hkratnih monitorjev.

Tabela 8: Večzaslonska podpora pri uporabi RDC.

na Win7/R2 iz:	Win7/R2	Vista SP1	Vista SP1	XP SP3	XP SP3	XP SP2	XP SP2
z uporabo RDC	7.0	7.0	6.1	7.0	6.1	6.1	5.2
več monitorjev	DA	DA	razpon	DA	razpon	razpon	NE

2.5.5 Večpredstavnost

Protokol PCoIP podpira večpredstavnost ne glede na verzijo operacijskih sistemov. V tabeli 9 je prikazana podpora za dvosmerni zvok in aero vmesnik. Podpora za večpredstavnost pri uporabi RDC pride v veljavo v primeru uporabe ene izmed spodaj naštetih verzij programskega orodja RDC, če se povezujemo na napravo z operacijskim sistemom Windows 7 ali Windows Server 2008 R2.

Tabela 9: Podpora za večpredstavnost pri uporabi RDC.

na Win7/R2 iz:	Win7/R2	Vista SP1	Vista SP1	XP SP3	XP SP3	XP SP2	XP SP2
z uporabo RDC	7.0	7.0	6.1	7.0	6.1	6.1	5.2
dvosmerni zvok	DA	DA	NE	DA	NE	NE	NE
AERO vmesnik	DA	NE	NE	NE	NE	NE	NE

3 PRIMERJAVA PROTOKOLOV V PRAKSI

V današnjih produkcijskih okoljih večjih korporacij, kjer je nadzor in upravljanje večjega števila računalnikov ključnega pomena, se kot rešitev vedno bolj izpostavlja VDI. S stališča končnega uporabnika se glede na dosedanje računalniško okolje spremeni zelo malo, v primeru da zahteve niso prevelike ali specifične. Zato pa VDI pomeni veliko spremembo za upravitelje sistemov, saj jim močno olajša delo pri pripravi, razdeljevanju in vzdrževanju tovrstnega okolja [4]. S tehnologijo VDI lahko hitro in le z nekaj kliki zagotovimo novo virtualno namizje za novega uporabnika ali pa zamenjamo okvarjeno obstoječe namizje s svežo kopijo. Virtualna namizja delujejo na strežnikih z visoko zanesljivostjo, prav zato pa je za njihovo brezhibno delovanje in varnost bolje poskrbljeno kot pri navadnih osebnih računalnikih.

Pri tehnologiji VDI odjemalci (programski ali strojni) dostopajo do virtualnega namizja v podatkovnem centru. Ključno nalogo pri takšnem načinu dostopa omogočajo protokoli za oddaljeno namizje, ki virtualno namizje prikažejo na zaslonu odjemalca. Takšnih protokolov je veliko, zato se bomo v diplomskem delu bolj podrobno posvetili le dvema, ki igrata veliko vlogo pri VDI rešitvah. To sta protokola Microsoft RDP in VMware PCoIP. V nadaljevanju bomo omenili komponente, ki so potrebne za postavitev VDI okolja tako z Microsoft Hyper-V kot tudi VMware View arhitekturo. Prikazali bomo tudi konkretni postavitvi POC (angl. *proof of concept*) obeh proizvajalcev za potrebe testiranja ter primerjanja zmogljivosti obeh omenjenih protokolov.

Če želimo postaviti pravo VDI okolje, moramo začeti pri virtualizaciji fizičnega strežnika oz. njegovih komponent. Namen virtualizacije je ločiti strojno opremo od programske oz. odstraniti njuno neposredno odvisnost. To vlogo bomo zaupali hipervisorju imenovanemu tudi VMM (angl. *virtual machine manager*), ki uporablja strojno virtualizacijo za potrebe deljenja fizičnih komponent (procesor, delovni spomin, itd.) med več gostujočimi operacijskimi sistemi. Takšna hipervisorja sta Microsoft *Hyper-V* in VMware *ESXi*, ki spadata pod kategorijo tipa »*bare metal*«. Ta kategorija je razširjena predvsem v strežniških sistemih, saj poteka direktno na strežniški strojni opremi in je zato tudi odgovorna za kontrolo le-te. Poleg tega pa upravlja še z gostujočimi operacijskimi sistemi, ki potekajo na višjem nivoju, na hipervisorju.

Na strani fizičnega strežnika tokrat ne bomo gostili samo virtualnih strežnikov (kar je praksa že skoraj v vsakem podatkovnem centru), pač pa tudi navidezne računalnike z operacijskim sistemom, tako imenovana virtualna namizja. V rešitvah VDI poleg strežnikov, ki gostijo

virtualna namizja, potrebujemo še dodatne strežnike, ki posredujejo seje med odjemalci in virtualnimi namizji ter strežnike, ki omogočajo pripravo novih namizij. V prvem primeru je vloga strežnika ta, da zahtevke posameznega odjemalca preusmeri na točno določeno virtualno namizje ali pa na prvo prosto (odvisno od implementacije dodeljevanja, ki je lahko statično ali dinamično). Takšno vrsto strežnika globalno imenujemo posrednik (angl. *broker*), medtem ko ga Microsoft imenuje *Remote Desktop Connection Broker*, VMware pa *View Connection*. V drugem primeru pa strežnik sistemskemu administratorju omogoča spremljanje delovanja celotne VDI rešitve ter pripravo novih virtualnih namizij – bodisi samodejno na zahtevo, bodisi ročno s strani administratorja. Strežnika takšne narave sta *Remote Desktop Virtualization Host*, ki temelji na infrastrukturi Microsoft *Hyper-V* ter *VMware vCenter*, temelječ na infrastrukturi VMware *vSphere*.

Za lažje razumevanje VMware infrastrukture moramo razložiti dve različni komponenti, ki pa sta tesno povezani med seboj. To sta VMware *View* in *vSphere*. VMware *vSphere* infrastruktura je sestavljena iz različnih komponent, ki spremenijo standardno fizično strežniško infrastrukturo v strežniško infrastrukturo, ki ponuja deljene vire za virtualne naprave. V VMware *vSphere* programsko opremo so vgrajeni tudi mehanizmi za visoko razpoložljivost in nadzor aplikacij. VMware *View* je celovita rešitev za upravljane VDI okolja z vgrajenim sistemom za dostavo varnega namizja uporabnikom. VMware *View* je zgrajen na platformi VMware *vSphere* in omogoča orodja za upravljanje navidezne namizne infrastrukture ter vsebuje tehnologije, ki zagotavljajo bogato uporabniško izkušnjo.

Tako Microsoft kot tudi VMware za VDI infrastrukturo ponujata poleg že omenjenih strežnikov še dodatne strežnike za lažje upravljanje virtualnih namizij (*System Center Virtual Machine Manager* in *View Composer*) za dostop do virtualnih namizij preko interneta (*Remote Desktop Gateway* in *View Security*). Microsoft ponuja še strežnik Remote Desktop Web Access, ki je nekakšne vrste spletni portal, na katerem so objavljene aplikacije RemoteApp in virtualna namizja, do katerih lahko uporabnik dostopa preko spleta. VMware dodatno ponuja še *View Transfer* strežnik, ki omogoča mobilnim uporabnikom t.i. »*off-line*« VDI. V praksi to pomeni, da se virtualno namizje iz podatkovnega centra prenese na prenosnik. Posledično je to namizje uporabno tudi kadar nismo povezani s službenim omrežjem. Ko se vrnemo in priklopimo v službeno omrežje, *View Transfer* strežnik poskrbi, da se spremembe virtualnih namizij sinhronizirajo.

3.1 Postavitev testnega okolja na VMware View 5 infrastrukturi

Za namestitev hipervisorja VMware ESXi 5 je potreben 64-bitni procesor z vsaj dvema jedroma (novejši procesorji tipa AMD Opteron in Intel Xeon), vsaj 2 GB delovnega

pomnilnika ter trdi disk, ki je lahko priklopljen lokalno ali preko omrežja. Zaželena je tudi podpora za strojno virtualizacijo (AMD-V in Intel VT-x).

V našem testnem okolju smo ESXi namestili na delovno postajo z dvojedrnim 64-bitnim procesorjem AMD Athlon 64 X2 5000+, ki podpira strojno virtualizacijo AMD-V. Delovna postaja je vsebovala 8GB DDR2 delovnega spomina, trdi disk WD Reptor 32GB z 10.000 obrati na minuto, negibljivi disk Crucial M4 SSD 120GB ter štiri mrežne kartice hitrosti 1000Mb. Glede na konfiguracijo smo zadostili osnovnim zahtevam za namestitev VMware ESXi. Žal se je v praksi izkazalo, da EXSi ne podpira strojne virtualizacije uporabljenega procesorja, zato bi lahko prišlo v testih do nekoliko slabših rezultatov. Funkcionalnosti so ostale neokrnjene.

V prvem koraku smo namestili VMware ESXi 5 na WD disk ter nastavili IP naslov primarne mrežne kartice. Na prenosniku smo v brskalnik vnesli IP naslov ESXi strežnika in prenesli programsko orodje *vSphere Client*, ki je namenjeno za dostop in upravljanje EXSi strežnikov. Povezali smo se na ESXi strežnik in ustvarili prvo virtualno napravo, na katero smo namestili operacijski sistem Windows Server 2008 R2 SP1. Temu strežniku smo dodali vlogo aktivnega imenika oz. AD (angl. *active directory*) in sistem domenskih imen oz. DNS (angl. *domain name system*). Ker strežniki vsebujejo podatke o uporabnikih in računalnikih v domeni ter njihovih pravilih in pravicah, smo v tem koraku ustvarili nov uporabniški račun.

V drugem koraku smo na strežnik ESXi dodali novo podatkovno shrambo (angl. *data storage*) s SSD diskom velikosti 120GB. V to podatkovno shrambo smo namestili še tri nove virtualne naprave, dvakrat z operacijskim sistemom Windows Server 2008 R2 SP1 in enkrat Windows 7 Enterprise SP1. Na prvi Windows strežnik smo namestili VMware *vCenter*, ki zahteva dva virtualna procesorja, 4GB delovnega pomnilnika in velikost diska 40GB. VMware *vCenter* je programska oprema, ki omogoča centralizirano upravljanje navidezne infrastrukture. Omogoča podroben vpogled v vse vidike navidezne infrastrukture. Z možnostjo avtomatizacije infrastrukture se olajša delo dnevnih nalog, s katerimi se srečuje administrator okolja. Maksimalno število virtualnih naprav, ki jih lahko upravlja en VMware *vCenter* strežnik, je 2000.

Na drugi Windows strežnik smo namestili *View Connection*, ki zahteva 4 virtualne procesorje, 10GB delovnega pomnilnika in 40GB trdega diska. *View Connection* je posrednik uporabniških sej in omogoča administracijo navideznih namizij s pomočjo spletnega portala. Posrednik uporabniških sej omogoča končnim uporabnikom dostop do navideznih namizij. Maksimalno število RDP ali PCoIP sej, ki jih lahko upravlja en *View Connection* strežnik, je 2000.

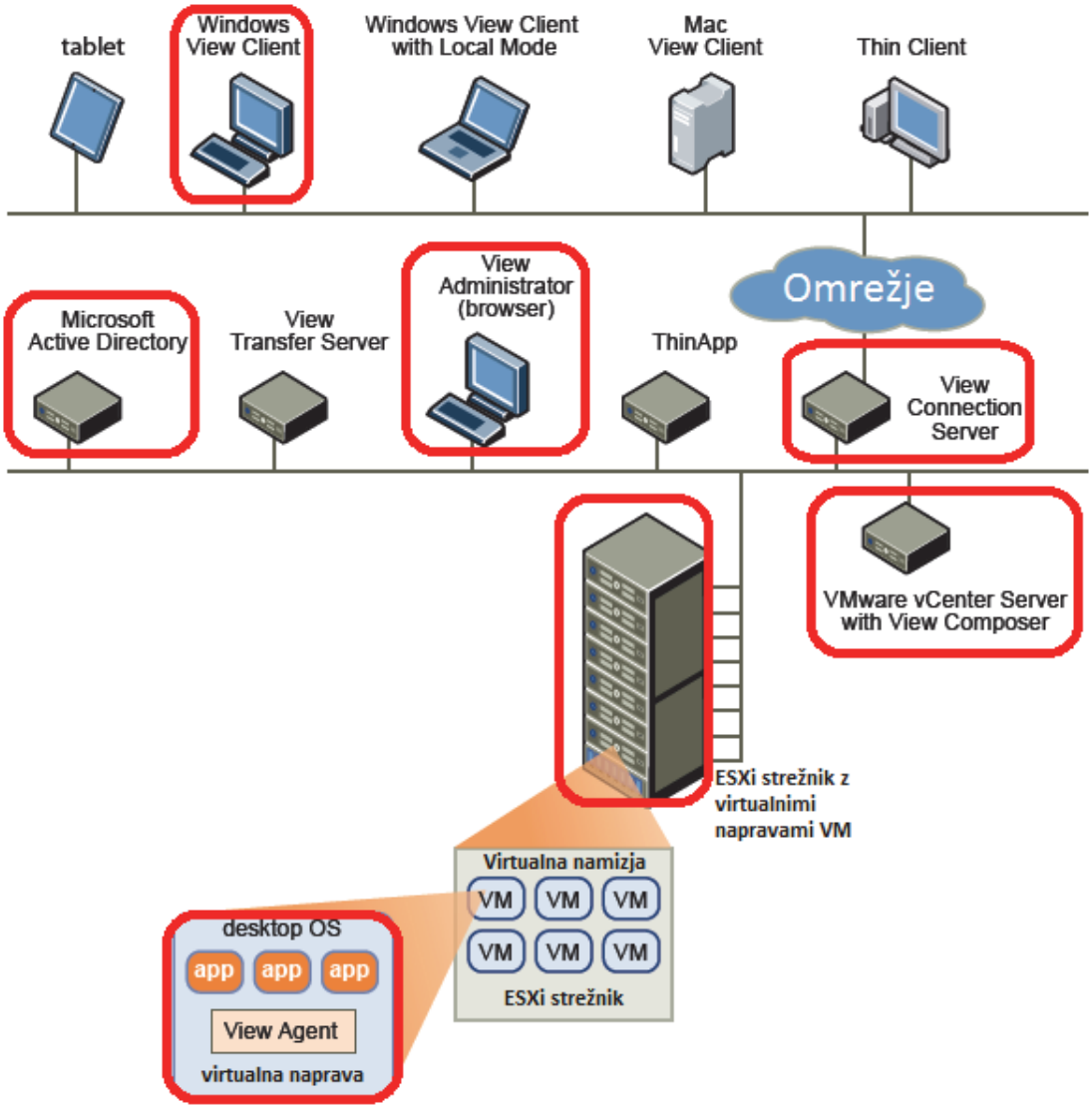
Obe namestitvi lahko opravimo tudi na slabši konfiguraciji, saj so te zahteve namenjene za produkcijsko okolje in so za našo tesno okolje predimenzionirane. Zgolj za informacijo podajmo, da je število virtualnih naprav v VMware *View 5* okolju omejeno na 16 naprav na eno procesorsko jedro. Ker nam virtualizacija omogoča dodeljevanje računalniških virov, (kljub temu da nam le-ti niso na voljo v takšni meri), smo za čas namestitve uporabili priporočeno konfiguracijo. To smo storili zato, ker se nam namestitev na manj zmogljiv sistem samodejno nastavi in omeji določene nastavitve. Po namestitvi smo konfiguracijo virtualnih naprav zmanjšali glede na fizične zmogljivosti našega sistema. Virtualni napravi z Windows 7 smo dodelili 1 virtualni procesor in 1GB delovnega spomina, na njo pa smo namestili storitev *View Agent*, ki je potrebna za uspešno komunikacijo s posrednikom *View Connection*.

Končna konfiguracija virtualnih naprav je razvidna iz tabele 10. Preko spletnega vmesnika smo se povezali na *View Connection*, kjer smo ustvarili nov bazen (angl. *pool*) za virtualna namizja ter dodali virtualno napravo z operacijskim sistemom Windows 7 in nameščenim *View Agent*. Tej virtualni napravi smo dodeli še uporabnika, ki smo ga ustvarili na AD strežniku.

Tabela 10: Konfiguracija virtualnih naprav za infrastrukturo vSphere.

ESXi / AMD X2 5000+ / 8GB RAM / HDD 32GB 10.000 / SSD 120GB				
OS	Server 2008 R2	Server 2008 R2	Server 2008 R2	Windows 7
vCPU	1	2	2 (4)	1
RAM (GB)	1	2 (4)	2 (10)	1
disk (GB)	20	20 (40)	20 (40)	12
vloga	AD, DNS	VMware vCenter	View Connection	View Agent

Testno okolje z VMware *vSphere* infrastrukturo je bilo pripravljeno. Vse kar smo morali še storiti je bila namestitev aplikacije *VMware View Client* na odjemalca, ki je bil v našem primeru prenosnik. Instalacijo smo dobili na IP naslov strežnika *View Connection* (ali preko FQDN), če smo ga vnesli v brskalnik. Pri prvem zagonu aplikacije *View Client* smo vnesli FQDN ali IP naslov strežnika *View Connection*, nato smo vnesli uporabniške podatke in kot odgovor dobili vsa dodeljena virtualna namizja. Izbrali smo velikost zaslona ter protokol (RDP ali PCoIP) in po kliku na gumb »poveži« v nekaj trenutkih dobili dostop do polno funkcionalnega oddaljenega namizja. Vse nameščene komponente, potrebne za postavitev VMware View okolja, so za lažjo predstavo prikazane v rdečih okvirjih na sliki 8.



Slika 8: Komponente VMware View infrastrukture.

3.2 Postavitev testnega okolja na Microsoft Hyper-V infrastrukturi

Postavitev testnega okolja za Microsoft ni bila tako obširna kot pri VMware, zadostovala je namreč že namestitev Windows Server 2008 R2 z vlogama hipervisorja Hyper-V ter Remote Desktop Virtualization Host. Pri slednji smo vklopili še opcijo RemoteFX, ki naj bi končnemu uporabniku omogočala poganjati 3D aplikacije znotraj virtualnega namizja. Vendar pa se je ravno zaradi te funkcije sprva lahka postavitev zakomplicirala. Namestitev hipervisorja Hyper-V na isti delovni postaji, kot je VMware okolje, je potekala brezhibno, prav tako tudi namestitev strežnika Windows in nato virtualne naprave z Windows 7 ni bila problematična. Težave so se pojavile, ko smo želeli vklopiti RemoteFX, in jih nismo uspeli rešili drugače kot z drugo strojno opremo. Razlog tiči v strojni podpori procesorja, ki mora

podpirati funcijo SLAT (angl. *second-level address translation*), ki jo Intel imenuje EPT (angl. *extended page tables*), AMD pa NPT (angl. *nested page tables*) ali RVI (angl. *rapid virtualization indexing*). Žal te funkcije podpirajo le najnovejši procesorji, zato smo bili primorani postaviti Microsoftovo testno okolje na prenosniku z dvojedrnim procesorjem Intel i5-2410M, 8GB DDR3 delovnega pomnilnika, 320GB trdega diska ter grafično kartico AMD Radeon HD 6630M. Grafična kartica je ključnega pomena pri uporabi RemoteFX tehnologije, saj njena zmogljivost določa število zaslonov oz. virtualnih naprav, ki bodo lahko uporabljali tehnologijo RemoteFX. Koliko pomnilnika si mora grafična kartica rezervirati glede na število zaslonov je prikazano v tabeli 11. Trenutno ena grafična procesna enota podpira do 12 virtualnih naprav.

Tabela 11: Poraba grafičnega pomnilnika glede na število zaslonov.

Maksimalna resolucija	**Maksimalno število zaslonov**			
	1 zaslon	**2 zaslona**	**3 zasloni**	**4 zasloni**
1024x768	75 MB	105 MB	135 MB	165 MB
1280x1024	125 MB	175 MB	225 MB	275 MB
1600x1200	184 MB	257 MB	330 MB	/
1920x1200	220 MB	308 MB	/	/

Ko smo imeli potrebno strojno opremo, je bila namestitev hipervisorja Hyper-V zelo enostavna. Microsoftov hipervisor smo nato upravljali preko programskega orodja Server Manager, kjer smo ustvarili novo virtualno napravo in nanj namestili operacijski sistem Windows 7. Za ustvarjeno virtualno napravo smo vklopili še opcijo RemoteFX, in poenostavljeno Microsoft VDI okolje je bilo postavljeno. Po večdnevnih začetnih težavah smo imeli postavljeni obe testni okolji in meritve zmogljivosti oddaljenih protokolov RDP 7 in PCoIP so se začele.

3.3 Zmogljivostna analiza

Na zmogljivost in odzivnost VDI tehnologije vpliva več komponent, kot so zasnova omrežja in strežnikov, zmogljivost hipervisorja, gostota virtualnih naprav, protokoli za oddaljeno prikazovanje in pasovna širina. V lokalnem omrežju lahko administrator z optimizacijo in s prilagajanjem teh komponent ustvari zmogljiv in odziven VDI sistem. Povsem nekaj drugega pa je situacija, ko se administrator znajde v WAN omrežju, kjer nima več pristojnosti.
Nepredvidljiva zmogljivost WAN povezave, visoke zakasnitve in omejen pretok so ključni elementi, na katere je treba biti pozoren, kadar želimo dostaviti virtualna namizja preko WAN povezave. Prav na tem področju pa lahko protokoli za oddaljeno prikazovanje pokažejo svojo učinkovitost in zasluge. Vsak protokol sprejema kompromise med kakovostjo slike in

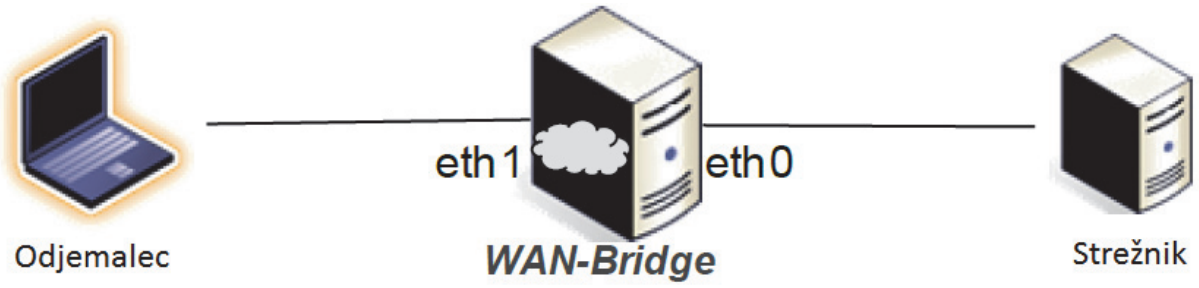
odzivnostjo glede na prepustnost povezave (slika 2), vendar vsak na svoj način. Kako pa se ti različni načini obnesejo v praksi, nas je zanimalo v praktičnem delu našega diplomskega dela. Z analizo zmogljivosti oddaljenih protokolov smo želeli pridobiti čim bolj realne podatke o odzivnosti in zahtevnosti posameznega protokola, zato smo za metodo analize izbrali meritve. Kljub temu, da je izbrana metoda veliko zahtevnejša od simulacije in analitičnega modela, smo se za njo odločili, saj je verodostojnost tako pridobljenih podatkov večja. Pri analizi zmogljivosti opazovanega računalniškega sistema nas zanima, kako se sistem obnaša pod različnimi bremeni.

»*Breme je množica vhodnih zahtev, ki obremenjujejo strežniške sposobnosti opazovanega računalniškega sistema. Računalniško breme je neposredno odvisno od realnega bremena, ki ga povzroča uporabnik. Realno breme je odvisno od storitve, ki jo računalniški sistem nudi uporabniku* [1]«. Realno breme je v primeru oddaljenih protokolov predstavljajo različne aplikacije (ki jih zaganja uporabnik) oz. njihovo upodabljanje na strani odjemalca. Z vidika oddaljenega protokola se aplikacije razlikujejo predvsem po intenzivnosti osveževanja in velikosti slikovnega medpomnilnika. Za primer podajmo pisanje teksta v aplikaciji Microsoft Word in predvajanje video vsebine v namenski aplikaciji. Protokol osvežuje le delček slikovnega pomnilnika, kjer se pojavljajo črke, intenzivnost osveževanja pa ne igra bistvene vloge. Na drugi strani pa predvajanje video vsebine zahteva hitro osveževanje celotnega območja videa. Intenzivnost in velikost slikovnega pomnilnika, potrebnega za osveževanje, se odraža v porabi pasovne širine.

»*Testno breme je množica vhodnih zahtev, ki se uporabljajo pri meritvah s ciljem pridobitve podatkov za analizo zmogljivosti računalniških sistemov. S testnim bremenom obremenimo sistem v smislu strežbe in opazujemo njegovo obnašanje v tem stanju. Opazovanje razumemo predvsem kot merjenje vseh pomembnih parametrov sistema, ki se ob prisotnosti bremena spremenijo. Testno breme je lahko realno ali pa model bremena* [1]«. Ker je realno breme odvisno od realnega stohastičnega okolja in s tem neponovljivo, je analiza zmogljivosti v takšnem primeru težko izvedljiva. Iz tega razloga smo se za potrebe našega diplomskega dela odločili za model bremena, ki omogoča ponavljanje meritev pri različnih parametrih sistema.

Modeli bremena so lahko sintetični ali umetni. Umetna bremena so sestavljena iz podatkov o realnem bremenu, pridobljenih na osnovi domnev, sintetična bremena pa so pridobljena na osnovi meritev. Za izvedbo meritev smo izbrali tako osnovna sintetična bremena kot tudi hibridna sintetična bremena [1]. Pri osnovnih sintetičnih bremenih smo meritve izvajali pri enem samem bremenu (eni aplikaciji) istočasno. Osnovna sintetična bremena v naših meritvah so bile aplikacije Internet Explorer, Microsoft Word, Acrobat Reader ter Windows Media Player. Hibridno statično breme je predstavljalo namenski program za meritve (angl.

benchmark) GUIMark, ki meri sposobnosti upodabljanja spletnih brskalnikov. Ker smo izbrali model bremena in ne realno breme, smo lahko meritve z istim modelom bremena poljubnokrat ponovili z različnimi parametri. Poglavitni parametri, ki smo jih uporabili med meritvami, so bile različne hitrosti povezave, saj le-te igrajo ključno vlogo v oddaljenih protokolih. V testu smo uporabili povezave pasovne širine 1Gb/s za potrebe LAN okolja, ter 2Mb/s in 1Mb/s za potrebe WAN okolja. Testi so bili zastavljeni tako, da so poskušali ustvaritvi čim bolj realne pogoje v različnih situacijah, zato sta bila v primeru povezav hitrosti 2Mb/s in 1Mb/s simulirana tudi čas obhoda RTT (angl. *round-trip time*) in izgubljeni paketki (angl. *packet loss*). Simulacije povezav smo dosegli z WAN emulatorjem *WAN-Bridge* z dvema mrežnima karticama, ki smo ga priklopili na omrežje pred odjemalcem oz. prenosnikom, kot kaže slika 9. Naloga emulatorja je bila simulirati omrežje, ki je predstavljeno na sliki 8. Povezava z 2Mb/s pasovne širine je bila simulirana z obhodnim časom 40ms RTT in deležem izgubljenih paketkov 0.001%, medtem ko je bila povezava hitrosti 1Mb/s simulirana z obhodnim časom 100ms in deležem izgubljenih paketkov 0.01 %.



Slika 9: Postavitev WAN emulatorja »wanbridge«.

Poleg obnašanja RDP in PCoIP protokola na povezavah z različno pasovno širino, smo primerjali tudi različne optimizacijske metode in podporo za 3D grafiko. Tudi ti parametri lahko močno vplivajo na porabo pasovne širine ter posledično tudi na uporabniško izkušnjo. Meritve smo izvajali z različnimi kombinacijami parametrov. Parametri pri protokolu PCoIP so bili vklopljena ali izklopljena funkcija BTL (angl. *built to lossless*), omejevanje maksimalne pasovne širine povezave in vklopljena ali izklopljena 3D funkcionalnost. Parametri pri protokolu RDP so bili vklopljena ali izklopljena 3D funkcionalnost, barvna globina ter kakovost slike.

Protokol RDP po privzetih nastavitvah porabi veliko pasovne širine, zato smo v primeru 2Mb in 1Mb WAN povezave izvajali meritev samo z optimiziranimi nastavitvami za počasnejše povezave. Ta optimizacija je vključevala izklop ozadja namizja, izklop glajenja pisav, izklop prikaza vsebine oken med vlečenjem ter izklop animacije menijev in oken. Prav tako je bila barvna globina omejena na 16-bitov. Preko lokalnih skupinskih politik (angl. l*ocal group*

policy) smo nastavili pravilo za stiskanje podatkov RDP z namenom, da protokol za prenos porabi čim manj pasovne širine.
Meritve smo izvajali tako pod različnimi parametri, kot tudi pod različnimi bremeni. Realno breme za oddaljene protokole predstavlja aplikacije oz. njihovo upodabljanje na strani odjemalca.

Za osnovni sintetični model bremena smo izbrali naslednje aplikacije:

- Windows Media Player za predvajanje 30 sekundnega HD posnetka, velikosti 1280x720, s hitrostjo 29 sličic na sekundo. Meritve smo izvedli v pomanjšanem (830x465) in v celozaslonskem (1680x1050) načinu predvajanja.
- Internet Explorer za predvajanje 360p video posnetka iz internetne strani YouTube.
- Microsoft Word, Acrobat Reader in Internet Explorer za simulacijo pisarniške uporabe.

Za hibridni sintetični model bremena smo izbrali naslednje aplikacije:

- GUIMark za zmogljivostni test upodabljanja tehnologije HTML v spletnem brskalniku Internet Explorer.
- GUIMark2 za zmogljivostni test upodabljanja tehnologije Flash v spletnem brskalniku Internet Explorer.

Za meritve smo uporabljali tri programske monitorje. Za merjenje porabe povprečne in maksimalne pasovne širine, za porabo centralno procesne enote na virtualnem namizju, za delež izgubljenih paketov ter za obhodni čas RTT, smo uporabili programski monitor Performance Monitor, ki je že vgrajen v operacijski sistem Windows. V kombinaciji z že prej nameščeno aplikacijo View Agent v virtualnem namizju nam Performance Monitor omogoča opazovanje vseh prej naštetih parametrov.
Drugi programski monitor Fraps je bil namenjen merjenju števila sličic na sekundo FPS, nameščen pa je bil na odjemalca oz. na prenosnik, ker nas je zanimala uporabniška izkušnja na strani končnega uporabnika.
Tretji programski monitor smo uporabili na strani hipervisorja za nadzor obremenitve centralno procesorske enote hipervisorja. Trajanje posameznega testa je bilo 30 sekund, zato so bili tudi monitorji nastavljeni tako, da so se po tridesetih sekundah ustavili.

3.4 Rezultati meritev

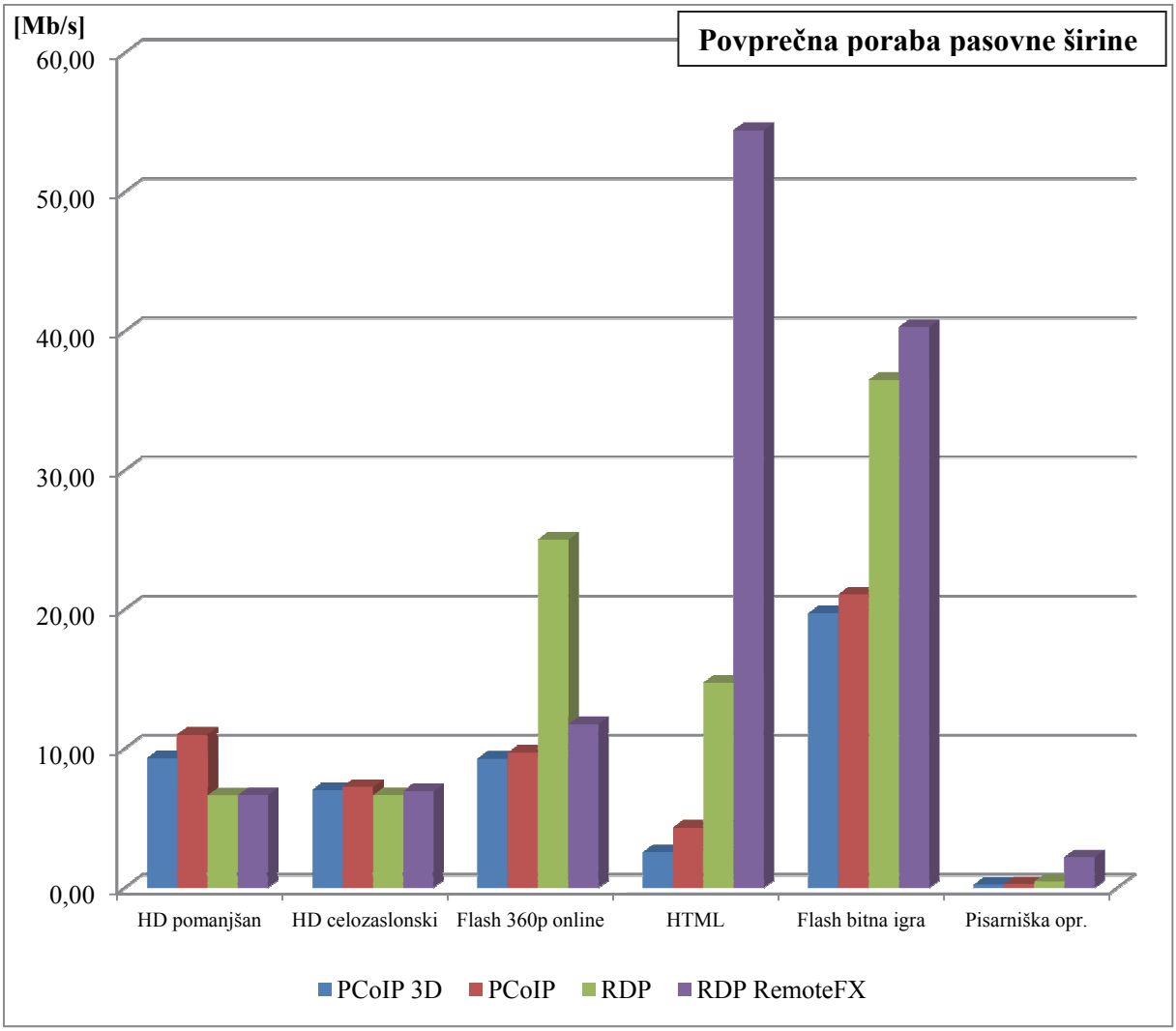
Sledeči rezultati testov so kombinacija različnih parametrov pod različnimi pogoji. Te kombinacije so sestavljene iz spodnjih treh razdelkov.

I. Povezave smo razdelili na:
- LAN 1Gb/s,
- WAN 2Mb/s, 40ms RTT in delež izgube paketov 0.001%,
- WAN 1Mb/s, 100ms RTT in delež izgube paketov 0.01%.

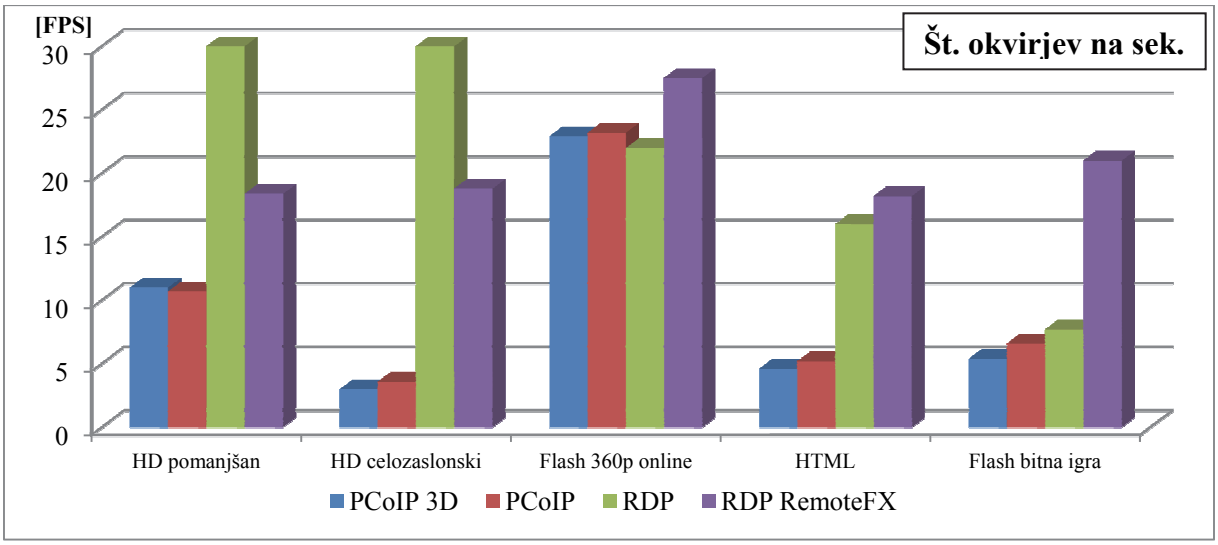
II. Različne nastavitve in optimizacija (v oklepajih so oznake na grafih):
- PCoIP z vklopljeno podporo 3D (*PCoIP 3D*),
- PCoIP z izklopljeno podporo 3D (*PCoIP*),
- RDP 7 z vklopljeno podporo 3D (*RDP RemoteFX*),
- RDP 7 z izklopljeno podporo 3D (*RDP*),
- RDP 7 za počasne širokopasovne povezave 256Kb/s – 2Mb/s (*RDP optimiziran*),
- PCoIP z izklopljeno podporo 3D in izklopljeno funkcijo BTL (*PCoIP / BTL off*).
- PCoIP z vklopljeno podporo 3D in izklopljeno funkcijo BTL ter omejitev maksimalne pasovne širine na 1900Kb/s ali 920Kb/s (*PCoIP 3D / BTL off / max 1900 ali max 920*),
- PCoIP z izklopljeno podporo 3D in izklopljeno funkcijo BTL ter omejitvijo maksimalne pasovne širine na 1900Kb/s (*PCoIP / BTL off / max 920*).

III. Različna bremena oz. aplikacije (v oklepajih so oznake na grafih):
- Windows Media Player s pomanjšanim videom (*HD pomanjšan*),
- Windows Media Player s celozaslonskim videom (*HD celozaslonski*),
- Internet Explorer z 360p videom na YouTube strani (*Flash 360p*),
- GUIMark zmogljivostni test HTML (*HTML*),
- GUIMark2 zmogljivostni test Flash bitna igra (*Flash bitna igra*),
- simulacija pisarniških opravil z aplikacijami Microsoft Word, Acrobat Reader in Internet Explorer (*Pisarniška opr.*).

Prve teste smo opravili na lokalni povezavi hitrosti do 1Gb/s. Ker smo tako imeli na razpolago dovolj pasovne širine, je bila optimizacija oddaljenega protokola odveč. Med seboj smo primerjali protokola RDP 7 in PCoIP, vsakega z vklopljeno in izklopljeno podporo za 3D. Prvi graf predstavlja povprečno porabo pasovne širine v Mb na sekundo, drugi graf pa predstavlja število okvirjev oz. slik na sekundo.



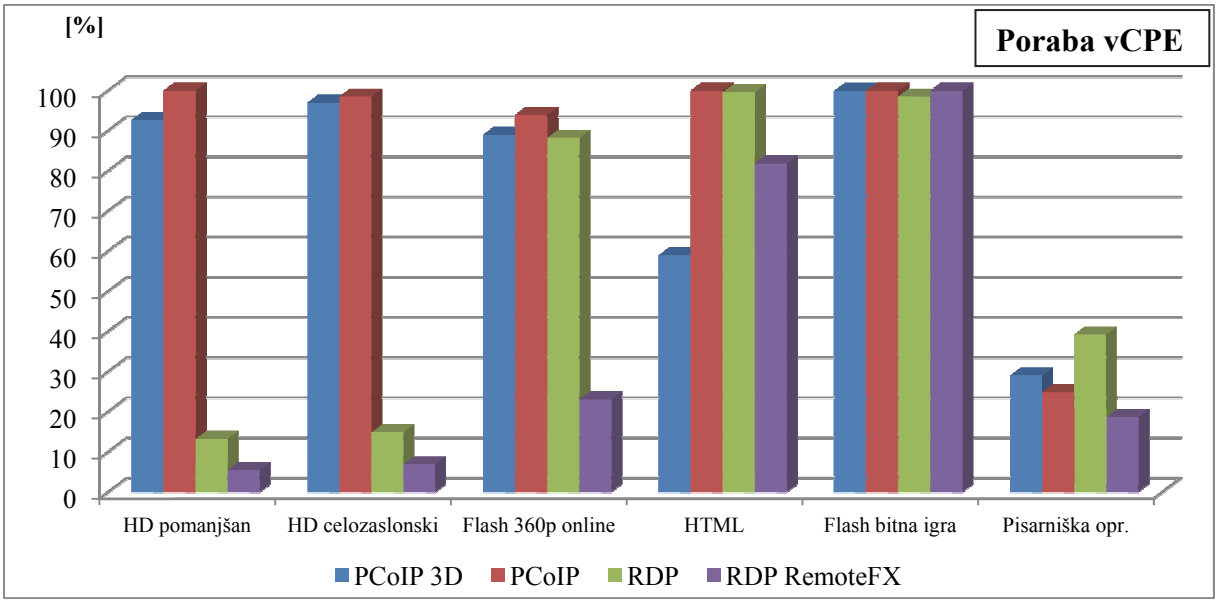
Slika 10: Povprečna porabe pasovne širine [Mb/s] preko povezave hitrosti 1Gb/s.



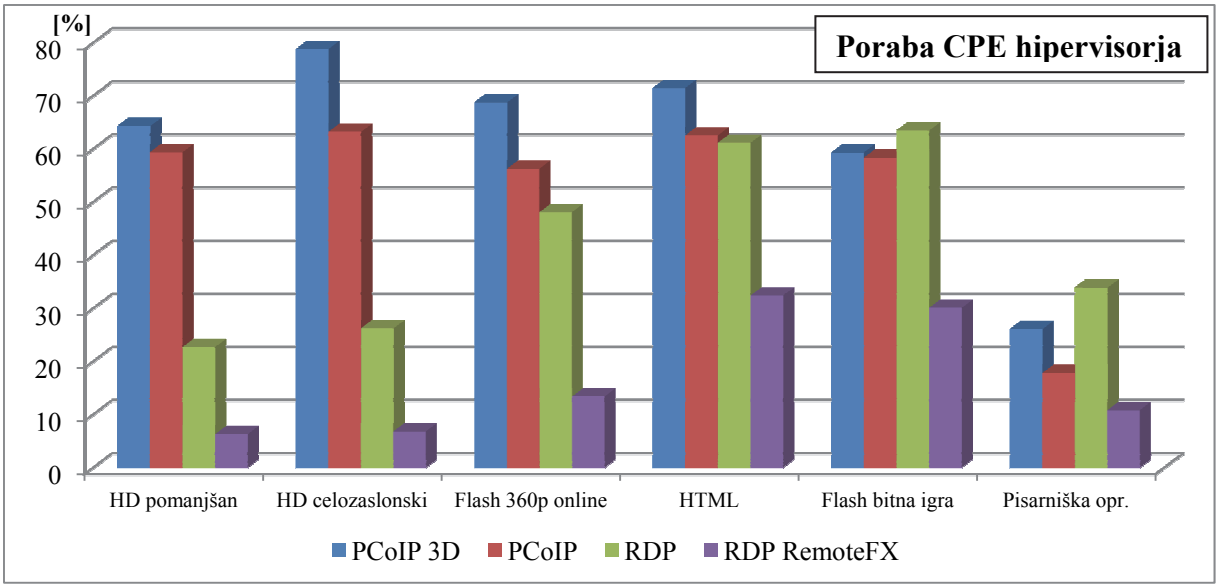
Slika 11: Št. okvirjev na sekundo [FPS] preko povezave hitrosti 1Gb/s.

Iz grafičnega prikaza na sliki 10 lahko razberemo, da protokol RDP 7 pri predvajanju videa v predvajalniku Windows Media Player (WMP) porabi manj pasovne širine tako v primeru vklopljene 3D funkcionalnosti, kot tudi brez nje. Razlog za to gre iskati v novi funkcionalnosti »Windows Media Player remoting«, ki omogoča tekoče predvajanje tako, da odjemalcu pošilja samo surove medijske podatke (angl. *raw data*), ki se nato upodobijo na odjemalcu. Podatki se shranijo v WMV datoteki, ki je nato poslana preko RDP protokola odjemalcu. Ta izvede dekodiranje, podatke upodobi in tako prihrani veliko pasovne širine. Po vrhu vsega je predvajanje posnetka zelo gladko, saj se praktično izvaja lokalno. To je razvidno iz grafa na sliki 11, kjer lahko opazujemo močno povečano število okvirjev na sekundo pri RDP protokolu, v primerjavi s PCoIP. Vendar pa v primeru uporabe katerega drugega predvajalnika, kot naprimer VLC, ta funkcionalnost ni podprta. Prav iz tega razloga se vsa video vsebina upodobi na strežniku in se pošlje v obliki bitnih slik preko UDP protokola odjemalcu, kar občutno poveča porabo pasovne širine [12].
Pri protokolu RDP je pasovna širina skoraj identična, ne glede na to, ali video predvajamo čez celoten zaslon ali pomanjšano, saj se v obeh primerih pošiljajo samo surovi podatki. V primeru celozaslonskega predvajanja pa je zanimiva zmanjšana poraba pasovne širine pri PCoIP protokolu. Razlog za to lahko najdemo na sliki 11, kjer je opazen močan upad števila okvirjev na sekundo, kar posledično zmanjša tudi porabo pasovne širine.
Prednost protokola PCoIP v primerjavi z RDP bi lahko bila manjša poraba pasovne širine pri vseh testih, vendar bo to ključnega pomena pri počasnih povezavah. Dober oddaljen protokol se prilagaja in zna izkoristiti toliko pasovne širine, kolikor mu jo je ponujeno [7]. Zanimiva je tudi poraba pasovne širine v primeru vklopljene podpore za 3D pri protokolu PCoIP, saj je le-ta manjša od porabe z izklopljeno podporo. Razlog za to gre iskati tudi v zmanjšanem število okvirjev na sekundo, kar je razvidno iz slike 11.
Za oddaljen dostop preko LAN povezave se je za uspešnejšega izkazal protokol RDP, saj je ponudil boljšo uporabniško izkušnjo. Predvsem se je izkazal RDP RemoteFX, saj je pri skoraj vseh testih prikazal največ okvirjev na sekundo, le v primeru predvajalnika WMP ne, ker so bili nekateri medijski podatki vseeno najprej upodobljeni na strani gostiteljeve grafične procesne enote. RDP RemoteFX je pred tekmeci izstopal predvsem pri tehnologiji Flash.

Graf na sliki 13 predstavlja porabo virtualne centralno procesne enote na virtualni napravi v procentih. Graf na sliki 14 predstavlja porabo centralno procesne enote na hipervisorju v procentih.



Slika 12: Poraba vCPE [%] preko povezave hitrosti 1Gb/s.



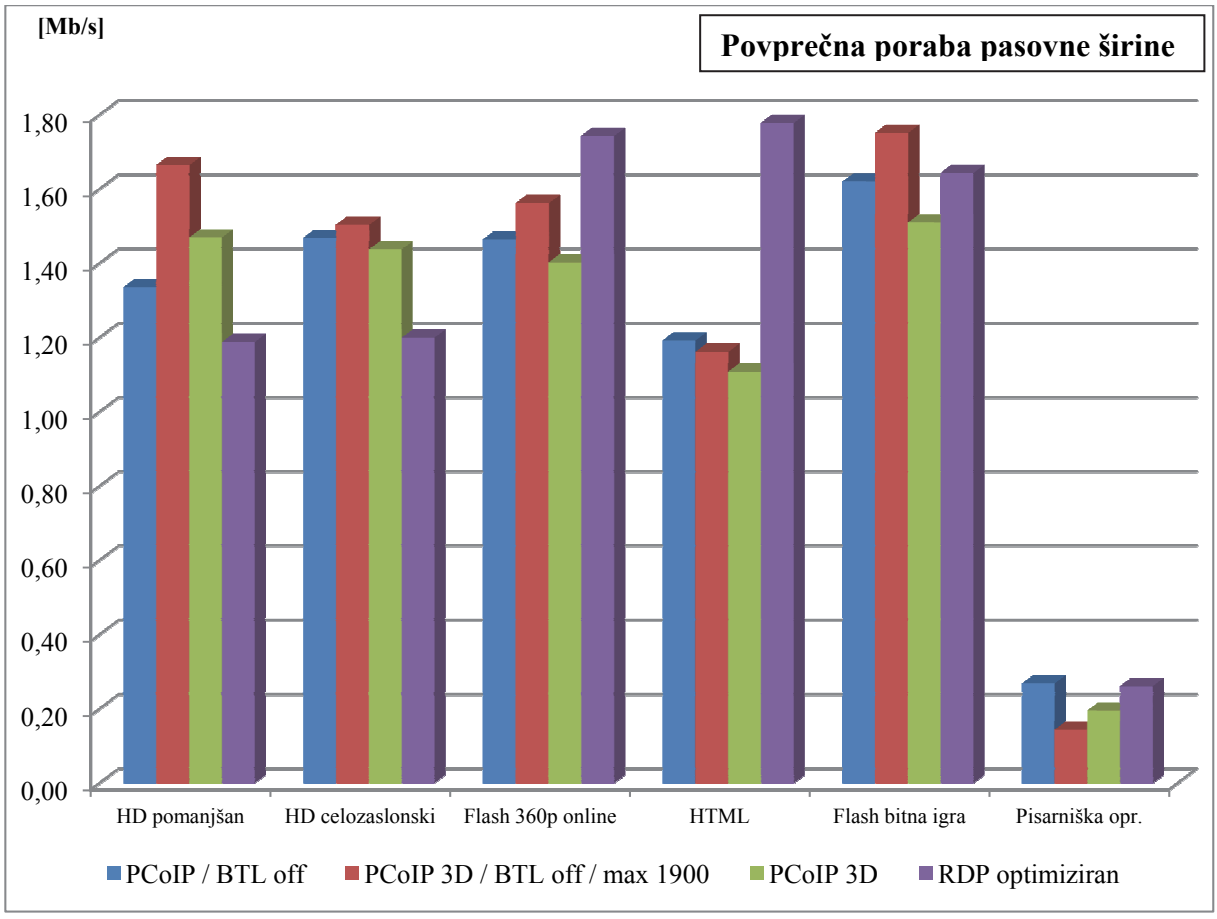
Slika 13: Poraba CPE [%] hipervisorja preko povezave hitrosti 1Gb/s.

Grafični prikaz na sliki 12 nam ponovno pokaže prednosti protokola RDP v primeru predvajalnika WMP, saj je procesna enota virtualne naprave pri predvajanju video vsebine popolnoma neobremenjena. Vklopljena funkcija 3D pri PCoIP protokolu je nekoliko razbremenila vCPE na virtualni napravi na račun povečane porabe CPE na hipervisorju. Tudi v segmentu porabe CPE se je protokol RDP RemoteFX izkazal kot najboljša rešitev, saj je porabil najmanj procesorske moči tako na virtualni napravi, kot tudi na hipervisorju. Razlog razbremenitve CPE gre pripisati tehnologiji RemoteFX, ki za upodabljanje uporablja grafično procesno enoto na strežniku.

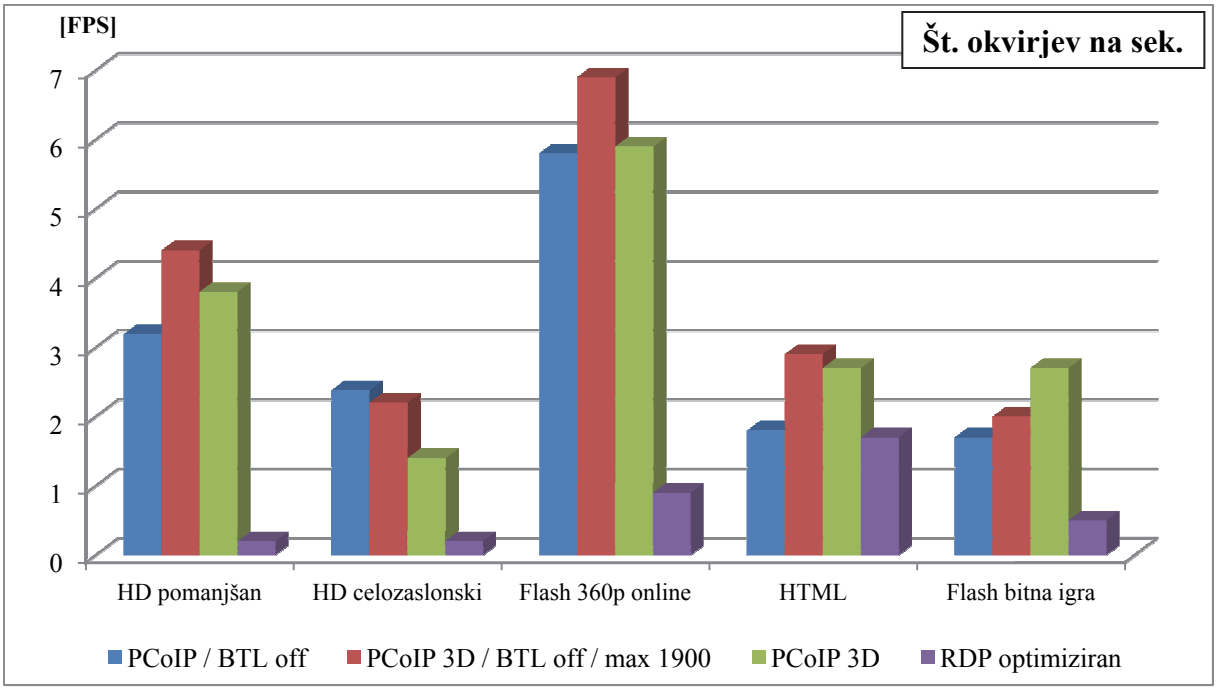
V lokalnem omrežju, kjer imamo na voljo dovolj pasovne širine, pod pogojem da imamo potrebno strojno opremo, je nesporni zmagovalec RDP RemoteFX protokol. Najmanjša poraba centralne procesorske enote in najboljša uporabniška izkušnja sta ključna faktorja za takšno odločitev.

Meritve pri hitrosti povezave 2Mb/s

Naslednji testi so bili opravljeni pri hitrosti povezave 2Mb/s z obhodnim časom RTT 40ms ter deležem izgubljenih paketov 0.001%. Tokrat smo imeli na razpolago omejeno pasovno širino, zato je bila optimizacija oddaljenega protokola zaželjena. Med seboj smo primerjali različne optimizacijske nastavitve protokola PCoIP ter optimiziran protokol RDP. Pri protokolu PCoIP smo meritve izvajali tako z vklopljeno, kot tudi izklopljeno podporo za 3D. Pri protokolu RDP z vklopljeno 3D podporo testov nismo izvajali, saj tehnologija RemoteFX ni primerna za počasne povezave. Pri RemoteFX se je neodzivnost oddaljenega namizja pokazala že pri najosnovnejših pisarniških opravilih. Graf na sliki 14 predstavlja povprečno porabo pasovne širine v Mb na sekundo. Graf na sliki 15 predstavlja število okvirjev oz. slik na sekundo.



Slika 14: Povprečna porabe pasovne širine [Mb/s] preko 2Mb/s povezave.



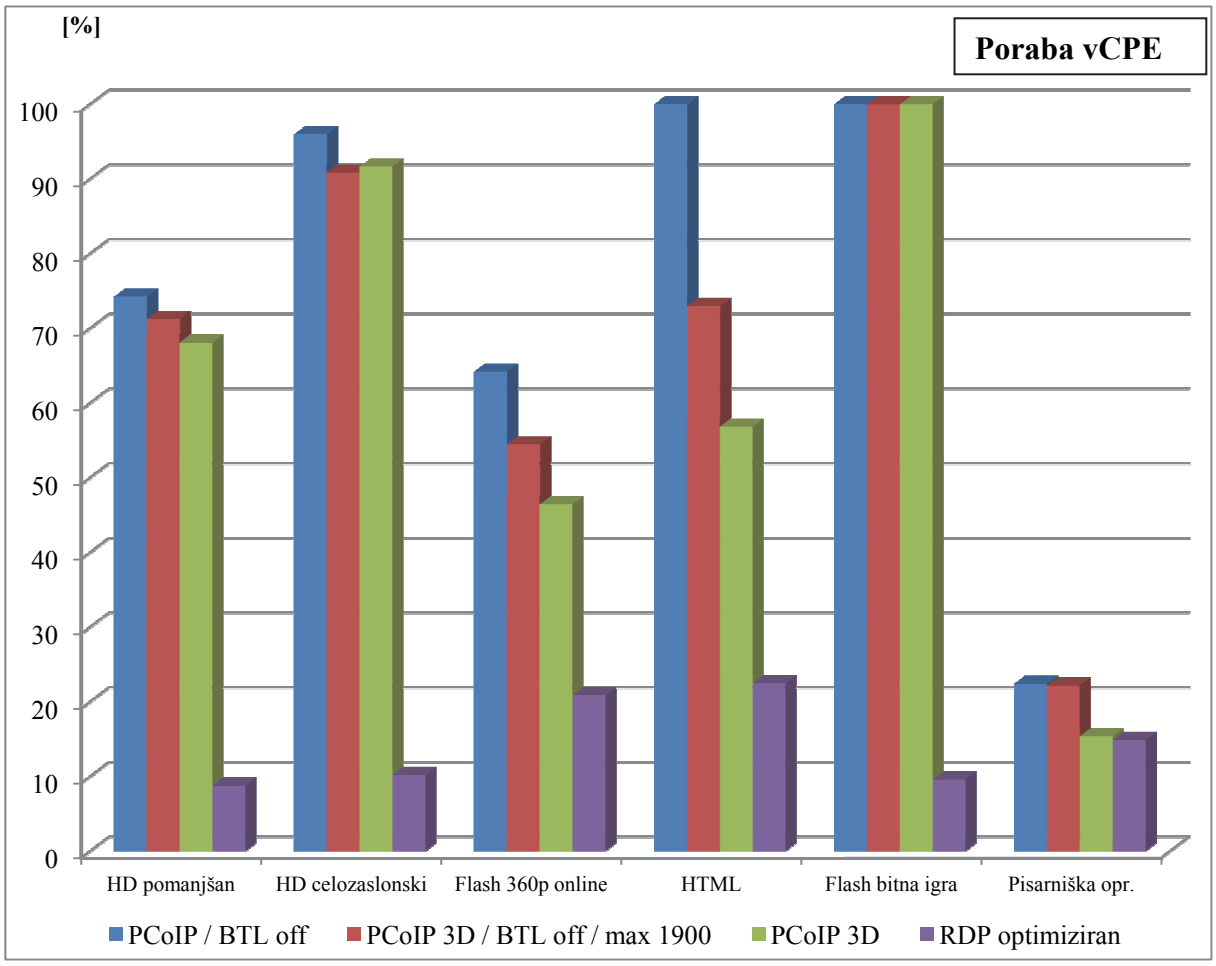
Slika 15: Št. okvirjev na sekundo [FPS] preko povezave hitrosti 2Mb/s.

Protokol PCoIP se je v testih na povezavi 2Mb/s odrezal bistveno bolje kot protokol RDP. Na skoraj vseh izbranih nastavitvah je PCoIP imel manjšo porabo pasovne širine, prav tako pa je bila uporabniška izkušnja bolj odzivna in tekoča. Protokol RDP je porabil manj pasovne širine samo ob predvajanju video posnetka v predvajalniku WMP iz razloga, ker odjemalcu pošilja samo surove grafične podatke. Za razliko od hitre LAN povezave pa je tokrat ta pristop napačen, saj se je na odjemalcu video posnetek, ki je dolg 30 sekund, predvajal 150 sekund. Zvok se je predvajalal pravilno. Razlog za takšno vedenje so surovi grafični podatki, ki so zapakirani v WMV datoteko, katera je nato poslana preko RDP protokola, ki pa uporablja zanesljivi TCP protokol. Tako noben paket ni izgubljen, poslan pa je vsak okvir videa. V uporabniški izkušnji se to kaže sicer v dobri kvaliteti, vendar kot bi gledal video v počasnem načinu (angl. *slow motion*); namesto nekaj sličic na sekundo moramo tukaj na eno sličico počakati več sekund. PCoIP protokol deluje na prenosnem protokolu UDP, kar pomeni da lahko izgubi kakšen paket in ga nato ne poizkuša poslati še enkrat. V praksi to pomeni manj sličic na sekundo, vendar tekoče predvajanje in sinhronizacijo slike in zvoka [6]. Z omejitvijo pasovne širine na 1900Kb/s smo dosegli bolj konstantno porabo. Ni bilo več ekstremnih poskokov, kar pa se je posledično poznalo na manjšemu deležu izgubljenih paketov in bolj tekoči uporabniški izkušnji. Pri PCoIP protokolu z nastavitvami *3D / BTL off / max 1900* je izklop BTL pripomogel k višjemu številu okvirjev na sekundo, kar je razumljivo, saj protokolu ni bilo treba sličic postavit v brez-izgubno (angl. *lossless*) stanje. Zanimiva je poraba pasovne širine ob vklopljeni 3D podpori brez omejitev, saj v povprečju porabi manj kot ostali načini. *PCoIP 3D* nima omejitev pasovne širine. Posledično lahko maksimalna

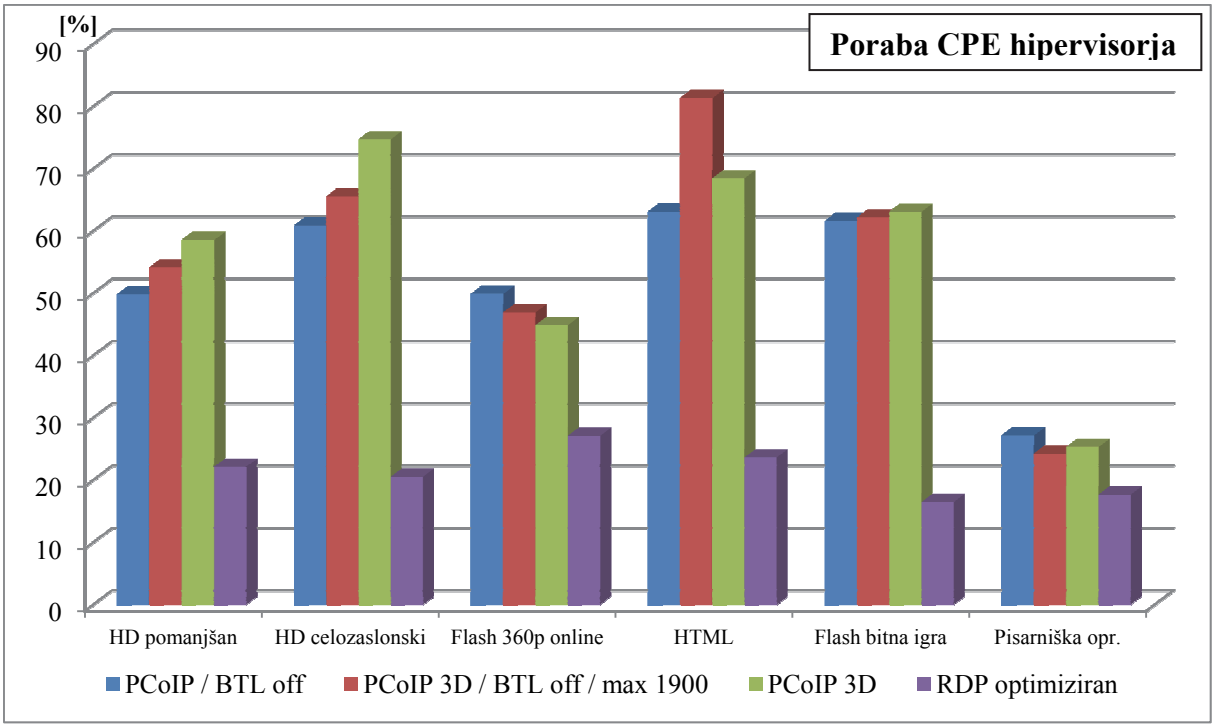
trenutna poraba poskoči in takrat se pojavi nekoliko več izgub. Podobna situacija je s *PCoIP / BTL off*, ki sicer porabi malo pasovne širine, vendar dosega višje vrednosti maksimalne porabe in posledično več izgub. Glede na predstavljene rezultate meritev se je najbolj odrezal protokol PCoIP z vklopljeno 3D podporo, izklopljeno funkcijo BTL in na 1900 Kb/s omejeno maksimalno porabo pasovne širine.

Graf na sliki 16 predstavlja porabo virtualne centralno procesne enote na virtualni napravi v procentih. Graf na sliki 17 predstavlja porabo centralno procesne enote na hipervisorju v procentih.

Iz grafov je razvidno, da najmanj procesorske moči porabi protokol RDP, torej je tako posledično bolj primeren za večje število nezahtevnih uporabnikov. Pri PCoIP protokolu vklopljena 3D podpora razbremeni vCPE na virtualni napravi na račun večje obremenitve CPE na hipervisorju. Za zahtevne uporabnike bomo posegli po PCoIP protokolu z vklopljeno 3D podporo ter tako razbremenili njihove virtualne naprave in jim zagotovili nekoliko boljšo uporabniško izkušnjo. Vendar pa ne smemo pozabiti na spremljanje monitorjev na hipervisorju, da ne bi prišlo do preobremenitve sistema.



Slika 16: Poraba vCPE [%] preko povezave hitrosti 2Mb/s.



Slika 17: Poraba CPE [%] hipervisorja preko povezave hitrosti 2Mb/s.

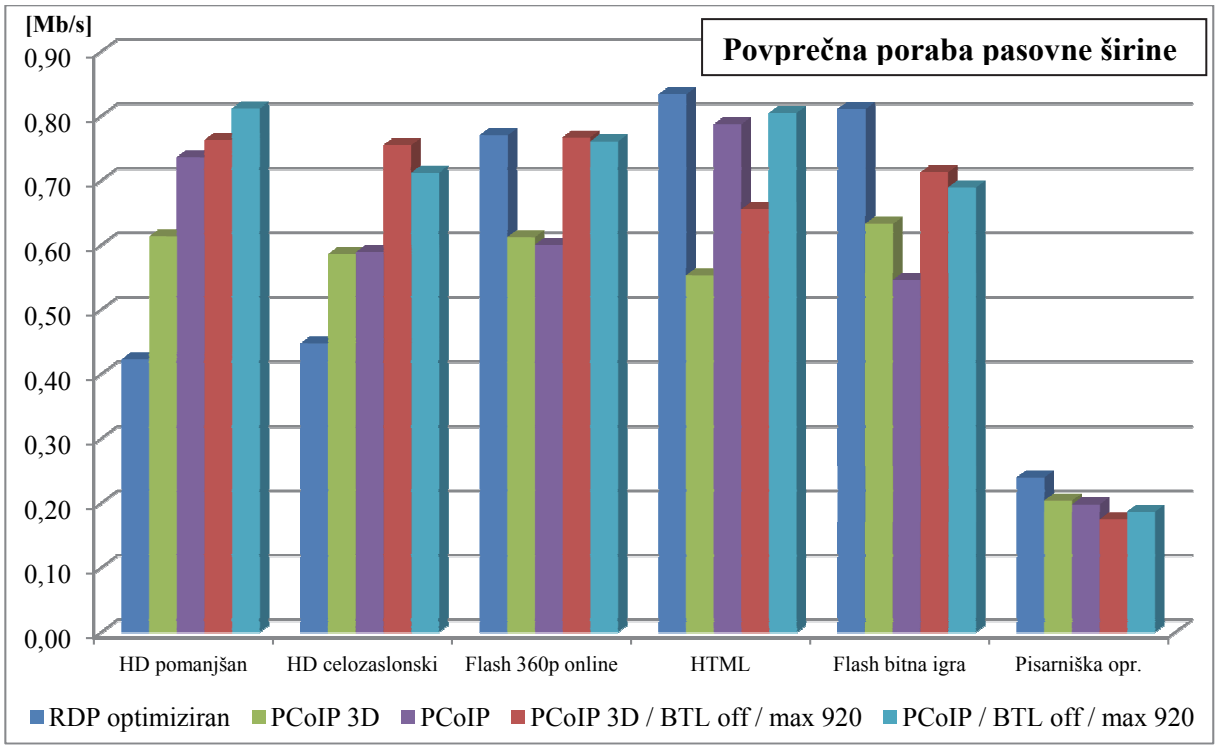
Meritve pri hitrosti povezave 1Mb/s

Naslednji testi so bili opravljeni pri hitrosti povezave 1Mb/s z obhodnim časom RTT 100ms ter deležem izgubljenih paketov 0.01%. Med seboj smo primerjali različne optimizacijske nastavitve protokola PCoIP ter optimiziran protokol RDP. Pri protokolu PCoIP smo meritve izvajali tako z vklopljeno, kot tudi izklopljeno podporo za 3D.

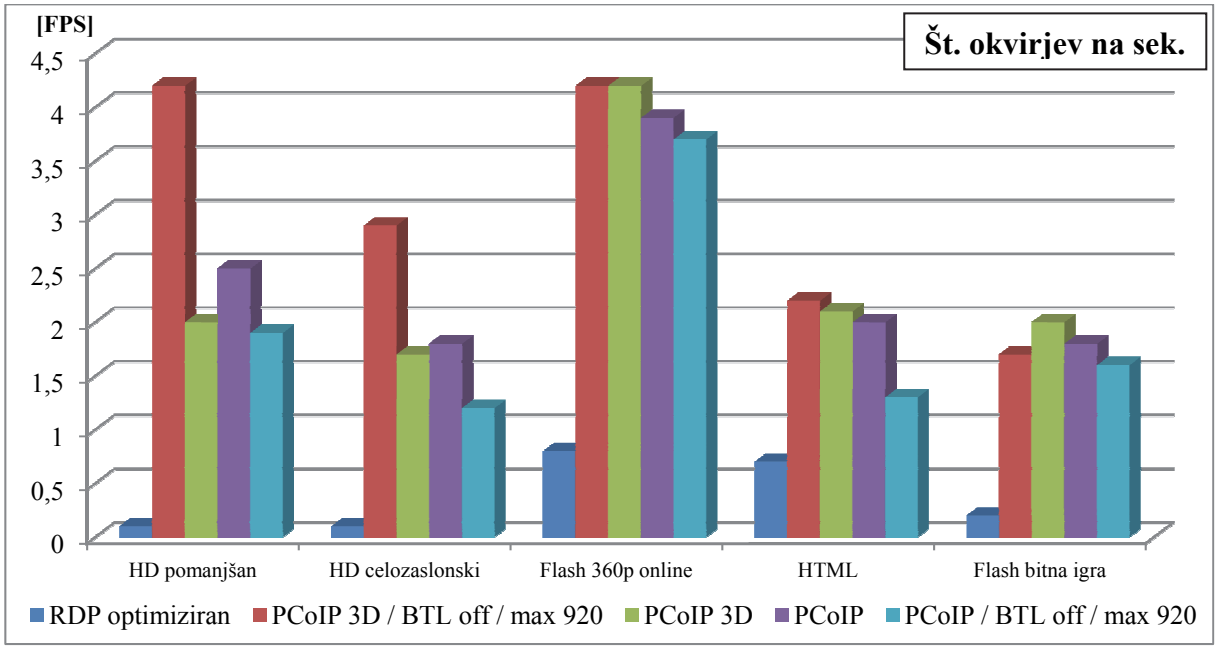
Graf na sliki 18 predstavlja povprečno porabo pasovne širine v Mb na sekundo. Graf na sliki 19 predstavlja število okvirjev oz. slik na sekundo. Graf na sliki 20 predstavlja porabo virtualne centralno procesne enote na virtualni napravi v procentih. Graf na sliki 21 predstavlja porabo centralno procesne enote na hipervisorju v procentih.

Rezultati meritev so podobni prejšnjim pri hitrosti povezave 2Mb/s. Ponovno se je bolje izkazal protokol PCoIP, predvsem z vklopljeno 3D podporo, izklopljeno funkcijo BTL in omejitvijo pasovne širine na 920Kb/s. Izguba paketkov se je najmanj poznala pri obeh nastavitvah z omejeno pasovno širino *PCoIP / BTL off / max 920* in *PCoIP 3D / BTL off / max 920*. Slednja je nudila najboljšo uporabniško izkušnjo, saj je prikazala največ sličic na sekundo, kar je razvidno iz grafičnega prikaza na sliki 19. Negativno je presenetil PCoIP protokol z nastavitvami *PCoIP / BTL off / max 920*, saj bi iz teoretične plati moral porabiti najmanj pasovne širine in ob zmanjšani kvaliteti ponuditi najbolj tekočo uporabniško izkušnjo, vendar temu ni bilo tako. Zadovoljivo uporabniško izkušnjo so ponudile tudi privzete nastavitve *PCoIP*, prav tako tudi PCoIP 3D, le da je v obeh primerih prihajalo do občasnih izgub paketkov, kar se je odražalo v popačenem zvoku in krajših zamrznitvah slike.

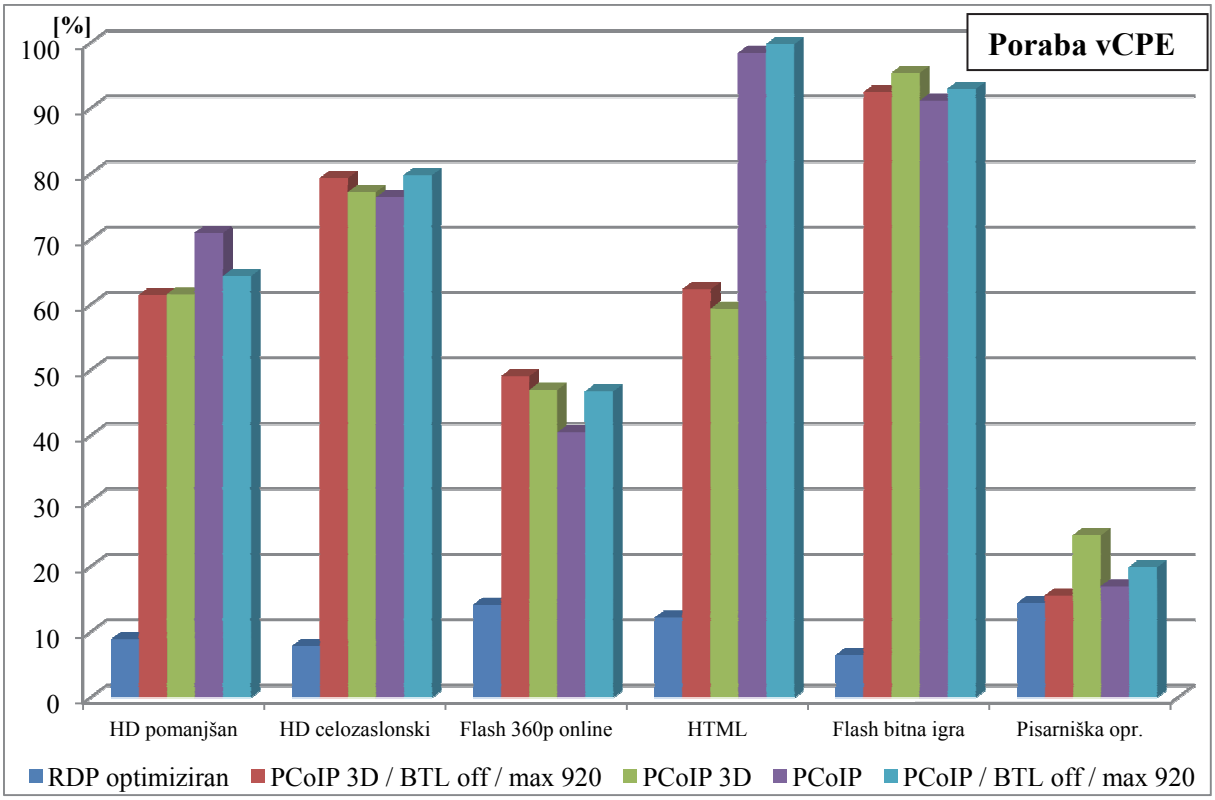
Najmanj procesorske moči zopet porabi protokol RDP. Pri PCoIP protokolu vklopljena 3D podpora razbremeni vCPE na virtualni napravi, na račun večje obremenitve CPE na hipervisorju.



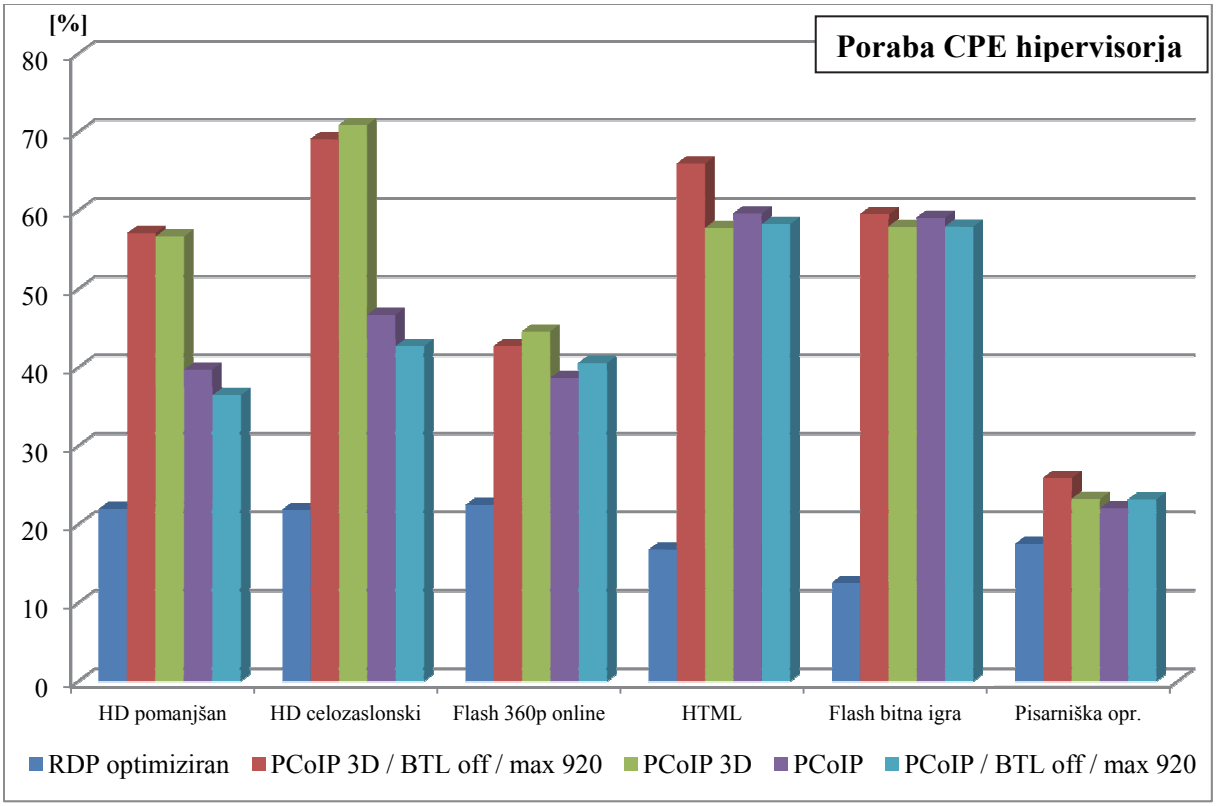
Slika 18: Povprečna porabe pasovne širine [Mb/s] preko 1Mb/s povezave.



Slika 19: Št. okvirjev na sekundo [FPS] preko povezave hitrosti 1Mb/s.



Slika 20: Poraba vCPE [%] preko povezave hitrosti 1Mb/s.



Slika 21: Poraba CPE [%] hipervisorja preko povezave hitrosti 1Mb/s.

Meritve so po pričakovanjih pokazale različne rezultate pri različnih pogojih. Oba protokola za prikaz oddaljenega namizja sta pokazala velik napredek v primerjavi s prejšnjimi različicami. Protokol RDP se je kot izjemno odzivnega pokazal v primeru uporabe tehnologije RemoteFX. Uporabniška izkušnja je tako primerna tudi za grafično zelo zahtevne aplikacije, poleg tega pa navdušuje še dejstvo, da lahko to izkušnjo prenesemo na lahke odjemalce, ki podpirajo protokol RDP 7. Slaba stran tehnologije RemoteFX je zahteva po dodatni strojni opremi, ki je standardni strežniki v podatkovnih centrih še nimajo, ali pa je celo ne morejo imeti.

Standardni RDP protokol verzije 7 ponuja mnogo vidnih izboljšav in se je v LAN okolju obnesel odlično, v večini primerov celo boljše kot PCoIP protokol. Žal pa ne podpira 3D funkcionalnosti in posledično ni mogoče izvajati zahtevnejših opravil, kot je npr. uporaba aplikacije Google Earth.

Če smo povzeli, da se je RDP protokol bolje obnesel v LAN okolju, pa je v WAN okolju situacija popolnoma drugačna. PCoIP nudi zadovoljivo uporabniško izkušnjo skoraj v vseh pogojih, predvsem po zaslugi pametnega dinamičnega prilagajanja glede na zmožnosti povezave. Ponuja tudi 3D podporo brez dodatne strojne opreme, na račun nekoliko večje porabe procesorske moči na strani gostitelja. Omeniti moramo še dodatno povečanje podatkovne shrambe za minimalno 256MB za vsako virtualno napravo s 3D podporo. Teh 256MB se doda sekundarni izmenjevalni (angl. *swap*) datoteki, z namenom da pomaga virtualni napravi, da ne bi prišlo do težav med izvajanjem 3D operacije. Ker PCoIP celotno upodabljanje naredi na strani strežnika, lahko to bogato uporabniško izkušnjo dostavimo skoraj na katerokoli končno napravo, tudi na »zero« odjemalce.

Zaključek te raziskave je naslednji:

- Za hitre LAN povezave, kjer imamo na voljo strežnike z namenskimi grafičnimi karticami, je brezkompromisni zmagovalec protokol RDP z RemoteFX.
- Za hitre LAN povezave, kjer nimamo na voljo strežnikov z namenskimi grafičnimi karticami in ne potrebujemo 3D podpore, bomo izbrali protokol RDP 7.
- Za hitre LAN povezave, kjer nimamo na voljo strežnikov z namenskimi grafičnimi karticami, vendar potrebujemo 3D podporo, bomo izbrali protokol PCoIP z vklopljeno funkcijo 3D.
- Za počasne WAN povezave bomo izbrali protokol PCoIP s podporo za 3D in vklopljeno funkcijo BTL, če potrebujemo kvaliteto brez izgub. V nasprotnem primeru lahko BTL izklopimo in prihranimo nekaj pasovne širine. Kot opombo dodajmo da vklop 3D podpore izvedemo le v primeru zadostnih računalniških virov.
- Za ekstremno počasne WAN povezave bomo izbrali protokol PCoIP brez podpore za 3D in izklopljeno funkcijo BTL. V primeru da bo prihajalo do velikih izgub paketov, bomo omejili še maksimalno pasovno širino protokola PCoIP na vrednost 10% manj kot je celotna pasovna širina povezave.

4 ZAKLJUČEK

V diplomskem delu smo obravnavali različne protokole za prikaz oddaljenega namizja s poudarkom na protokolih za prikaz oddaljenega *virtualnega* namizja. Zmogljivostno analizo smo izvedli nad zelo dobro uveljavljenim protokolom RDP ter nekoliko manj poznanim in novejšim protokolom PCoIP. Motivacija teh raziskav je bila predvsem hiter razvoj virtualizacije računalniških namizij (angl. *VDI*), ki se pojavljajo kot alternativa klasičnim osebnim računalnikom. Ključno vlogo pri dostopu do virtualiziranih računalniških namizij odigrajo prav ti obravnavani protokoli. Zanimalo nas je, če so protokoli za prikaz oddaljenega virtualnega namizja že dovolj zmogljivi in prilagodljivi, da lahko nudijo zadovoljivo uporabniško izkušnjo in s tem zagotovijo nadaljnje širjenje tehnologije VDI.

Analiza protokolov je potekala na dveh različnih VDI infrastrukturah, saj protokol PCoIP deluje samo v VMware View okolju, medtem ko protokol RDP s tehnologijo RemoteFX deluje samo v Microsoft Hyper-V okolju. Pri testiranju nas je zanimalo, kako uspešno protokola prikazujeta oddaljeno virtualno namizje v različnih pogojih.

Rezultati analize so pokazali, da protokola v lokalnem omrežju ponujata končnemu uporabniku odlično uporabniško izkušnjo, v določenih primerih pa celo identično kot na lokalnem računalniku. Kljub temu, da oba protokola zadovoljita potrebe večine uporabnikov, moramo izpostaviti RDP protokol s tehnologijo RemoteFX. Ne le, da ponuja odlično odzivnost, temveč tudi razbremeni centralni procesni enoti na virtualni napravi in hipervisorju, ker večji del grafično zahtevnih opravil opravi grafična procesna enota v fizičnem strežniku. Slabost te tehnologije je v potrebi po dodatni strojni opremi v strežniku, saj ima danes zelo malo strežnikov dodatno grafično kartico, ki jo podpira tehnologija RemoteFX. Tudi brez te napredne tehnologije se je RDP 7 v lokalnem omrežju izkazal dobro pod pogojem, da ne potrebujemo 3D podpore za razne aplikacije, kot je Google Earth. Če potrebujemo podporo za 3D aplikacije in nimamo strežnikov z močnimi grafičnimi karticami, bomo posegli po VMware rešitvi in uporabili protokol PCoIP z vključeno podporo za 3D. Zaradi vklopljene 3D podpore bomo nekoliko bolj obremenili centralno procesno enoto na hipervisorju, medtem ko bomo virtualno procesno enoto na virtualni napravi nekoliko razbremenili. Omeniti moramo še dodatno povečanje podatkovne shrambe za minimalno 256MB za vsako virtualno napravo s 3D podporo. Uporabniška izkušnja protokola PCoIP je sicer nekoliko slabša od RDP RemoteFX, vendar še vedno zadovoljiva za večino opravil, vključno s 3D aplikacijami.

Če smo bili nad odzivnostjo protokolov v lokalnem omrežju pozitivno presenečeni, so nas v prostranem WAN omrežju žal postavili na realna tla. Protokol RDP RemoteFX je na počasnih povezavah neodziven in zato ni bil primeren za izvajanje testov. Protokol RDP 7 brez RemoteFX je pokazal zadovoljive rezultate z vidika nezahtevnega uporabnika za potrebe

pisarniških opravil. Protokol PCoIP je ponudil najboljšo uporabniško izkušnjo, ki je bila zadovoljiva tudi pri nekoliko zahtevnejših opravilih, kot je predvajanje video vsebin. Napredno dinamično prilagajanje glede na zmožnost povezave se je pri PCoIP protokolu izkazalo ključnega pomena. Prednost protokola PCoIP se kaže tudi v opcijsko dodatni ročni optimizaciji, saj ponuja ogromno nastavitev, ki lahko močno izboljšajo uporabniško izkušnjo. Optimizacijo sicer ponuja tudi protokol RDP, vendar nikakor ne v takšnem obsegu, kot jo ponuja protokol PCoIP. Na račun nekaj dodatnih računalniških virov PCoIP ponuja 3D podporo brez dodatne strojne opreme tudi na počasnih povezavah.

Z zmogljivostno analizo smo pokazali, kako se aktualne verzije protokola RDP in PCoIP obnesejo v različnih pogojih. Pokazali smo, da je nemogoče posplošiti, kateri protokol je najboljši, saj vsak protokol uporablja svoje pristope, ki se v nekaterih pogojih odlično obnesejo, spet v drugih pa ne najbolje. Teste smo izvajali z nekaj površinskimi optimizacijskimi metodami, kajti verjamemo, da bi z bolj globinsko optimizacijo lahko z obema protokoloma dosegli boljše rezultate. Zanimivo bi bilo vključiti v raziskavo še druge protokole, predvsem Citrixov protokol ICA/HDX, ali Ericom Blaze. Slednji pospešuje in optimizira RDP protokol. Vendar je to žal presegalo obseg našega diplomskega dela.

Vsaka novejša verzija protokolov prinaša nekaj izboljšav, dodatnih funkcionalnosti in optimizacij, vendar ključno vlogo igra povezava med odjemalcem in virtualnim namizjem. Protokoli so dokazali, da v lokalnem hitrem omrežju že dosegajo zavidljive rezultate, vendar bo potrebno le-te dosegati tudi na prostranih WAN omrežjih, če bomo želeli slediti napovedi, kjer naj bi virtualna namizja zamenjala 40% poslovnih namiznih računalnikov. Korak naprej so naredili tudi ponudniki internetnih storitev, saj imamo danes na voljo veliko hitrejše povezave in s tem zmanjšan vpliv povezave na odzivnost protokola.

Rešitev VDI uporabljam tudi v podjetju NIL, kjer do svojega virtualnega namizja dostopam preko lahkega odjemalca. Do istega virtualnega namizja dostopam tudi od doma preko prenosnika. Ker pa je moja internetna povezava hitrosti 100Mb/s, je uporabniška izkušnja popolnoma enaka, kot če bi sedel na delovnem mestu. Ko bodo takšni rezultati mogoči tudi na počasnejših povezavah (xDSL, 3G), se bodo napovedi VDI tehnologije uresničile. Vedno bolj optimizirani protokoli in vedno hitrejše WAN povezave peljejo VDI tehnologijo v pravo smer. Kljub temu pa se to odvija nekoliko počasneje, kot so napovedali pri analitični skupini Gartner, kjer naj bi do leta 2013 uporabljalo VDI že 49 milijonov uporabnikov [3].

LITERATURA IN VIRI

[1] N. Zimic, M. Mraz, Temelji zmogljivosti računalniških sistemov. Ljubljana: Fakulteta za računalništvo in informatiko, 2006, str. 35, str. 23-78.

[2] T. Richardson, *The RFB Protocol ver. 3.8*, nov 2010, str. 3-6 in 33-34.

[3] A. Jump, B. Gammage, *Emerging Technology Analysis: Hosted Virtual Desktops*,. Gartner for Business Leaders, feb 2009, str. 6

[4] V. Djurdjič, »VDI – revolucija na namizju«, *Sistem poletje 2010*, maj 2010

[5] (2010) The performance impact of distance, bandwidth and latency on VDI. Dostopno na: http://searchvirtualdesktop.techtarget.com/tip/The-performance-impact-of-distance-bandwidth-and-latency-on-VDI

[6] (2011) The Layer 4 protocols behind PCoIP and HDX: Which is better for VDI? Dostopno na: http://searchvirtualdesktop.techtarget.com/feature/The-Layer-4-protocols-behind-PCoIP-and-HDX-which-is-better-for-VDI

[7] (2011) Comparing Microsoft RemoteFX to VMware PCoIP. Dostopno na: http://searchvirtualdesktop.techtarget.com/feature/Comparing-Microsoft-RemoteFX-to-VMware-PCoIP

[8] (2011) VMware details PCoIP bandwidth improvements for View 5 virtual desktops. Dostopno na: http://searchvirtualdesktop.techtarget.com/news/2240039008/VMware-details-PCoIP-bandwidth-improvements-for-View-5-virtual-desktops

[9] (2009) Why is VDI changing into Terminal Server? Dostopno na: http://www.jimmoyle.com/2009/05/why-is-vdi-changing-into-terminal-server/

[10] (2007) When to use VDI, when to use server-based computing, and how the Citrix Ardence dynamic desktop fits into all this. Dostopno na: http://www.brianmadden.com/blogs/brianmadden/archive/2007/03/14/when-to-use-vdi-when-to-use-server-based-computing-and-how-the-citrix-ardence-dynamic-desktop-fits-into-all-this.aspx

[11] (2010) VMware View-ThinPrint. Dostopno na: http://bright-streams.com/?p=120

[12] (2011) RemoteFX and a Rich RDP Experience. Dostopno na: http://www.windowsitpro.com/content1/topic/remotefx-rich-rdp-experience-139872/catpath/virtual-machine-vm-virtualizationclient/page/4

[13] (2009) Remote display protocols for VDI: will RDP be enough? Dostopno na: http://blogs.msdn.com/b/rds/archive/2009/08/21/remote-desktop-connection-7-for-windows-7-windows-xp-windows-vista.aspx