Applications Area Working Group
Internet-Draft
Intended status: Informational
Expires: November 14, 2011

S. Ma, Ed.
L. Liang
J. Wang
ZTE Corporation
May 13, 2011

Survey of Virtual Desktop Infrastructure System
draft-ma-appsawg-vdi-survey-00

Abstract

This document presents a survey of VDI (Virtual Desktop Infrastructure) system. Current popular VDI solutions are focused on, such as Microsoft RDS, Citrix XenDesktop, Redhat enterprise virtualization for desktops and VMware View. By analyzing the architecture and protocol flows of these solutions, the common features of VDI architecture and protocol are summarized.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on November 14, 2011.

Table of Contents

1. Introduction

Virtual desktop infrastructure (VDI) is the practice of hosting a desktop operating system within a virtual machine (VM) running on a centralized server. VDI provides services for users to access their desktops remotely and users can choose PC, tablet computer or mobile phone as the client device. The OS (Operating System) and applications of the desktop are running on virtual machines which are deployed in Data Center. All desktops are managed under a central management system.

The benefits of VDI include reduction of IT cost and improvement of security. Centralized desktop management makes IT maintenance tasks more effective, for example backup and upgrade can be done on Data Center side. Since enterprise's data is kept in Data Center, it will not be lost even when client devices are broken. Users can access the data but can't copy it out of the datacenter, so the data security is guaranteed.

The mainstream VDI products include Microsoft's RDS, Citrix's XenDesktop, Redhat's Enterprise Virtualization for Desktops and VMware's VMware View. The architecture, features and VDI protocols of these products are analyzed in this document.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

1.2. Terminology and Abbreviation

- Device redirection: In VDI system, remote desktop must provide the same user experience as local desktop, so the physical interfaces of client device, such as USB, audio/video interface and serial/ parallel ports should be accessed by Guest OS on remote desktop. So client device's interfaces must be redirected to remote virtual machine.

- GCC: Generic Conference Control

- GPU: Graphic Processor Unit

- Guest OS: An OS runs on top of a virtual machine manager.

- HDX: High-Definition user eXperience

- ICA: Independent Computing Architecture
- MCS: Multipoint Communication Service
- OS: Operation System
- PDU: Protocol Data Unit
- RD: Remote Desktop
- RDP: Remote Desktop Protocol
- RDS: Remote Desktop Service
- Resource pool: Resource pool is based on a large quantity of physical server. The computing and storage resources of the servers are managed as a whole, and it is a logical resource pool to users. Users need not to know which server the resource is on when he use it. Resource pool can improve resource's utilization and scalability, and resource will be managed more effectively.
- RFB: Remote Framebuffer
- SPICE: Simple Protocol for Independent Computing Environment
- TPKT: ISO transport services on top of the TCP
- VDA: Virtual Desktop Agent, which acts as the bridge between virtual machine and VDI client, such as accepting VDI connection, taking input from VDI client, transferring graphics output of virtual machine to VDI client and etc.
- VDI: Virtual Desktop Infrastructure
- VM: Virtual machine is a logical machine. Several virtual machines can share one physical machine's resource. The virtual CPU, memory and storage are allocated from physical resource by virtualization layer software, and they can be scheduled under certain policy.
- VMM: Virtual Machine Manager

2. Survey of VDI System

Remote terminal technology was brought into market very early, such as X-WINDOW, Microsoft's Terminal Services and RealVNC's VNC. Early products are used for server sharing, remote collaboration, and

remote control. The messages transferred between server and client contain screen images and the keyboard/mouse events. VDI inherits from the remote terminal technology, with exception that the desktops are not running on physical servers, but in virtual machines. Virtual machines can be allocated on demand, so VDI is more flexible and has higher resource utilization. VDI also adopts variety of technologies to improve user experience, such as enhanced desktop display quality, supporting for USB/audio/video device redirection and etc.

2.1. Citrix XenDesktop

XenDesktop is Citrix's VDI solution. For Guest OS in XenDesktop, it could only be Microsoft's Windows. For Client device, it could support different platforms, like Windows, Linux and Mac OS. Virtualization platform can be Citrix's XenServer, or 3rd party's products, such as Microsoft Hyper-V and VMware ESX. The protocol between client and server is ICA (Independent Computing Architecture).

The architecture of XenDesktop is shown in Figure 1:

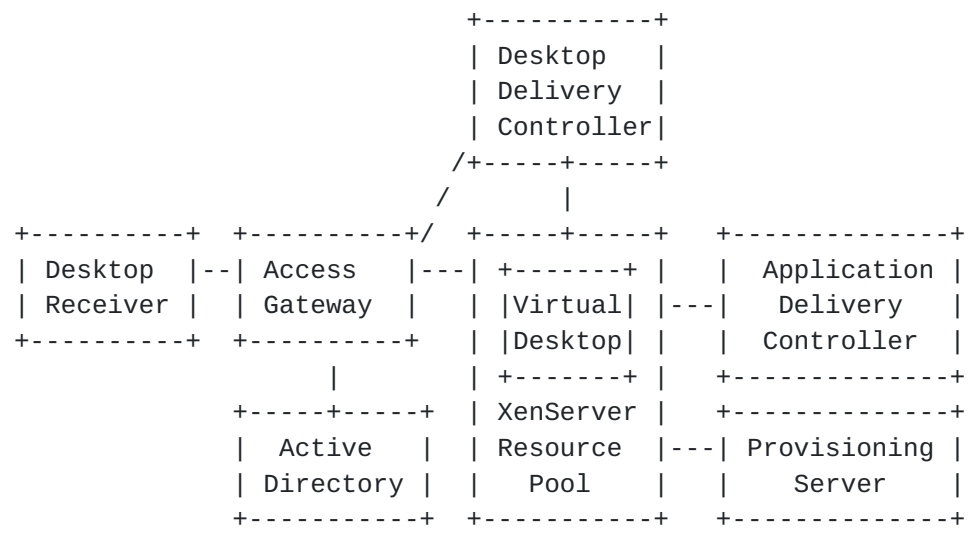

Figure 1 XenDesktop architecture

The key components of XenDesktop:

1. Desktop Receiver: The client-side software, which is running on client devices, connects to remote desktop through ICA protocol.

2. Access Gateway: Responsible for external user authentication. Before the external user accesses the desktop, he must connect to

the Access Gateway and pass the authentication. External users's connection to Access Gateway is secured by SSL.

3. Desktop Delivery Controller: Responsible for management of virtual desktop and client connection. When a virtual desktop boots up, it will register with Desktop Delivery Controller. Desktop Delivery Controller will maintain the state of the desktop. When a client initiates a connection, Desktop Delivery Controller chooses an appropriate virtual desktop to connect. Finally it forwards the client's connection to the desktop.

4. XenServer Resource Pool: XenServer is Citrix's virtualization platform, provides virtual machine for virtual desktop and manages all of the virtual machines.

5. Provisioning Server: Responsible for providing OS images. OS images contain pre-configured profile, but applications are not included. When virtual desktop boots up, the OS image will be sent to virtual machine over network.

6. Application Delivery Controller: Responsible for application delivery. User's profile contains assigned application list, and user can choose which application to load.

7. Virtual Desktop Agent: Running on virtual desktop, which accepts the ICA connection from client.

8. Active Directory: Responsible for user authentication.

Besides above specific components, additional general components could be deployed, such as License server, DNS Server, DHCP Server and so on.

Before users access their virtual desktops, they must pass the authentication. Following is the procedure:

1. User launches the browser, and connects to Access Gateway. Access Gateway generates the login page;

2. User inputs credentials, and Access Gateway authenticates the user's credentials against Active Directory;

3. Upon successful authentication, Access Gateway connects to Desktop Delivery Controller;

4. Desktop Delivery Controller retrieves the credentials from Access Gateway, and authenticates the user against Active Directory;

5. After success authentication, Desktop Delivery Controller determines which virtual desktops are available for the user;

6. Access Gateway creates a web page containing a list of virtual desktops for the user, and sends the web page to user's browser.

After successful authentication, user can access virtual desktop by clicking the icon on the web page. Following is the procedure:

1. User clicks the virtual desktop's icon on the web page, Access Gateway sends the information to Desktop Delivery Controller;

2. Desktop Delivery Controller checks the status of user's virtual desktop. If there is no available desktop, Desktop Delivery Controller will request XenServer to launch one. The desktop then opens the ICA port to wait for client connections;

3. Desktop Delivery Controller creates an ICA file for the virtual desktop, and transfers the file to Access Gateway. Then Access Gateway transfers the ICA file to VDI client;

4. After receiving the ICA file, VDI client executes it and sends a connection request to Access Gateway;

5. Access Gateway forwards the connection request to virtual desktop;

6. When detecting the connection request, the virtual desktop sends the logon information to Desktop Delivery Controller for validation;

7. For valid validation, Desktop Delivery Controller fetches a license from the license server and sends credentials and license to virtual desktop. The connection is created after virtual desktop logon against Active Directory with the credentials.

To optimize user experience, XenDesktop provides HDX (High Definition eXperience) technology. HDX contains several techniques as following:

1. HDX MediaStream: Optimized for multimedia stream, the compressed multimedia packages are transferred in raw format to client device, and the client device plays them locally;

2. HDX RealTime: Enhances real-time voice and video using advanced encoding and streaming to ensure a no compromise end-user experience;

3. HDX Broadcast: Optimized for bandwidth to ensure high-performance of virtual desktops and applications over any network, including high-latency and low-bandwidth environments. User can set bandwidth limits and close some features to save bandwidth;

4. HDX Plug-n-Play: Enables devices redirection for USB, printers and peripherals;

5. HDX RichGraphics: Optimizes for 2D/3D applications by using the rending ability of server and client;

6. HDX WAN Optimization: Adopts compression and cache technologies to reduce bandwidth on the WAN.

2.2. Microsoft RDS

RDS (Remote Desktop Service) is next-generation product of Microsoft TS (Terminal Service). In Windows server 2008 R2, it is named as RDS. The protocol used in RDS between desktop client and server is RDP (Remote Desktop Protocol).

The architecture of RDS is presented in Figure 2:

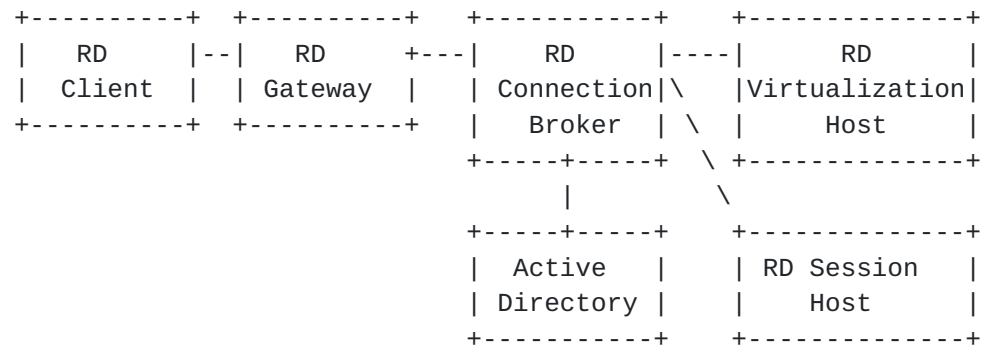

Figure 2 RDS architecture

The key components in RDS:

1. RD Client: Remote desktop client software running on client devices, which connects to virtual desktop using RDP protocol;

2. RD Gateway(RDG): RD Gateway authorizes external RD clients and provides secure connection for external RD client to access virtual desktop;

3. RD Connection Broker (RDCB): Responsible for assigning correct resource to RD clients and session load balancing. User can have

a personal desktop, which means that the RDCB assigns the same virtual machine to the user whenever he logins. RDCB also can assign virtual machine dynamically. When user logs on, RDCB selects one from virtual machine pool and assign it to user. Every user shares the same virtual machine image;

4. Remote Desktop Session Host (RDSH): Responsible for hosting windows-based programs or the full Windows desktop for RD clients. Users can connect to an RD Session Host server to run programs, save files, and use network resources on that server;

5. RD Virtualization Host(RDVH): RDVH integrates with Hyper-V to provide virtual machines for virtual desktops;

6. Active Directory: Responsible for user access authentication.

The connection procedure for RD Client is shown as below:

1. A RD Session Host works as a RD Redirector. When a client connects to RD Redirector, RD Redirector forwards the connection to RD Connection Broker;

2. RD Connection Broker queries personal desktop configuration from Active Directory;

3. RD Connection Broker retrieves user's virtual machine from RD Virtualization Host. If user's virtual machine is not running, RD Virtualization Host will launch it;

4. RD Connection Broker returns the information of virtual machine to RD Redirector;

5. RD Redirector forwards the information to RD Client,and redirects RD Client to virtual machine;

6. RD Client connects to virtual machine.

To improve user experience, RDS provides RemoteFX technology in RDP 7.1. RemoteFX contains a set of features shown as following:

1. Host side rendering: Allows graphics to be rendered on the host device instead of the client device. This enables support for all graphics types by sending highly compressed bitmap images to the endpoint device in an adaptive manner. This also allows the applications to run at full speed on the host computer by taking advantage of the GPU and the CPU, providing an experience which is similar to that of a local computer;

2. GPU virtualization: Supports several virtual GPU with one physical GPU, and each virtual GPU is used by one virtual machine;

3. Intelligent Screen Capture: Only the screen change will be encoded and transferred to client. Intelligent Screen Capture also tracks network speed and then dynamically adjusts according to the available bandwidth;

4. RemoteFX Encoder: Allows encoding to be done either on the processor, GPU, or dedicated hardware;

5. RemoteFX Decoder: Decodes bitmaps on the client computer which have been remoted from the virtual desktop to the client computer. The client computer can decode by using software in the GPU, processor, or by using a hardware decoder;

6. USB device redirection: Supports all usb devices redirection over RDP, no client side drivers needed.

2.3. Redhat Enterprise Virtualization for Desktops

Redhat Enterprise Virtualization for Desktops is Redhat's VDI solution,which supports both Windows and Linux desktop. The protocol between client and desktop is SPICE (Simple Protocol for Independent Computing Environment).

The architecture of Redhat Enterprise Virtualization for Desktops is shown in Figure 3:

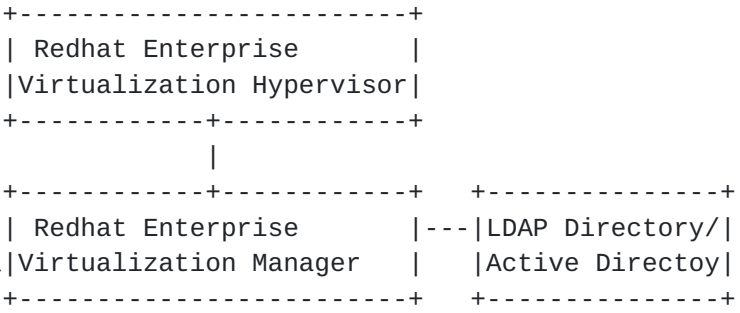

Figure 3 Redhat Enterprise Virtualization for Desktops architecture

The key components of Redhat Enterprise Virtualization for Desktops:

1. SPICE client: The client side software installed on client devices, which connects to virtual desktop using SPICE protocol;

2. SPICE Protocol: An open source VDI protocol, used between SPICE client and virtual desktop;

3. Red Hat Enterprise Virtualization Hypervisor(RHEV-H): Redhat virtualization platform to provide virtual machine for virtual desktops;

4. Red Hat Enterprise Virtualization Manager(RHEV-M): Management System for all components in Redhat's VDI solution;

5. LDAP Directory/Active Directory: Responsible for authentication.

The connection procedure for SPICE client is described as below:

1. User launches browser and connects to RHEV-M, and RHEV-M returns a logon page. Then user inputs corresponding ID and password information;

2. RHEV-M authenticates user against Active Directory;

3. After successful authentication, RHEV-M returns a web page containing a list of available virtual desktops for the user;

4. User selects the desktop by clicking the icon on the web page. Then RHEV-M requests RHEV-H to run the specified virtual desktop;

5. The SPICE client connects to the virtual desktop with SPICE protocol.

2.4. VMware View

VMware View is VMware's virtual desktop solution, which uses VMware vSphere as the virtualization platform. VMware View supports several VDI protocols, like Microsoft RDP, PCoIP and HP RGS protocols. For HP RGS, it can only be used for connecting HP Blades.

The architecture of VMware View is shown in Figure 4.

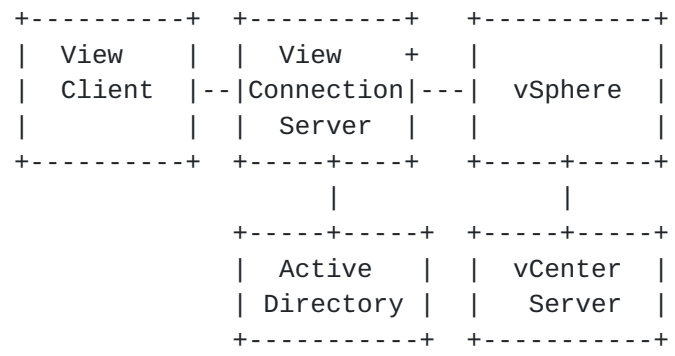

Figure 4 VMware View architecture

The key components of VMware View:

1. View client: The client software for accessing View desktops, which runs either on a Windows or Mac PC as a native application or on a thin client;

2. View Connection Server: View Connection Server acts as a broker for client connections. It authenticates users through Active Directory and directs the request to the appropriate virtual machine;

3. vSphere: The virtualization platform, which provides virtual machines for virtual desktops;

4. vCenter Server: Management System for configuration, deployment and virtual machine management;

5. View Agent: View Agent is installed on all virtual machines. This agent communicates with View Client to provide features such as connection monitoring, virtual printing, and access to locally connected USB devices;

VMware View has following features:

1. VMware View supports USB, printers and other peripherals redirection;

2. Users can configure the amount of bandwidth used by Adobe Flash content to improve the overall Web browsing experience and make other applications more responsive;

3. VMware View supports multimedia redirection, and the media file is send to client to be decoded locally;

4. VMware View supports single-sign-on, which means that users need to log in only once to access virtual desktop;

5. Besides Active Directory authentication, VMware also supports RSA SecurID and Smart Card authentication.

3. VDI Related Protocols

In this section, current hot-discussed VDI protocols, like RDP, SPICE and ICA, are investigated. For closed protocol, such as ICA, the information is collected from published documents on Internet. For each protocol, they are organized from perspective of: Basic architecture, Bandwidth saving and User experience improvement. Bandwidth saving and User experience improvement are two key factors for VDI protocol success.

3.1. T.120

T.120 is listed here for better understanding of Microsoft's RDP, which is described in next section.

3.1.1. Introduction

The T.120-series Recommendations, which are issued by ITU, collectively define a multipoint data communication service for use in multimedia conferencing environments. Depending on the type of T.120 implementations, the result product can make connections, transmit and receive data, and collaborate using compatible data conferencing features, such as program sharing, whiteboard conferencing, and file transfer.

The key functionalities of T.120 are:

1. Establish and maintain conferences without any platform dependence;

2. Manage multiple participants and programs;

3. Send and receive data accurately and securely over a variety of supported networking connections.

3.1.2. T.120 protocol suite

3.1.2.1. Basic architecture

The T.120 system model is comprised of a communications infrastructure and the application protocols that make use of it.

Figure 5 shows the full model with both standardized and non-standardized applications.

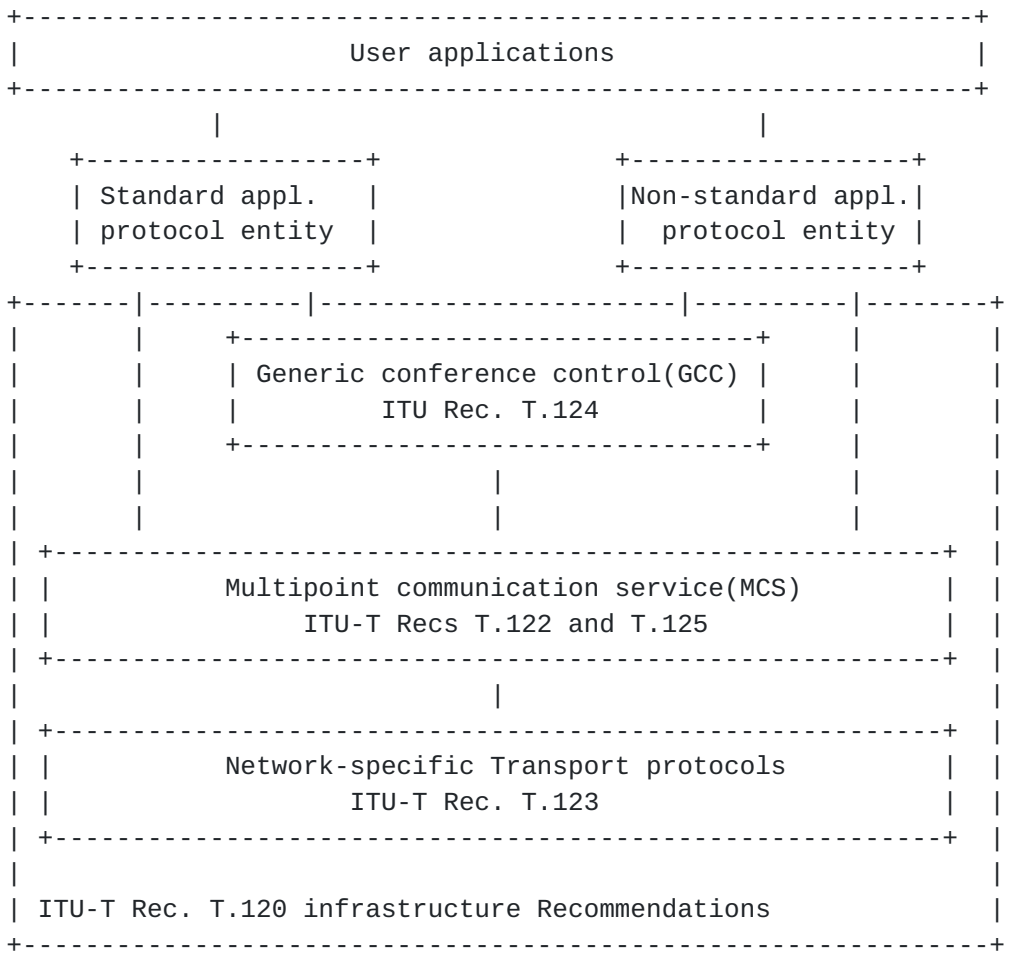

Figure 5 T.120 protocol

Generally, each layer provides services to the layer above and communicates to its peer(s) by sending Protocol Data Units (PDUs) via services provided by the layer below. The lower level layers (T.122, T.123, T.124, and T.125) specify an application-independent mechanism for providing multipoint data communication services to any application that can use these facilities. The upper level application layers define protocols, such as shared white boarding and multipoint file transfer. Applications using these standardized protocols can co-exist in the same conference with applications using proprietary protocols.

Following is one brief description of each protocol:

1. T.121: T.121 provides a template for T.120 resource management that developers should use as a guide for building application protocols. It also provides guidance to user application developers on how to utilize the T.120 infrastructure in a coherent and consistent way;

2. T.122:T.122 defines the multi-point services available to the developer. Together with T.125, they form MCS, the multi-point "engine" of the T.120 conference. MCS relies on T.123 to actually deliver the data. MCS is a powerful tool that can be used to solve virtually any multi-point application design requirement;

3. T.123: T.123 specifies transport profiles for different transport networks like, Public Switched Telephone Networks (PSTN), Integrated Switched Digital Networks (ISDN), TCP/IP and etc. From perspective of MCS layer, it presents a uniform OSI transport interface and services (X.214/X.224). Built-in error correction facilities are also included in this layer;

4. T.124:The T.124 specifies the Generic Conference Control (GCC), which provides a comprehensive set of facilities for establishing and managing the multi-point conference;

5. T.125: T.125 defines: 1) Procedures for a single protocol for the transfer of data and control information from one MCS provider to a peer MCS provider. 2) The structure and encoding of the MCS protocol data units used for the transfer of data and control information.

3.2. RDP

3.2.1. Introduction

Remote Desktop Protocol, which is issued by Microsoft, provides remote display and input capabilities over network connections for Windows-based applications running on a server. It is based on, and is an extension of, the T-120 family of protocol standards. So it inherits corresponding capabilities and benefits from T.120, such as the architectural features necessary to support multipoint and multipoint data delivery, which allows data from an application to be delivered in "real-time" to multiple parties without having to send the same data to each session individually.

3.2.2. RDP protocol stack

3.2.2.1. Basic architecture

As RDP is an extension of the core T.120 protocol, it re-uses services from T.123, T.122, T.125 and T.124 of T.120 protocol suite. In T.120 protocol architecture, RDP is at the same place of application protocols. It defines application PDU for information exchange between server and client, like capability set, bitmap update, keyboard/mouse event and so on.

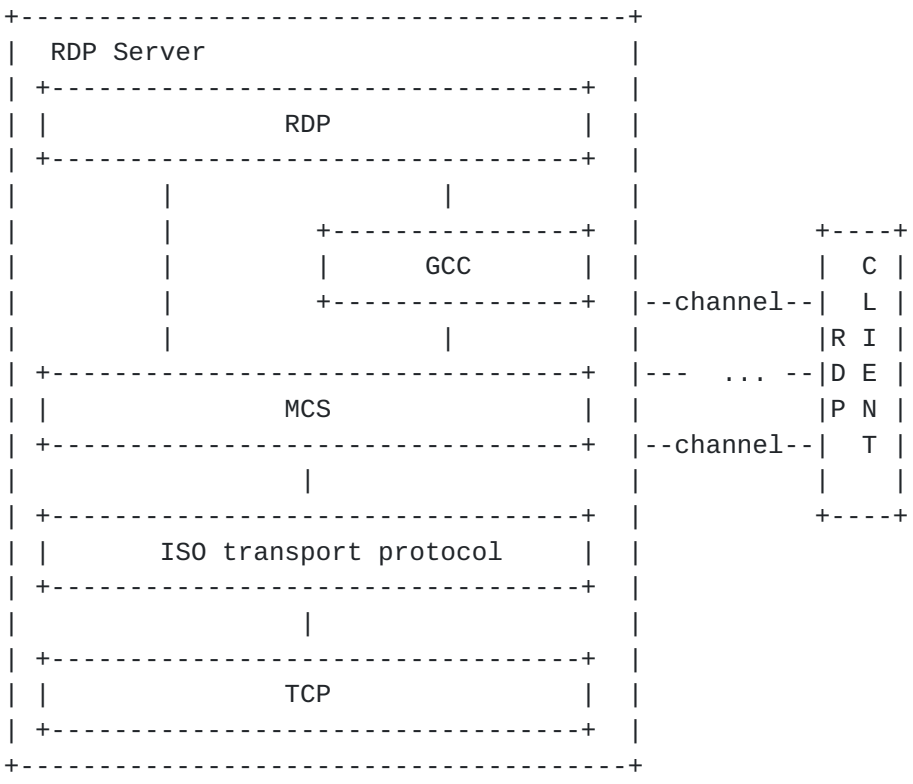

Figure 6 RDP protocol

RDP uses channel service provided by MCS layer for lossless communication between client and server components over the main RDP data connection. Each channel is targeted for one specific application data, such as printer redirection, clipboard redirection and etc.

There are two types of virtual channels in RDP: static virtual channel and dynamic virtual channel. A maximum of 31 static virtual channels can be created at connection time. The list of desired virtual channels is requested and confirmed during the Basic Settings Exchange phase of the server-client connection sequence and the endpoints are joined during the Channel Connection phase. Virtual channel data is application-specific and opaque to RDP. Each virtual

channel acts as an independent data stream. The client and server examine the data received on each virtual channel and route the data stream to the appropriate endpoint for further processing. Dynamic virtual channel is put forward to solve the limitation of static virtual channel number. It is implemented on top of a static virtual channel named DRDYNVC. In the DRDYNVC static virtual channel, different services could be allocated and deleted dynamically.

RDP supports two security classifications: standard RDP security and enhanced RDP security:

1. For standard RDP security, client and server use matched encryption and decryption keys for data security. The encryption and decryption keys are generated during connection sequence, based on random number generated on server and client. The client and server both generate a 32-byte random value using a cryptographically-safe pseudorandom number generator. Then the server sends the random value which it generated (along with its public key embedded in a certificate) to the client in the Server Security Data during the Basic Settings Exchange phase of the connection sequence. After receiving that, the client sends its random value to the server (encrypted with the server's public key) in the Security Exchange PDU as part of the RDP Security Commencement phase of the connection sequence. As the client doesn't authenticate server credential, this security method could not protect data from man in-the-middle. It still exists in RDP for backward compatibility;

2. For enhanced security, RDP traffic is no longer protected by using the techniques described in standard RDP security. Instead, all security operations (such as encryption and decryption, data integrity checks, and server authentication) are implemented by external security protocols, like TLS. In this way, there is no longer need to manually implement protocol security mechanisms, which is replaced with well-known and proven security protocol packages to provide end-to-end security.

Server redirection procedure is defined in RDP for load-balancing scenarios. A client connection can be redirected to a specific session on another server by using the Server Redirection PDU. In the Server Redirection PDU, necessary information is include, such as session ID, redirection information flag, user credential information, load balance information, target server address and etc. When the client receives this redirection indication, it will disconnect from current server and try to connect target server, which is specified in Server Redirection PDU.

In RDP framework, it's flexible to define more service using separate

static/dynamic channel.

3.2.2.2. Bandwidth saving

RDP uses a bulk compressor to compress virtual channel data and some data in PDUs sent from server to client. During information exchange phase of the connection sequence, server and client negotiate the bulk compressor. One version of the bulk compressor (the RDP 4.0 bulk compressor) is based on the Microsoft Point-To-Point Compression (MPPC) Protocol and uses an 8 kilobyte history buffer. A more advanced version of the compressor (the RDP 5.0 bulk compressor) is derived from the RDP 4.0 bulk compressor, but uses a 64 kilobyte history buffer and modified Huffman-style encoding rules. Besides employing bulk compression for generic data, RDP also uses variations of run length encoding (RLE) rules to implement compression of bitmap data sent from server to client. "NSCodec Extension" and "RemoteFX Codec Extension" (lossy image codec) specify the image codec that can be used to encode screen images by utilizing efficient and effective compression.

As RDP is one extension of T.120 protocols, each RDP packet will be wrapped with corresponding headers defined in T.120 protocols, like TPKT header, X.224 Class 0 Data TPDU and etc. In some situations, for example point-to-point connection, these additional overhead is not necessary. In RDP protocol, it defined two choices: slow-path which follows strictly T.120 protocols and fast-path with the goal of improving bandwidth. For fast-path, the TPKT Header, X.224 Class 0 Data TPDU, and MCS Send Data Indication are replaced. The Security Header is collapsed into the fast-path output header, and the Share Data Header is replaced by a new fast-path format. The contents of the graphics and pointer updates are also changed to reduce their size, particularly by removing or reducing headers.

The real-time display interaction is accomplished by continuously sending updated bitmap images from server to client. Even though these bitmaps may be compressed, it is still not a bandwidth efficient mechanism to employ, especially when dealing with graphics-intensive applications that refresh regularly. "GDI Acceleration Extension" aims to reduce the bandwidth associated with graphics remoting by encoding the drawing operations that produce an image instead of encoding the actual image. For example, instead of sending the bitmap image of a filled rectangle from server to client, an order to render a rectangle at coordinate (X, Y) with a given width, height, and fill color is sent to the client. The client then executes the drawing order to produce the intended graphics result. In addition to defining how to encode common drawing operations, "GDI Acceleration Extension" also facilitates the use of caches to store drawing primitives such as bitmaps, color tables, and characters.

The effective use of caching techniques helps to reduce wire traffic by ensuring that items used in multiple drawing operations are sent only once from server to client (retransmission of these items for use in conjunction with future drawing operations is not required after the item has been cached on the client).

For video and audio application, "Video Redirection Virtual Channel Extension" and "Audio Output Virtual Channel Extension" define video and audio delivery message and procedure between server and client. Before stream traffic starts, server and client will exchange their own capabilities, also for media format. After successful negotiation, media stream data will be sent between server and client with both agreed format, for example MPEG-2, instead of transferring raw data.

3.2.2.3. User experience improvement

For VDI applications, user experience is very important for more widely acceptance of this technology. When using client component to access desktop on remote side, the user hopes to get experience of using local computer, such as real-time interactive, plug-n-play service, bi-directional multimedia services and etc.

In RDP protocol suite, it uses techniques mentioned in previous "bandwidth saving" section to improve interactive capability and multi-media services.

Following extension protocols are defined to mitigate environment experience of server and client:

1. Clipboard Virtual Channel Extension: specifies how to keep two distinct system clipboards in sync so that at any given time, the data available to an application on one computer (via its local clipboard) is identical to the data available to another application on a remote computer (via its local clipboard);

2. Remote Programs Virtual Channel Extension: provides support to present a remote application (running remotely on the server) as a local user application (running on the client machine). It extends the core RDP protocol to deliver this seamless windows experience;

3. File System Virtual Channel Extension: provides support to redirect access from the server to the client file system. In a typical terminal server scenario, many of the nonvolatile resources used by the server (such as hard drives, flash drives, and floppy disks) are located on the client. The server exposes a file system driver that is visible to server-based applications

as a hard drive, which allows the applications to access the client file systems;

4. Serial Port Virtual Channel Extension: provides support to redirect serial and parallel ports from client to the server. This allows the server to access client ports as if the connected devices were local to the server;

5. Print Virtual Channel Extension: provides support to redirect printers from client to the server. This allows the server access to printers physically connected to the client as if the devices were local to the server;

6. Smart Card Virtual Channel Extension: is designed to remotely execute requests on a client's Smart Cards for Windows. When Smart Card Redirection is in effect, server application smart card subsystem calls are automatically remapped to the client"Cside Smart Cards for Windows, which will then receive the corresponding request;

7. Plug and Play Devices Virtual Channel Extension: This protocol is used to redirect Plug and Play (PNP) devices from client to the server, for example USB devices. This allows the server access to devices that are physically connected to the client as if the device were local to the server.

3.3. SPICE

3.3.1. Introduction

SPICE was originally developed by Qumranet, which was acquired by Red Hat in 2008. Later Redhat open sourced its SPICE hosted virtual desktop protocol. Now open-source SPICE project aims to provide a complete open source solution for interaction with virtualized desktop devices. The SPICE protocol, acting similar functions as Microsoft RDP, provides client and server communication to support virtual desktop infrastructure.

3.3.2. SPICE Protocol

3.3.2.1. Basic architecture

SPICE protocol defines a set of protocol messages for accessing, controlling, and receiving inputs from remote computing devices (e.g., keyboard, video, and mouse) across networks, and sending output to them. SPICE uses simple messaging and does not depend on any RPC standard or a specific transport layer.

From architecture perspective, SPICE is architected a bit different than RDP and ICA. While RDP and ICA are made of up two components (a remote software component that runs in the OS of the Windows host you're connecting to, and a client), SPICE is actually made up of three components:

1. Remote guest component: A virtual graphics adapter running in the VM, just like RDP/ICA

2. Client component; The SPICE client software, just like RDP/ICA;

3. Remote host component: A virtual graphics device which the hypervisor makes available to the VM.

In other words, because SPICE has a hypervisor component, it will only work when your remote hosts are VMs.

Similar with RDP, SPICE protocol spilt the communication session into multiple communication channels. Currently the following communication channels are defined in the protocol: a) the main channel serves as the main spice session connection; b) display channel for receiving remote display updates; c) inputs channel for sending mouse and keyboard events; d) cursor channel for receiving pointer shape and position; e) Playback channel for receiving audio stream; f) Record channel for sending audio capture. More channel types will be added as the protocol evolves. Since RDP is based on T.120, it could use channel service from lower layer. But for SPICE, it has to do channel management along with application protocols, which may lead to some implementation complexity.

For user authentication in SPICE, the server generates a 1024 bit RSA key and sends the public part to the client (via RedLinkInfo). Client uses this key to encrypt the password and send it back to server (after RedLinkMess). Server decrypt the password, compare it to ticket.

SPICE doesn't define encryption mechanism in application layer, which means that it depends on external security protocols for security guarantee so that having maximum flexibility in choosing an encryption method.

SPICE defines full server redirection primitives to support live migration. RDP supports server redirection for load balance purpose, which means server redirection only occurs during 1st server access. SPICE could support server redirection during run-time and corresponding migration process control.

3.3.2.2. Bandwidth saving

SPICE uses different compression strategy for different media resource. It uses QUIC/LZSS/GLZ for lossless image compression. QUIC (based on SFALIC) and GLZ (based on LZSS with a history-based global dictionary) are Redhat proprietary algorithms. As the video/audio is played using different media player in the virtual machine, it's not easy to send raw media data to client directly. In current SPICE architecture, the video/audio media file firstly is decoded in virtual machine, then captured by SPICE and sent to client using algorithm MJPEG for video and CELT for audio.

Similar with RDP, SPICE supports transmission 2D graphic primitives to client to fully re-use client computing resource and save bandwidth. In current SPICE implementation, SPICE server also does optimization for graphic commands by building up one graphic command tree and removing un-necessary commands (for example, it is hidden by other commands) to further saving bandwidth.

Caching for pixmaps, palettes and cursors are also defined in SPICE in order to avoid redundant transmissions to the client. In current implementation, the server is responsible for cache management, such as add or remove from the cache. The client cache size is set by the client and transferred to the server through the display channel initialization message. The server monitors the current cache capacity and when it lacks space it removes the least recently used cache items until there is enough available cache space. The server sends an invalidate command with these items and the client removes them. In rendering-related messages, corresponding cache id will be specified instead of carrying over actual image data.

3.3.2.3. User experience improvement

SPICE defines a framework to support interactive operation between client and server based on techniques described in pervious section. For service like device redirection, SPICE protocol currently has no definition. But SPICE community has plan and is going on for following:

1. Network tunneling: using virtual network interface to enable sharing of network resources. Currently the focus is on printer sharing but is not limited to that;

2. USB sharing: allows clients to share their USB devices with SPICE servers. USB sharing is currently supported already in the official RHEV versions, through a proprietary component. An open source replacement for this is coming along nicely.

3.4. ICA

3.4.1. Introduction

Independent Computing Architecture is a proprietary protocol for an application server system, designed by Citrix Systems. The protocol lays down a specification for passing data between server and clients, but is not bound to any one platform. The ICA protocol is used within several of Citrix's products to provide desktop virtualization across many different platforms. The three specific Citrix products utilizing ICA are WinFrame, XenApp and XenDesktop. The main purpose of these products is to allow a centralized set of Microsoft Windows based servers to deliver Windows applications/ virtual machines to users on various platforms such as Linux, UNIX, MacOS and even Windows CE.

3.4.2. ICA Protocol

3.4.2.1. Basic architecture

From perspective of 7-Layer OSI model, ICA is one protocol on Presentation layer, which is defined for services like, data conversion, compression, decompression, encryption, decryption and etc.

Well known ports 1494/UDP/TCP and 2598/UDP/TCP are the two primary ports used for the ICA protocol. Over these ports, the ICA protocol handles the transmission of data between the client and the server. Data transfer from the client to the server generally includes mouse movement, events and other requests. Data transfers from the server to the client generally involve a high-level view of the display rather than a full image of the screen, dramatically saving bandwidth and reducing the dependency on latency.

ICA protocol also uses channel-based architecture. It is comprised of 32 virtual channels and each channel with a default priority. QoS is provided based on prioritization and less important data is sacrificed if necessary.

3.4.2.2. Bandwidth saving and User experience improvement

Since ICA is Citrix's proprietary protocol, more detailed information like how compression is defined is Citrix's secret.

From Citrix's product view, Citrix provides HDX technology to deliver a "high definition" desktop virtualization user experience to end users for any application, device or network. These user experience enhancements balance performance with low bandwidth - anything else

becomes impractical to use and scale. HDX technology provides network and performance optimizations to deliver the best user experience over any network, including low bandwidth and high latency WAN connections.

3.5. RFB

3.5.1. RFB Protocol

3.5.1.1. Basic architecture

RFB protocol adopts server-client architecture. The remote endpoint where the user sits (typically with a display, keyboard, and pointer) is called the RFB client or viewer. The endpoint where changes to the framebuffer originate (i.e., the windowing system and applications) is known as the RFB server. RFB protocol works at the framebuffer level, which means that pixel information is transferred instead of graphics commands between RFB server and RFB client. In this way, it makes RFB a "thin client" protocol, which has very few requirements of the client. Thus RFB clients can run on the widest range of hardware, and the task of implementing a client is made as simple as possible.

RFB protocol depends on one reliable transport layer. From current practice, it usually operates over a TCP/IP connection. During initialization, client will setup one TCP connection with server. After that, all information is carried over this connection. This is different with SPICE or RDP, which defines several logical channels and each channel has one corresponding TCP connection.

In current RFB protocol, two kinds of information are defined to be exchanged between server and client:

- Inputs: The input side of RFB protocol is based on a standard workstation model of a keyboard and multi-button pointing device. Input events are simply sent to the server by the client whenever the user presses a key or pointer button, or whenever the pointing device is moved. These input events can also be synthesized from other non-standard I/O devices. For example, a pen-based handwriting recognition engine might generate keyboard events.

- Display: The display side of RFB protocol is based around a single graphics primitive: "put a rectangle of pixel data at a given x,y position" whenever the server found there is change in the framebuffer or it is requested by the client explicitly. In existing implementation, there are two practices for the server to find out there is change in framebuffer: a) the server periodically pulls the data from the framebuffer and compares it

with previous one. Obviously this is computing resource consuming and not effective; b) The server utilizes mirror driver functionality in Windows OS. It depends on notification from mirror driver whenever there is change in framebuffer.

For RFB protocol, it has three ways for extension: a) adding new encodings or pseudo encodings; b) adding new security types; c) adding new message definitions. The former two methods work like the plug-in mechanism and will not result in RFB protocol version upgrade.

In RFB protocol, only VNC authentication method is defined. The work flow is: 1) The server sends a random 16-byte challenge; 2) The client encrypts the challenge with DES, using a password supplied by the user as the key. To form the key, the password is truncated to eight characters, or padded with null bytes on the right; 3) The server verifies the response from client. Beside this default security method, the client and server might also agree to use an extended security type to encrypt the session, or the session might be transmitted over a secure channel such as IPsec or SSH.

3.5.1.2. Bandwidth saving and User experience improvement

As described before, the display side of RFB protocol is based around a single graphics primitive: "put a rectangle of pixel data at a given x,y position". This might seem an inefficient way of drawing many user interface components. However, allowing various different encodings for the pixel data gives us a large degree of flexibility in how to trade off various parameters such as network bandwidth, client drawing speed, and server processing speed.

Current encoding algorithms defined in the standard includes: CopyRect Encoding, RRE Encoding, Hextile Encoding, TRLE and ZRLE. The core idea of these encodings is to combine different compression techniques, like palettization/run-length encoding/zlib, to compress the pixel data.

4. Conclusion

Through the survey of several VDI products, we found the common key components of VDI architecture:

1. Desktop client: The software installed on client devices. It connects to virtual desktop over protocols such as RDP, ICA and SPICE;

2. Virtual desktop agent: The agent is installed on virtual machine for accepting the client's connection. Not all solutions have same architecture. For instance, in Redhat's VDI solution, SPICE server , which runs on host machine, plays this role and is shared by all virtual machines on the server;

3. Virtual machine pool: Virtual machine pool provides computing resource for virtual desktops. It launches new virtual machine when new client connects to it. Virtual machine pool could contain huge numbers of virtual desktops,so the pool should be highly scalable and reliable;

4. Desktop connection broker. The broker manages desktop connections. It chooses the best virtual machine for the user according to some policy (such as load balance) and directs the connection to virtual machine;

5. Access Gateway. Responsible for external client's access. It authenticates external users and provides secure connections between client and virtual desktop;

6. Authentication Server: Provides authentication for users. It can be directory server or others.

To optimize user experience, all solutions adopt a set of technologies, such as multimedia compress, image cache, device redirection, client rending and adaptive bandwidth management. These technologies ensure that virtual desktop users feel the desktop is running locally.

There are two modes about how the client connects to virtual desktop:

1.The client connects to agent which is running on virtual machine. In Citrix, Microsoft and VMware's solutions, there is a virtual desktop agent on each virtual machine. The agent opens protocol ports to wait for client's connection, and exchange information with the client. This modes is shown in Figure 7:

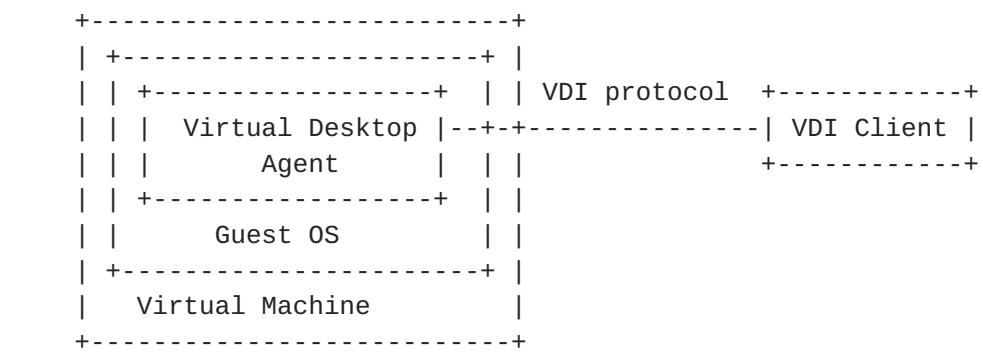

Figure 7 Agent on virtual machine

2.The client connects to agent which is running on host. In Redhat's solution, for each virtual machineGBP[not]there is a SPICE server playing the role of virtual desktop agent, which is installed on host, instead of being installed in every virtual machine. The SPICE server captures the virtual machine's screen information, and transfers it to the client. This mode is shown in Figure 8

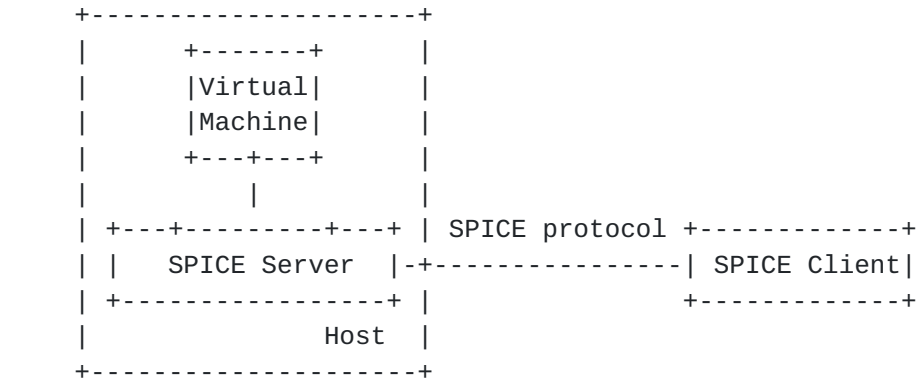

Figure 8 Agent on host

For different VDI protocols, they have similarities from perspective of functionality:

1. Support experience of operating local-computer when accessing remote server, through defining multi-channels and each channel is for one specific data type, like display/audio/input;

2. Enhance user experience through bandwidth saving techniques and more value-added services like Plug-n-Play: 1) define framework to support client and server negotiation of compression method and cache management; 2) define different protocol extensions to support different services, like device redirection; 3) define framework for graphic command transmission for 2D and 3D

applications instead of large-size image.

Looking inside each protocol, they are different with each other, especially for:

1. Since they are proposed by different vendors and targeted for different purpose at initial time, detailed message and procedure are totally different, even for the architecture. For example, RDP is based on T.120, while SPICE is self-contained and ICA is proprietary;

2. Compression codec definition in each framework is incompatible. For example, NSCodec and RemoteFX codec are defined in RDP framework for image compression, while SPICE uses GLZ/LZ/QUIC/ JPEG to compress the image;

3. Service definition completeness of each protocol is different. RDP protocol currently contains rich extension for some services, like file-system and device redirection, Plug-n-Play and etc. For SPICE, the community is working hard on that to catch it up.

5. Acknowledgements

6. Security Considerations

Related security issues will be addressed in subsequent draft.

7. References

7.1. Normative References

[ITUT120] ITU, "ITU T.120", 2009, <http://www.itu.int/rec/T-REC-T.120/en>.

[MCPP] Microsoft, "Windows Communication Protocols webpage", 2011, <http://msdn.microsoft.com/zh-cn/library/cc216513>.

[MSRDS] Microsoft, "Microsoft RDS webpage", 2011, <http://www.microsoft.com/windowsserver2008/en/us/rds-product-home.aspx>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[SPICE] Redhat, Spice project., "Spice remote computing protocol

definition v1.0", 2009.

[VMWARE] VMware, "VMware View webpage", 2011, <http://www.vmware.com/products/view/>.

[XENDESKTOP] Citrix, "Citrix XenDesktop webpage", 2011, <http://www.citrix.com/xendesktop>.

7.2. Informative References

[RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, June 1999.

Authors' Addresses

Suan Ma (editor)
ZTE Corporation
No.68 Zijinghua Road
Nanjing,
CN

Phone: +86 15950586953
Email: ma.suan@zte.com.cn

Liang Liang
ZTE Corporation
No.68 Zijinghua Road
Nanjing,
CN

Phone: +86 18913825360
Email: liang.liang12@zte.com.cn

Jun Wang
ZTE Corporation
No.68 Zijinghua Road
Nanjing,
CN

Phone: +86 13770604455
Email: wang.jun17@zte.com.cn