Virtual Desktop Infrastructure Problem Statement
draft-wang-appsawg-vdi-problem-statement-01

Abstract

The Virtual Desktop Infrastructure is a technology to separate local desktop and remote computing/storage resources, which was initially derived from the remote desktop administration but with new business models and very different use cases. Most of existing VDI systems are based on proprietary implementation, and positioning different market with different features. Since virtual desktop technology is believed to be a mainstream application delivery method, like http protocol against web applications, so it’s important to make the virtual desktop access protocol open and standard. This draft summarizes the limitations of existing virtual desktop systems, and proposes the intent standardization work in IETF.

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 January 1, 2012.

Copyright Notice

Copyright (c) 2011 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents
Table of Contents

1. Introduction .................................................. 3
   1.1. Requirements Language .................................. 3
   1.2. Terminology and Abbreviation ........................... 3
2. Virtual Desktop System Architecture ......................... 4
   2.1. Common Framework ...................................... 4
   2.1.1. VDI server in the VMM ............................... 5
   2.1.2. VDI Server In The Guest OS .......................... 6
   2.2. Deployment System Architecture ....................... 6
   2.3. VDI Protocol Stack Reference Architecture ............ 8
3. Application scenarios ...................................... 11
   3.1. scenario 1: Enterprise IT Application ................. 11
   3.2. scenario 2: Cloud Hosted Virtual Computer ............ 11
   3.3. scenario 3: Cloud Hosted Telecommunication Terminals .. 12
   3.4. scenario 4: Alternative SaaS delivery method .......... 12
4. Known Issues ................................................. 12
5. Intended Work In IETF ........................................ 14
6. Acknowledgements .............................................. 14
7. Security Considerations ..................................... 15
8. References .................................................... 15
   8.1. Normative References .................................. 15
   8.2. Informative References ................................ 15
Authors’ Addresses ............................................ 15
1. Introduction

The Virtual Desktop technology is a combination of remote desktop control and virtualization. Client can access a remote virtual desktop or a remote application through VDI protocol. Currently, there are a lot of VDI system vendors, including Citrix, Microsoft, VMware etc., each of them provides bundle of components based on proprietary implementation, it’s difficult to interwork between different vendors. In addition, the existing solutions focus on enterprise scale application, we haven’t found any one designed for public virtual desktop service offering.

Since the VDI will be an alternative application delivery method that is even more powerful than web, it can benefit a lot from a unified open and standard protocol, which ensures any qualified client can access any qualified server system. Thinking about WEB, without standardized http and other related protocols, its long-lasting property could have been unimaginable.

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

- **OS**: Operation System
- **Guest OS**: An OS runs on top of a virtual machine manager.
- **VDI**: Virtual Desktop Infrastructure.
- **VDI agent**: software module running in Guest OS to perform special tasks, such as Clipboard sharing between Guest OS and VDI client. Specific functionality depends on VDI implementation.
- **VDI server**: software module which implements VDI protocol stack and runs in the server side. Sometimes it is also called as Virtual Desktop Agent.
- **VDI client**: software module which implements VDI protocol stack and runs in the client side. Sometimes it is also called as VDI viewer.
- **VM**: Virtual Machine, contains at least one guest OS, usually several additional application processes.
2. Virtual Desktop System Architecture

2.1. Common Framework

A VDI system includes at least one client and one server. VDI adopts the client-server model, in which the client runs a VDI viewer software which connects to the server through virtual desktop access protocol. Unlike the remote administration, VDI clients always connect to a dedicated virtual machine and don’t share the desktop with others. The service provider or the enterprise’s IT department must prepare exclusive VM image and virtual storage space for each client, and keep track of the clients’ modification on desktop and storage.

Depend on the VDI system implementation, the VDI server module lies in either the VMM or the guest OS. Accordingly, the VDI systems can be divided into two categories.
2.1.1. VDI server in the VMM

A VDI back-end system makes up of a VDI server module running inside the VMM and some virtual devices drivers. The VDI server is the virtual desktop access protocol processing endpoint on the server side, and also simulates several necessary virtual devices such as graphic card, usb devices and sound card. The corresponding drivers are installed into the guest OS, so the I/O operation can be pass-through along the application - guest OS - virtual driver-virtual device-VDI server - virtual desktop access protocol - VDI client path.

The different I/O information encapsulates into respective transport channel of the protocol, normally, including control, screen, mouse/keyboard input, audio input, audio output and usb devices channel. Any screen change of the server will be sent back to the client which is in charge of re-rendering the information to the client screen; similarly, the mouse, keyboard, audio input will be sent from client to the server and audio output, usb writing operation at the server sent to the client. All of these VDI system behaviors make sure that the user gets the same experience as operating his local computers.
while accessing remote virtual desktop.

2.1.2. VDI Server In The Guest OS

Compared to the first architecture, the only difference is virtual desktop server be installed in the guest OS. As a result, the connection terminates in the guest OS rather than in the VMM, so the client can’t communicate with its virtual desktop until the guest OS finished startup.

Sometimes, The virtual desktop is also called ‘virtual desktop agent’, which performance the functions of both the virtual device and virtual driver modules, plus other necessary function.

2.2. Deployment System Architecture

A typical virtual desktop system comprises of clients, access gateways or connection brokers, authentication server, virtual desktop server pool and virtual application server, some additional component will also be deployed if extra features be required.
Figure 3 virtual desktop system deployment architecture

- VDI Clients, usually run a virtual desktop client software, sometimes they’re called viewer or receiver. The client software performs service discovery, resource connection and virtual desktop access through virtual desktop access protocol.

- Access Gateway, which is a virtual desktop infrastructure entry point. If the client lies in the public network, the access gateway is used to authenticate the client and perform other security related function. In some specific VDI systems, the access gateway is replaced with redirector, which directs the traffic to a virtual desktop resource instead of forcing all of the traffic going through a designated gateway.

- Authentication server, which stores required security information, responsible for authenticating the clients. Most of the VDI systems reuse existing facilities such the windows’ Active Directory.

- Authorization server, it contains the VDI subscriber’s profile information, including the virtual desktop cpu number, memory and disk quota, application preference, etc. Sometimes, the authorization server is merged into authentication server, e.g., the active directory or Home subscriber server defined in 3GPP. In certain case, the authorization server may also act as resources allocating server, accepting the authorization request from the user or access gateways and checking if the user has the rights to access certain services. If the result is positive, the server sends a virtual resource allocating request to the resource...
pool management server. After the virtual desktop resource successfully allocated, the user will be directed to connect the designated resource.

- virtual desktop server pool, they’re a bunch of computing and storage resources organized by virtualization technology. The Management server in charge of monitoring the VM state and allocating available resource to the user. Virtual machines are modified for desktop virtualization. Depending on the VDI business model, the scale of resources pool may vary from hundreds to millions virtual machine.

Not every vendor provides aforementioned components. Some of them doesn’t implement access gateway, e.g., Redhat provides a web portal to redirect the authenticated client to the designated VM. Most of the vendors support application virtualization and application streaming, in which the application software are installed on some dedicated servers instead of keeping a copy on every virtual desktop machine, reducing the disk space requirement significantly and making the virtual machine mobility simpler.

Since the VDI systems intend to realize a ‘virtual real desktop’ for the client, besides the screen, mouse and keyboard input/output information transfer, most of them support the I/O redirection features, which means the remote virtual desktop can access the client’s I/O devices, including audio and usb equipment, just like the devices have been installed at the remote servers.

2.3. VDI Protocol Stack Reference Architecture

Figure 4 provides one reference architecture for VDI protocol stack. From a high level view, it has Session Control Layer, Virtual Channel Layer and Application Layer above traditional Transport Layer. Each layer provides services to its upper layer while receiving services from the layer below. One thing needed to be mentioned here is that not all layers in this figure are in the scope of VDI protocol. VDI protocol will focus only on Session Control Layer and Virtual Channel Layer. For the other two layers, they are listed here for completeness and better understanding.
o Transport Layer

VDI protocol doesn’t include transport protocol, but depends on the service provided by transport layer. It will rely on: 1) reliable connection-oriented service from transport layer to deliver control and sequential information, like graphics commands; 2) fast connection-less service from transport layer for delivering multimedia data; 3) security services from transport layer for data encryption, integrity and authentication. So any transport protocol that could satisfy above three requirements could be fit for VDI and they may be, but not limited, TCP/UDP/TLS/DTLS.

o Application Layer

Application Layer utilizes service from different VDI channels and provides VDI services to end-user. Its relationship with VDI protocol is somewhat like the relationship between Web browser (such as IE, Firefox) and HTTP protocol. It is also not in the scope of
VDI protocol.

- Session Control Layer

Session Control Layer is mainly responsible for controlling the dialogues (connections) between VDI server and VDI client and general VDI services.

1. Channel Control: Functionalities of Channel Control include channel connection setup, deletion and maintenance. From implementation perspective, it will provide Service Access Point (SAP) and hook mechanism to virtual channel layer for these services. In this way, it isolates virtual channel layer from detailed connection management and provides foundation services for virtual channel plug-in. Also virtual channel layer will not deal with IP address directly, which makes session mobility easier.

2. User Authentication: User Authentication is responsible for authentication between VDI server and VDI client during connection setup. It provides one framework to exchange control information, for example algorithms negotiation. For detailed authentication algorithms, they are not included in the scope of VDI protocol.

3. Session Mobility: As indicated before, virtual channel layer will deal with unique name of end points, instead of network address, like IP. The mapping management between the unique name and IP address is done in Session Mobility. Whenever IP address of end-point is changed due to handoff between different networks (for example, between LTE and WiFi), it will notify this change to the other end-point. Channel Control will re-establish all the connection, which is transparent to virtual channel layer. In this way, continuous service is guaranteed.

4. Redirection: Redirection is responsible for user redirection. When the user first accesses the VDI server, the server may redirect the user to another VDI server because the virtual desktop has been migrated away. This functionality could be used in scenarios like load balance, user roaming scenarios and etc.

- Virtual Channel Layer

Different information of VDI desktop is encapsulated into different channels. Adopting this multi-channel architecture is due to different service requirements of different channels, such as security, priority (QoS) and etc. Basic channels include:
1. Screen Channel: This channel is responsible for delivering desktop graphics information from VDI server to VDI client. The content in the message may include: graphics commands, bitmaps, cache ID and etc.

2. Input Channel: This channel is responsible for exchanging input information between VDI server and VDI client. VDI client sends keyboard/mouse event to the server, while the server transfers cursor shape and pointer position to the client.

3. Audio Channel: This channel is responsible for exchanging audio information between VDI server and VDI client, including audio playing and recording.

4. USB Channel: This channel is responsible for redirecting USB devices which plugs in the VDI client to the VDI server, so that user could operate USB devices in the VDI desktop.

5. Smart Card Channel: This channel is responsible for redirecting Smart Card devices which plugs in the VDI client to the VDI server, so that Smart Card based authentication could be performed.

Besides above channels, user could provide channels with specific functions as the plug-in. This is achieved through utilizing Channel Control services.

3. Application scenarios

3.1. scenario 1: Enterprise IT Application

The enterprise built a centralized virtual desktop resource pool, where each of the office staffs has been designated at least a VM. The employees run the office software and access the enterprise application through their virtual desktop. Any devices, e.g., thin clients, laptops, tablets and smartphones, can be used as virtual desktop clients without extra cross-platform software porting costs. Storing sensitive data at local computer is restricted so that information leakage risk can be minimized.

3.2. scenario 2: Cloud Hosted Virtual Computer

Virtual Computer service providers provide VM to their customers who access their VM via virtual desktop protocol. The subscription fees depend on the contracted computing resources, such as CPU number, memory capacity, disk quota, and their usage statistics.
The subscribers can access their virtual computer via any devices and any media, the benefit for them includes: 1) Reduction of the devices hardware upgrading period, thus hardware cost can be cut down. 2) Keeping data, documents synchronized among their devices. 3) Installing software once and using them anywhere and on any devices.

3.3. scenario 3: Cloud Hosted Telecommunication Terminals

The carriers supply the virtual desktop host service to their customers, which allows the clients to use the telecommunication application similar to local phones, such as voice/video calls, messages and address book, and furthermore, keep the same phone number as their real phones.

The benefits of this kind of business model are: 1) Multi-screen service convergence, which means the users can access the same application and content via different screens and get the same experience, and can migrate the services among different devices in realtime. 2) Get rid of the terminals’ hardware limitations, one can even play a complex 3D game on a ordinary smartphone. Of cause, the smart phone only act as an I/O device while game is running on the remote VM. 3) Accelerating the time-to-market period of new telecom services. Formerly, if the new application impacts on the terminals’ software, the operator must persuade the users upgrading their devices, this would spend very long time. But if the terminals were hosted in carrier’s network, the upgrading can be done centralized in a short time.

3.4. scenario 4: Alternative SaaS delivery method

Till now, most of the SaaS delivered by WEB technology, but it’s still a hard task to realize some sophisticated applications, such as graphical design and 3D game, in web pages. These cons of WEB are just the pros of virtual desktop. By using virtual desktop, SaaS can be extended to any software products.

4. Known Issues

At the very beginning, the virtual desktop solution targeted the enterprise market, so the office software support is a priority. Therefore, most VDI product, especially the Microsoft’s RDS/RDP product family, are highly optimized for the office software. Our test results show that RDP session only produces tens of kilobytes per second traffic while running Microsoft’s word or excel software with regular document reading or editing operation, as well as the internet surfing.
Some of the VDI vendors claimed that their systems have been optimized for multimedia applications, however, the test results reveal that they are not fully optimized as their assertion especially under low-speed access environment. We have tested the streaming application performance at the platform based on Redhat’s spice and Microsoft’s RDP protocol. The spice protocol chooses M-JPEG as the streaming data compression algorithm, and the test results disclose that it has obvious better performance compared to Microsoft’s RDP+terminal services combination. Despite the optimization efforts done by Redhat, the spice still consumes 12 times access bandwidth compared to the original bitrates of a standard definition video, that’s totally not acceptable while accessing the services via carrier’s wireless network.

M-JPEG does not use inter-frame encoding, which reduces the processing overhead but results in lower compression ratio. MPEG-4/h.264’s compression ratio is about 5-10 times compared better than M-JPEG, but requires more powerful CPU or GPU hardware. Recently, even the low end desktop or laptop systems have been equipped with integrated GPU capable of HD MPEG-4 video decoding. At the mobile sector, most of the 3G mobile phones have the h.264 decoding chipset installed, and the high-end handsets have shipped with an integrated GPU, e.g., iPhone 4/iPad.

Secondly, the audio streaming overhead also need to be cut down. The widely adopted audio codec in mobile network is AMR or it’s wideband version – the AMR-WB, cost bandwidth about 6.6 to 23.85 kilobits per second, if encoded within rtp, the bitrate ranges between 23kbps and 40kbps with 20ms ptime parameter. In contrast, the regular audio streaming in the VDI system consumes 128kps or higher bandwidth. the excess overhead can not be neglected in the mobile network.

Thirdly, assuming that the providers use VDI technology to deliver services, they purchase server hardware and software from different vendors for some commercial considerations, but still hope to provide the service to any potential customers whatever the device they used to access the network. So all of the participants would benefit from the fact that any VDI client device running any vendor’s software can connect to the VDI backend services built by any vendor’s solution, that is, a unified open and standard protocol should be widely adopted. It is true also for migrating virtual machines across different providers. Furthermore, due to the VDI backend system variance, the access security mechanism may totally different, e.g., Active Directory vs. 3gpp sim based access security, so it’d better to separate the security framework from the basic VDI protocol design.

Fourthly, there’re more and more mobile devices connecting to the
network via wireless technology, the mobility management and service continuity should be a fundamental capability of the VDI protocol. Someone may argue that the mobility is the network’s business, but actually, network based mobility can’t provide inter-network IP mobility at most cases, we still need keep eye on the mobility issues while VDI client switches between a 3G network to another carrier’s WiFi hotspot.

5. Intended Work In IETF

- Designing a fundamental virtual desktop access protocol, which enables the interworking between different VDI clients and backends. The protocol should support screen, mouse and keyboard mapping between clients and remote systems as well as audio and USB devices redirection.

- Defining a framework to accommodate advanced video and audio compression algorithm, especially allow the client negotiates intended algorithm with backend system.

- Defining an access security framework, in which the client and server can negotiate preferred security suite. To meet the minimal interworking requirement, some access security mechanism must be defined as a mandatory component, e.g., Pre-shared Key or PKI method, some other authentication mechanism also can be introduced, e.g., the 3gpp SIM based security. If the integrated security also need to be achieved, the protocol should support transfer credential information from clients to servers, e.g., restricted access to the client’s certificates or SIM card from virtual desktop server side.

- Integrating the mobility and service continuity capabilities. To facilitate the user roaming among different network providers, mobility and service continuity should not rely on the underlying network facilities. Another service continuity related issue is caused by the VM live migration in the backend system, the protocol design should also take it into account.

6. Acknowledgements

Many members from different societies and companies provided useful comments and feedback on this document. Thanks for their efforts in reviewing the document and providing valuable comments.
7. Security Considerations

Related security issues will be addressed in subsequent draft.

8. References

8.1. Normative References


8.2. Informative References


Authors’ Addresses

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

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

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

Phone:
Email: ma.suan@zte.com.cn
Liang Liang
ZTE Corporation
No.68 Zijinghua Road
Nanjing,
CN

Phone:
Email: liang.liang12@zte.com.cn