[MS-TSSO]:
Terminal Services System Overview

Release: Friday, February 4, 2011

This document provides an overview of the Terminal Services System Overview Protocol Family. It is intended for use in conjunction with the Microsoft Protocol Technical Documents, publicly available standard specifications, network programming art, and Microsoft Windows distributed systems concepts. It assumes that the reader is either familiar with the aforementioned material or has immediate access to it.

A Protocol Family System Document does not require the use of Microsoft programming tools or programming environments in order to implement the Protocols in the System. Developers who have access to Microsoft programming tools and environments are free to take advantage of them.

Abstract

This document describes The Terminal Services System, a system that enables a remote **client** to display and interact with a **desktop** or application running on a distant **server**. A remote **client** connected to the **server** can use software and resources available to the **server** according to license restrictions. This document describes the relationship of the system of protocols that comprise the Terminal Services System, background information about the system, use cases that exercise the component protocols, abstract data models of system components, internal system architecture, and details about the communications that occur between system components.

Revision Summary

Date	Revision History	Revision Class	Comments
02/27/2009	0.1	Major	First draft completed
04/10/2009	0.2	Minor	Updated the technical content.
05/22/2009	1.0	Major	Updated and revised the technical content.
07/02/2009	1.1	Minor	Updated the technical content.
08/14/2009	2.0	Major	Updated and revised the technical content.
09/25/2009	3.0	Major	Updated and revised the technical content.
11/06/2009	3.1	Minor	Updated the technical content.
12/18/2009	4.0	Major	Updated and revised the technical content.
01/29/2010	4.1	Minor	Updated the technical content.
03/12/2010	4.1.1	Editorial	Revised and edited the technical content.
04/23/2010	4.1.2	Editorial	Revised and edited the technical content.
06/04/2010	4.1.3	Editorial	Revised and edited the technical content.
07/16/2010	4.1.3	No change	No changes to the meaning, language, or formatting of the technical content.
08/27/2010	4.1.3	No change	No changes to the meaning, language, or formatting of the technical content.

Date	Revision History	Revision Class	Comments
10/08/2010	5.0	Major	Significantly changed the technical content.
11/19/2010	5.0	No change	No changes to the meaning, language, or formatting of the technical content.
01/07/2011	6.0	Major	Significantly changed the technical content.
02/11/2011	6.1	Minor	Clarified the meaning of the technical content.

Contents

1 Introduction

This Protocol Family System Document (PFSD) is primarily intended to cover the Protocol Family as a whole. In conjunction with Member Protocol Technical Documents (TDs), which are intended to cover Member Protocols, it presents the rules for information exchange relevant to those Member Protocols and the Protocol Family that are used to interoperate or communicate with a Windows operating system in its various environments.

The Terminal Services System provides protocols supporting secure connection and communication between remote **clients** and **servers**. Using the Terminal Services System, a user of a remote client can initiate a **user session** on a server and then run programs, save files, and use network resources as authorized. The Terminal Services System supports the **hosting** of multiple, simultaneous user sessions on servers.

With the release of Windows 7 and Windows 2008 Server R2, "Terminal Services" is referred to as "Remote Desktop Services", and a "Terminal Server" is now referred to as a "Remote Desktop Server". Component protocol documentation released prior to Windows 7 and Windows 2008 Server R2 still uses the "Terminal Services" terminology.

1.1 Glossary

The following terms are defined in [MS-GLOS]:

client
directory service (DS)
domain
Domain Name System (DNS)
handshake
protocol data unit (PDU)
remote application
Remote Desktop Protocol (RDP)
remote procedure call (RPC)
server
smart card
terminal server
Terminal Services
Transmission Control Protocol (TCP)
tunnel
universal serial bus (USB)

The following terms are defined in [MS-RDPEA]:

virtual channel

The following terms are defined in [MS-RDPEAI]:

dynamic virtual channel
Remote Desktop Protocol (RDP) Client

The following terms are defined in [MS-RDPEDYC]:

static virtual channel

The following terms are defined in [MS-RDPERP]:

remote application integrated locally (RAIL)

The following terms are defined in [MS-RDPCR2]:

desktop

The following terms are defined in [MS-RDPEDC]:

drawing order
sprite
sprite tree

The following terms are defined in [MS-TSGU]:

pipe

The following terms are defined in [MS-RDPBCGR]:

Multipoint Communication Service (MCS)
Network Level Authentication (NLA)
Server Authentication

The following terms are specific to this document:

audio redirection: The transfer of audio data from the TS Server to the RDP Client. For example, when an application running on a TS Server plays an audio file, this protocol is used by the TS Server to transfer the audio data to the RDP Client. The RDP Client may then play the audio.

clipboard redirection: The transfer of data from a remote application running on a TS Server to the local clipboard on the RDP Client, and the transfer of local clipboard data on the RDP Client to a remote application running on a TS Server.

file system redirection: The exchange of data between an application running on a TS Server and the file system of an RDP Client. The TS Server may access, read, or write to the file system of the RDP Client, depending on privileges that have been granted to it.

firewall: A firewall is a software component typically implemented on an Internet gateway device that is a part of a private network. The firewall is configured to either block or allow external access to resources within the private network.

gateway: A node on a network that serves as an entrance to the network for users that are outside the network. The gateway routes communications from external users to resources inside the network.

host: The TS Server running the user sessions, or the VM Host running the virtual machines. The action of creating a venue (a user session or virtual machine) for running a user desktop or applications.

hosting: The assignment, management, and operation of a user-dedicated session on a server for a user accessing the server. For example, when a user runs an application on a server, the application is running within a user session that the server is hosting.

Plug and Play redirection: The transfer of data between a TS Server and a Plug and Play device attached to an RDP Client.

printer redirection: Printer redirection routes printing jobs from a TS Server to a printer that is attached to an RDP Client computer or to a shared printer that is available to the RDP Client.

When a user establishes a remote session with a TS Server, the redirected printer will be available to applications running in that session.

remoting: A server sending graphical data or application data from a server-based application to a remote client.

server farm: A server farm is a group of similarly configured servers which can interchangeably host user sessions, usually depending on a load balancing algorithm.

session broker: A session broker is a software component that assigns users of remote clients to user sessions on servers. A session broker can also use an algorithm to determine what server to use for the user session, balancing the work load between servers. A session broker may reside on the same server as the assigned user sessions, or on a different server.

user session: An abstract venue on a server that is assigned to a user. The user interacts with the server and applications from within this venue.

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as described in [RFC2119]. Note that in [RFC2119] terms, most of these specifications should be imperative, to ensure interoperability. All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

Any specification that does not explicitly use one of these terms is mandatory, exactly as if it used MUST.

1.2 References

This section contains normative and informative references relevant to the Transaction Processing Services System.

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information. Please check the archive site, http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624, as an additional source.

[MS-AUTHSO] Microsoft Corporation, "Windows Authentication Services System Overview", August 2009.

[MS-GPSO] Microsoft Corporation, "Group Policy System Overview", February 2009.

[MS-RDPBCGR] Microsoft Corporation, "Remote Desktop Protocol: Basic Connectivity and Graphics Remoting Specification", June 2007.

[MS-RDPCR2] Microsoft Corporation, "Remote Desktop Protocol: Composited Remoting V2 Specification", December 2008.

[MS-RDPEA] Microsoft Corporation, "Remote Desktop Protocol: Audio Output Virtual Channel Extension", September 2007.

[MS-RDPEAI] Microsoft Corporation, "Remote Desktop Protocol: Audio Input Redirection Virtual Channel Extension", December 2008.

[MS-RDPECLIP] Microsoft Corporation, "Remote Desktop Protocol: Clipboard Virtual Channel Extension", July 2007.

[MS-RDPEDC] Microsoft Corporation, "Remote Desktop Protocol: Desktop Composition Virtual Channel Extension", December 2008.

[MS-RDPEDYC] Microsoft Corporation, "Remote Desktop Protocol: Dynamic Channel Virtual Channel Extension", June 2007.

[MS-RDPEFS] Microsoft Corporation, "Remote Desktop Protocol: File System Virtual Channel Extension", September 2007.

[MS-RDPEGDI] Microsoft Corporation, "Remote Desktop Protocol: Graphics Device Interface (GDI) Acceleration Extensions", June 2007.

[MS-RDPELE] Microsoft Corporation, "Remote Desktop Protocol: Licensing Extension", September 2007.

[MS-RDPEMC] Microsoft Corporation, "Remote Desktop Protocol: Multiparty Virtual Channel Extension", June 2007.

[MS-RDPEPC] Microsoft Corporation, "Remote Desktop Protocol: Print Virtual Channel Extension", September 2007.

[MS-RDPEPNP] Microsoft Corporation, "Remote Desktop Protocol: Plug and Play Devices Virtual Channel Extension", June 2007.

[MS-RDPEPS] Microsoft Corporation, "Remote Desktop Protocol: Session Selection Extension", July 2007.

[MS-RDPERP] Microsoft Corporation, "Remote Desktop Protocol: Remote Programs Virtual Channel Extension", July 2007.

[MS-RDPESC] Microsoft Corporation, "Remote Desktop Protocol: Smart Card Virtual Channel Extension", June 2007.

[MS-RDPESP] Microsoft Corporation, "Remote Desktop Protocol: Serial and Parallel Port Virtual Channel Extension", September 2007.

[MS-RDPEUSB] Microsoft Corporation, "Remote Desktop Protocol: USB Devices Virtual Channel Extension", April 2010.

[MS-RDPEV] Microsoft Corporation, "Remote Desktop Protocol: Video Redirection Virtual Channel Extension", December 2008.

[MS-RDPEXPS] Microsoft Corporation, "Remote Desktop Protocol: XML Paper Specification (XPS) Print Virtual Channel Extension", July 2007.

[MS-RDPNSC] Microsoft Corporation, "Remote Desktop Protocol: NSCodec Extension", April 2010.

[MS-RDPRFX] Microsoft Corporation, "Remote Desktop Protocol: RemoteFX Codec Extension", April 2010.

[MS-TSGU] Microsoft Corporation, "Terminal Services Gateway Server Protocol Specification", June 2007.

[MS-TSTS] Microsoft Corporation, "Terminal Services Terminal Server Runtime Interface Protocol Specification", September 2007.

[MS-TSWP] Microsoft Corporation, "Terminal Services Workspace Provisioning Protocol Specification", December 2008.

[RFC793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981, http://www.ietf.org/rfc/rfc0793.txt

[RFC1035] Mockapetris, P., "Domain Names - Implementation and Specification", STD 13, RFC 1035, November 1987, http://www.ietf.org/rfc/rfc1035.txt

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, http://www.ietf.org/rfc/rfc2119.txt

[RFC2246] Dierks, T., and Allen, C., "The TLS Protocol Version 1.0", RFC 2246, January 1999, http://www.ietf.org/rfc/rfc2246.txt

[SSL3] Netscape, "SSL 3.0 Specification", http://tools.ietf.org/html/draft-ietf-tls-ssl-version3-00

If you have any trouble finding [SSL3], please check here.

1.2.2 Informative References

[MS-GLOS] Microsoft Corporation, "Windows Protocols Master Glossary", March 2007.

[MS-RPCH] Microsoft Corporation, "Remote Procedure Call over HTTP Protocol Specification", January 2007.

2 Overview

Section 1, "Introduction", describes this **Protocol Family** System Document. This section introduces the system that is being documented.

2.1 System Summary

The **Terminal Services** System provides functionality for securely connecting remote clients and servers, for channeling communication between components of remote clients and servers, and for managing servers.

The Terminal Services System implements the **Remote Desktop Protocol (RDP)** which is a multi-channel protocol that allows users of a remote client to connect to a server over a network. This multi-channel capability enables the use of separate channels, called **virtual channels**, to carry different types of data including presentation data, highly-encrypted data (such as keyboard and mouse user input), device communication, and licensing information.

RDP is used to initialize connections, negotiate capabilities (including security), and transfer input and graphics between a remote client (**RDP Client**) and a server (Terminal Server or TS Server). An RDP Client is an application running on a personal computer, a PDA, or some other device.

When a user of an RDP Client runs a **remote application**, the application is executed on the TS Server, and the TS Server sends graphical output or other types of data to the RDP Client. User input from the RDP Client is transmitted to the TS Server. Additionally, the Terminal Services System protocols define an extensible method for communicating device and system data between an RDP Client and a TS Server.

The Terminal Services System enables an RDP Client and a TS Server to communicate directly, or to communicate across a **firewall** using a **gateway** protocol that tunnels RDP communications.

A Terminal Services System may also be deployed in other enterprise network topologies, such as virtual private networks, to allow access to user sessions on individual TS Servers or TS Servers configured in farms.

2.2 List of Member Protocols

The member protocols of the Terminal Services System include the following:

Terminal Services: Basic Connectivity and Graphics Remoting, as specified in [MS-RDPBCGR]. This protocol is Microsoft's implementation of the Remote Desktop Protocol (RDP) which in turn is based on the ITU T.share protocol (T.128 protocol). It facilitates user interaction with a remote computer system by transferring graphics display information from the remote computer to the user and transferring input from the user to the remote computer, where the input is injected into the user session. This protocol also provides an extensible mechanism allowing specialized communication between components on the user computer and components running on the remote computer. This protocol will be referred to throughout this document as the RDP protocol.

Terminal Services: Gateway Server Protocol, as specified in [MS-TSGU]. This protocol provides the ability to tunnel RDP communications through a gateway for a connection between an RDP Client and a TS Server behind a firewall. This protocol is based on the Remote Procedure Call over HTTP Protocol, as specified in [MS-RPCH].

Terminal Services: Terminal Server Runtime Interface, as specified in [MS-TSTS]. This is a protocol used for remotely querying and configuring various aspects of a TS Server. For example, this protocol can be used to query the number of active sessions running on a TS Server.

Terminal Services: Workspace Provisioning Protocol, as specified in [MS-TSWP]. This protocol allows a unified view of user work resources for administrators that have no access to non-managed computers. The protocol is used to transfer information so that the client computer can launch a remote **desktop** and remote applications on a server or virtual computer.

The following protocols are used by the Terminal Services System to optimize graphical data, support session management and licensing, and create dynamic virtual channels.

Terminal Services: Desktop Composition Extension, as specified in [MS-RDPEDC]. This protocol supports remote desktop composition (the composition of a **sprite tree** that represents the desktop with nodes representing the **sprites**).

Terminal Services: Graphics Device Interface (GDI) Acceleration Extension, as specified in [MS-RDPEGDI]. This protocol encodes the drawing operations that produce an image, reducing the bandwidth associated with graphics **remoting**.

Terminal Services: Composited Remoting V2 Specification, as specified in [MS-RDPCR2]. This protocol is used to display the contents of a desktop running on one machine (the server) on a second, remote, machine (the client) connected to the first via a network.

Terminal Services: NSCodec Extension, as specified in [MS-RDPNSC]. This protocol specifies an image codec that can be used to encode screen images by utilizing efficient and effective compression.

Terminal Services: RemoteFX Codec Extension, as specified in [MS-RDPRFX]. This protocol specifies a lossy image codec that can be used to encode screen images by utilizing efficient and effective compression.

Terminal Services: Session Selection Extension, as specified in [MS-RDPEPS]. This protocol describes the messages exchanged between an RDP Client and a Server to facilitate the precise targeting of an application sharing context.

Terminal Services: Licensing Extension, as specified in [MS-RDPELE]. This protocol allows authorized RDP Clients or users to connect to a TS Server. This extension involves communication between the RDP Client, the **terminal server** and a license server. The TS Server can be configured to be in per device or per user license mode. Client Access Licenses (CALs) are installed on a license server, and when a TS Server requests a license on a client's behalf, the license server issues a license out of its available pool of licenses.

The following list of protocols is used by the Terminal Services System to communicate device, system, and application information.

Terminal Services: Clipboard Virtual Channel Extension, as specified in [MS-RDPECLIP]. This protocol provides basic programmatic access to the clipboard provided by an operating system and ensures that any application has the capability to place data onto the clipboard, extract data from the clipboard, enumerate the data formats available on the clipboard, and register to receive notifications when the system clipboard is updated.

Terminal Services: Dynamic Channel Virtual Channel Extension, as specified in [MS-RDPEDYC]. This protocol implements a generic connection-oriented communication channel on top of the virtual channel protocol. A **dynamic virtual channel (DVC)** is established over an existing **static virtual channel**.

Terminal Services: File System Virtual Channel Extension, as specified in [MS-RDPEFS]. This protocol provides access between the TS Server and the RDP Client file system drivers by redirecting all input/output requests and responses between the two.

Terminal Services: Serial Port Virtual Channel Extension, as specified in [MS-RDPESP]. This protocol specifies the communication used to enable the redirection of ports between a terminal client and a TS Server. By redirecting ports from the RDP Client to the TS Server, applications running on a TS Server can access the remote devices attached to those ports.

Terminal Services: Print Virtual Channel Extension, as specified in [MS-RDPEPC]. This protocol specifies the communication used to enable the **redirection of printers** between a RDP Client and a TS Server. By redirecting printers from the RDP Client to the TS Server, applications running on a server can access the remote devices as if they were local printers.

Terminal Services: Smart Card Virtual Channel Extension, as specified in [MS-RDPESC]. This protocol enables client smart card devices to be available, within the context of a single RDP session, to server-side applications.

Terminal Services: Audio Output Virtual Channel Extension, as specified in [MS-RDPEA]. This protocol transfers audio data from the TS Server to the RDP Client. For example, when the TS Server plays an audio file, this protocol is used by the TS Server to transfer the audio data to the RDP Client. The RDP Client may then play the audio.

Terminal Services: Audio Input Virtual Channel Extension, as specified in [MS-RDPEAI]. This protocol enables the transfer of audio data from the RDP Client to the TS Server. For example, an application running on a TS Server may record audio data. This data will be transferred from the RDP Client to the TS Server, allowing the application to record from an audio device installed on the RDP Client.

Terminal Services: Multiparty Virtual Channel Extension, as specified in [MS-RDPEMC]. This protocol enables the remote display of desktop and application content. To effectively implement an application-sharing or collaborative solution, additional information must be conveyed to keep the participants apprised of whom else is involved, in addition to which applications or windows are being shared.

Terminal Services: Plug and Play Devices Virtual Channel Extension, as specified in [MS-RDPEPNP]. This protocol specifies the communication used to enable the **redirection of Plug and Play** devices between an RDP Client and a TS Server.

Terminal Services: USB Devices Virtual Channel Extension, as specified in [MS-RDPEUSB]. This protocol is used to redirect **universal serial bus (USB)** devices from a terminal client to the terminal server, which allows the server access to a device that is physically connected to the client as if the device were local to the server.

Terminal Services: Remote Programs Virtual Channel Extension, as specified in [MS-RDPERP]. This protocol is a Remote Desktop Protocol (RDP) feature (as specified in [MS-RDPBCGR]) that presents a remote application (running remotely on a **remote application integrated locally (RAIL)** server) as a local user application (running on the RAIL client computer). Also known as RAIL.

Terminal Services: Video Virtual Channel Extension, as specified in [MS-RDPEV]. This protocol enables the transfer of synchronized audio and video data from a TS Server to an RDP Client. The RDP Client can play the audio and video data and synchronize this data using the timing information provided by the protocol.

Terminal Services: XML Paper Specification (XPS) Print Virtual Channel Extension, as specified in [MS-RDPEXPS]. This protocol specifies communication between a virtual printer driver installed on a TS Server and a printer driver installed on the RDP Client. The primary purpose of this protocol is to acquire printing capabilities and to display a printer-specific user interface on the RDP Client.

2.3 Relevant Standards

The system uses the standards listed below to allow interoperability with other external systems.

DNS, as specified in [RFC1035]. This standard is used in name resolution (for locating the TS Server).

TCP/IP, as specified in [RFC793]. This standard is used as a reliable transport layer protocol for Remote Desktop Protocol.

SSL, as specified in [SSL3]. This standard is used for encrypting data that is sent through an RDP channel.

TLS, as specified in [RFC2246]. Remote Desktop Protocol uses TLS in security negotiation.

3 Foundation

This section describes the theoretical and practical information needed to understand this document and this system.

3.1 Background Knowledge and System-Specific Concepts

This section summarizes:

- Background knowledge required to understand this document.
- Concepts specific to this system.

Background knowledge of the following concepts is helpful in understanding the Terminal Services System Overview document:

- Data encryption using Secure Socket Layers (SSL) and Transport Layer Security (TLS)
- User authentication using **Network Level Authentication (NLA)**
- **Server Authentication**
- Licensing using Client Access Licenses (CALs)

Knowledge of the following system-specific concepts is helpful in understanding the Terminal Services System Overview document:

- Remote presentation of desktops and applications
- The Remote Desktop Protocol (RDP)
- Negotiating capabilities between clients and servers
- Virtual channels
- Redirection functionality
- Component protocol documentation

3.1.1 Remote Presentation of Desktops and Applications

In the Terminal Services System, a client computer or system can use applications and resources that are not installed on the client by connecting to a user session on a server where the software is running. The user interacts with the server using a desktop, similar to the desktop available on the client, but generated remotely as a part of the user session on the server and then transported to the client computer using a protocol known as the Remote Desktop Protocol (RDP). This process is known as remote presentation. Applications and resources are remotely presented to the user. This activity is also referred to as remoting, such as application remoting.

To support user interaction with remote applications and resources, RDP transports input from the user (such as from the keyboard or mouse) to the server. RDP can also be used to transport data from devices attached to the RDP Client, such as **smart cards** or microphones. Conversely, RDP is used to transport data from remote applications running on a server to devices attached to the RDP Client, such as sending audio data to the audio sub-system on the RDP Client, or sending print jobs to the print-spooler on the RDP Client.

The Terminal Services System is implemented using the Remote Desktop Protocol. A client that supports the Terminal Services System is referred to as an RDP Client, because the client has a software component installed that supports RDP. The type of server that an RDP Client communicates with is referred to as a terminal server, or more commonly, a TS Server.

3.1.2 Remote Desktop Protocol (RDP) Connections

The implementation of the Remote Desktop Protocol (RDP) in the Terminal Services System is described in [MS-RDPBCGR]. To establish a secure connection, an RDP Client negotiates capabilities, requirements, and security with a TS Server. RDP defines the **handshake** used to establish the connection between the RDP Client and TS Server, as well as input primitives for transporting user input to the TS Server and drawing primitives for transporting graphical information to the RDP Client.

RDP connections may be made directly between an RDP Client and a TS Server when the RDP Client is in the same intranet as the TS Server, or may pass through a separate server acting as a TS Gateway. A TS Gateway is typically used when an RDP Client connects over the internet to a TS Server.

3.1.3 Negotiating Capabilities

The goal of the initialization sequence is to establish the client and server capabilities, exchange settings, and synchronize the initial state of the client and server. Capability negotiation for Remote Desktop Protocol (RDP) is essentially the same as for T.128. The server advertises its capabilities in a **Demand Active PDU** sent to the client, and the client advertises its capabilities in the follow-up **Confirm Active PDU** (see the Capability Exchange phase in [MS-RDPBCGR] (section 1.3.1.1)). Capability sets are packaged in a combined capability set structure. This structure contains a count of the number of capability sets, followed by the contents of the individual capability sets.

3.1.4 User Authentication, Authorization, and Licensing

The Terminal Services System uses authentication, authorization, and licensing services. Not all of these services are defined as a part of the Terminal Services System or use component protocols of the Terminal Services System. The Terminal Services System depends on the availability of the following services:

- **Secure Sockets Layer (SSL) and Transport Layer Security (TLS):** Remote Desktop Protocol (RDP) supports SSL and TLS protocols for encrypting data that is sent through an RDP channel.

- **Network Level Authentication (NLA):** RDP supports Network Level Authentication, an authentication method that finishes the user authentication process before a full RDP connection is established and the logon screen appears.

- **Server Authentication:** RDP supports Server Authentication. This is used to verify that a client is connecting to an authorized TS Servers.

- **Terminal Server Licensing:** Terminal Services Licensing services are only associated with licensing an RDP Client to use a TS Servers using Client Access Licenses (CALs). CAL tokens are electronic representations of real licenses. Terminal Services Licensing services are designed to manage these license tokens.

3.1.5 Static and Dynamic Virtual Channels

When a Remote Desktop Protocol (RDP) connection is established, communication channels are set up for exchanging data between the RDP Client and the TS Server. These channels are known as static virtual channel. Component protocols of the Terminal Services System define the protocols used over these virtual channels.

In the following figure, a client and TS Gateway have already established a **tunnel** for RDP communication between the client and terminal server. The client is sending input to the terminal server, the terminal server is sending graphics to the client, and they are exchanging operational messages. In addition, RDP is acting as a tunnel for the exchange of virtual channel communication with the client clipboard, file system, print-spooler, and audio sub-system.

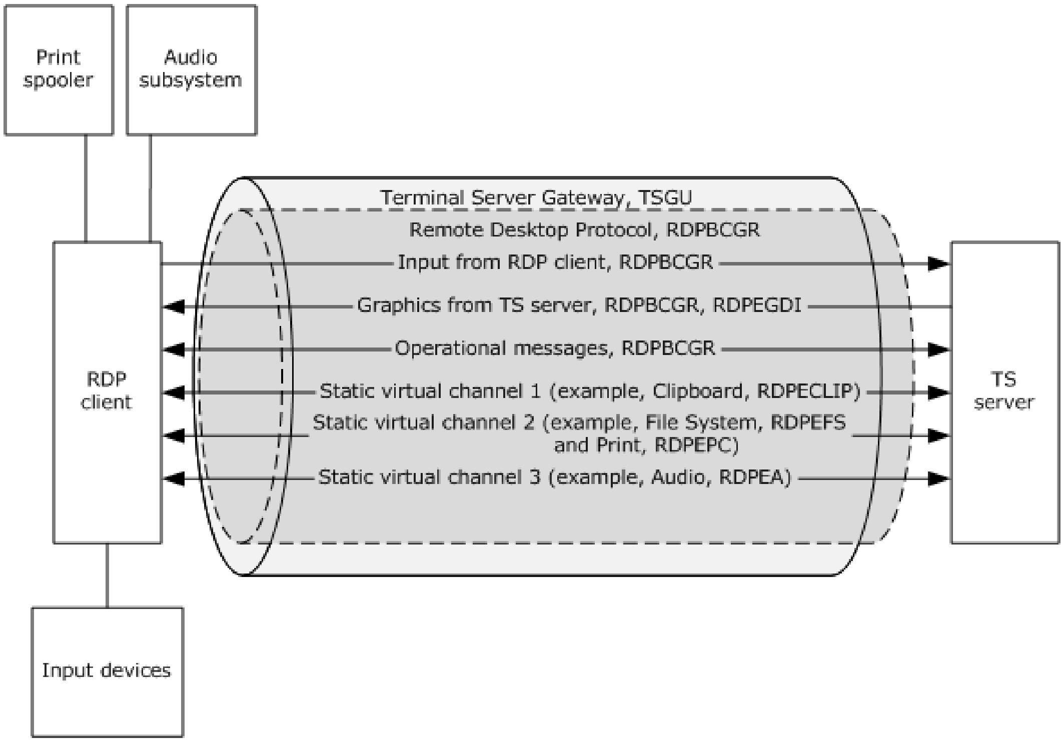

Print spooler
Audio subsystem
RDP client
Input devices
Terminal Server Gateway, TSGU
Remote Desktop Protocol, RDPBCGR
Input from RDP client, RDPBCGR
Graphics from TS server, RDPBCGR, RDPEGDI
Operational messages, RDPBCGR
Static virtual channel 1 (example, Clipboard, RDPECLIP)
Static virtual channel 2 (example, File System, RDPEFS and Print, RDPEPC)
Static virtual channel 3 (example, Audio, RDPEA)
TS server

Figure 1: TS Gateway tunneling RDP connection with virtual channels

To enable a virtual channel to be opened at anytime during a session, a static virtual channel **pipe** may be established when the session begins. Within this pipe, multiple new channels (dynamic virtual channels (DVC)) can be created and multiplexed anytime during the session.

In the following figure, an RDP Client and TS Gateway have already established a tunnel for RDP communication between the RDP Client and TS Server. The RDP Client is sending input to the TS Server, the TS Server is sending graphics to the RDP Client, and they are exchanging clipboard data using a virtual channel. Within the RDP pipe, a static virtual channel has also been established for tunneling DVCs. DVCs are used for communication between endpoints on the RDP Client and TS Server that may not be active when the initial connection is established, such as a Plug and Play device that has not yet been plugged in. DVCs can be created at anytime during a session in which a static virtual channel for DVCs has been created.

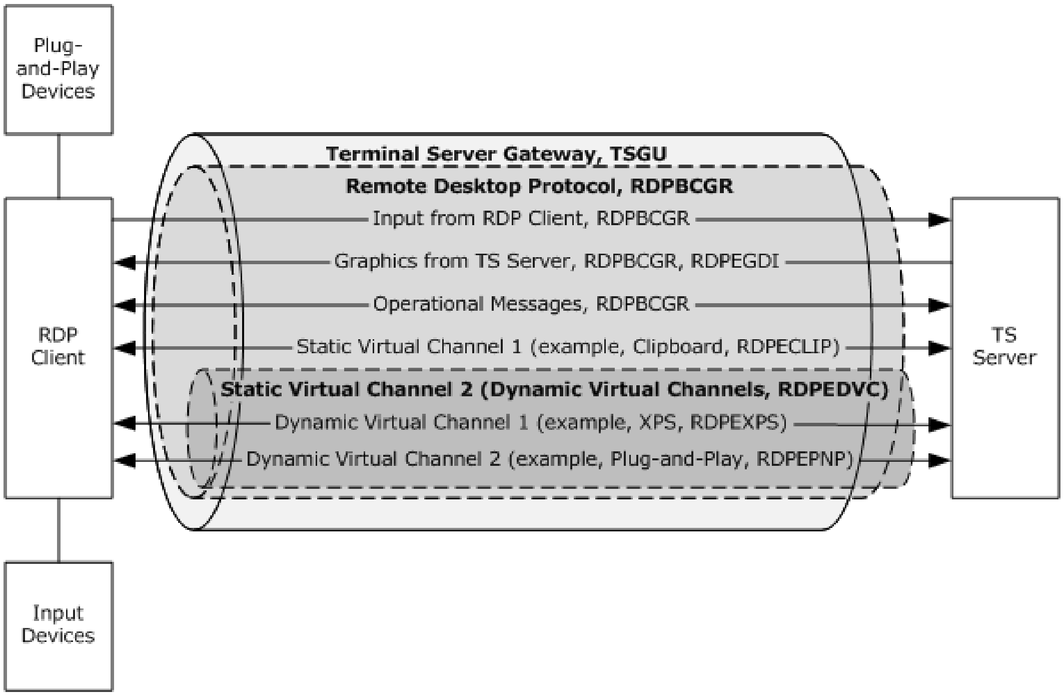

Plug-and-Play Devices
RDP Client
Input Devices
Terminal Server Gateway, TSGU
Remote Desktop Protocol, RDPBCGR
Input from RDP Client, RDPBCGR
Graphics from TS Server, RDPBCGR, RDPEGDI
Operational Messages, RDPBCGR
Static Virtual Channel 1 (example, Clipboard, RDPECLIP)
Static Virtual Channel 2 (Dynamic Virtual Channels, RDPEDVC)
Dynamic Virtual Channel 1 (example, XPS, RDPEXPS)
Dynamic Virtual Channel 2 (example, Plug-and-Play, RDPEPNP)
TS Server

Figure 2: TS Gateway tunneling RDP static virtual channel tunneling dynamic virtual channels

3.1.6 Redirection Functionality

When an Remote Desktop Protocol (RDP) connection exists between an RDP Client and TS Server, data and resources are frequently redirected. This redirection allows the TS Server to access resources on the RDP Client, as well as allowing a TS Server to redirect data from remote applications on the TS Server to the RDP Client. Some examples of redirection functionality include the following:

- **Keyboard and mouse input:** Data from the keyboard and mouse on the RDP Client is redirected to the user session on the TS Server.
- **Printer jobs:** Print jobs from the user session on the TS Server can be redirected to a printer attached to the RDP Client.
- **Media Player content:** An application running on the TS Server can redirect Media Player content to the RDP Client.
- **File System data:** A TS Server can access local drives on the RDP Client using **file system redirection**.
- **Clipboard:** **Clipboard redirection** enables a user to copy data from an application running on a TS Server to a clipboard located on the RDP Client.
- **Smart Card:** A TS Server can access credential data from a smart card connected to an RDP Client.
- **Ports:** A TS Server can access devices connected to serial and parallel ports on an RDP Client.

The redirection functionality in Microsoft Windows® client is configurable through Windows client user interface options to choose resource redirection at the clipboard, printer, or other Plug and Play device level. Based on the version of Windows client, this configuration can be overridden by the Terminal Services Gateway ([MS-TSGU]) protocol (Appendix A).

3.2 System Purposes

The Terminal Services System provides protocols and system components to implement a presentation remoting system while controlling the interactive input and output for the desktop or application from another location, in a secure, manageable and distributed network environment such as an Internet or intranet environment.

The Terminal Services System allows end users to access remote applications not available on their own computers. From an enterprise perspective, the Terminal Services System allows applications and data to be installed in a centralized location for access by multiple users, reducing the overhead burden of managing many locally installed applications.

3.3 System Use Cases

The Terminal Services System is designed to support scenarios that allow users to access applications and data on a remote computer over the network. When a User wants to interact with a remote computer, the system facilitates this interaction by transferring graphics display information from the remote computer to the User and transporting input (such as keyboard or mouse input) from the User to the remote computer. The Terminal Services System also supports an extensible transport mechanism for specialized communications between components on the user’s computer and components running on the remote computer.

This section provides the use cases that describe this functionality in terms of actors that participate in this system and their goals. The use case participants include an RDP Client, a terminal server (TS Server), and terminal server Gateway (TS Gateway). Participants may also include applications or devices running on the RDP Client, and applications or devices running on the TS Server.

The use cases described in this section are in three groups:

- Establishing a connection between an RDP Client and a TS Server.
- Redirecting data from an RDP Client to a Remote Desktop on a TS Server, or from a Remote Desktop on a TS Server, to an RDP Client. Because redirection is implemented similarly regardless of the type of data, and the type of data is extensible, this group contains a sampling of possible redirection use cases.
- Terminating a connection between an RDP Client and a TS Server.

Section 3.3.3 provides a table that maps the use cases described in section 3.3.4 to these use case groups.

3.3.1 Stakeholders and Interests Summary

RDP Client: The RDP Client connects to a TS Server, routes data from the TS Server to RDP Client endpoints, and disconnects from the TS Server.

TS Server: The TS Server processes RDP Client requests (connect, disconnect, and input), creates user sessions for the RDP Client, and sends the display output of the user session to the RDP Client.

Devices: Devices attached to the RDP Client are redirected to the TS Server using the RDP protocol through virtual channels established during the connection sequence. Devices include printers, smart cards, and audio components.

VM Host: A VM **Host** is an alternative implementation on Windows 7 of a server that connects to an RDP Client. This implementation is called a Virtual Machine Host, and creates Virtual Machines that host remote desktops (VM Guest Desktops), rather than the user sessions hosted on a TS Server. The same protocols of the Terminal Services System are used, regardless of whether a VM host or a TS Server is used.

TS Gateway: The TS Gateway is a server that tunnels an RDP connection through a firewall, enabling an RDP Client to connect to a TS Server or VM Host passing through network zones by tunneling the RDP connection.

TS Administrator: The individual responsible for configuring, deploying, monitoring, and troubleshooting the TS System. The TS Administrator is interested in the details of the administration and configuration of the interfaces exposed by the system. They need to know which are the member protocols that make up this system and communication details they expect to see flowing across the wire such as message format, error codes, and retry logic.

Architect: The individual who designs systems or applications that support remote access.

Developer: The individual who designs and implements an RDP Client, a terminal server, or an administrative tool for managing the TS Server. The developers of a Terminal Services System or a subcomponent of the system will be responsible for implementing the incoming and outgoing interfaces of the system as documented in this document and the respective technical documents of the member protocols of the Terminal Services System.

Tester and QA Personnel: The individuals who determine if the TS System meets the design and functionality requirements.

3.3.2 Supporting Actors and System Interests Summary

The Terminal Services System has the following supporting actors with the noted interests:

Domain Name System (DNS): Provides name resolution services.

Authentication System: The Authentication System as specified in [MS-AUTHSO] provides authentication services to secure communications in the Terminal Services System and the authentication services that support the client to server communication within and outside the system.

Licensing Services: Licensing Services obtain and issue licenses to users or devices using TS Servers.

3.3.3 Use Cases Diagrams

The following table provides an overview for the groups of use cases which span the functionality of the Terminal Services System. Detailed descriptions for these use cases are provided in section 3.3.4.

Use case group	Use cases
Establishing a connection between an RDP Client and a TS Server and presenting a remote desktop or remote application	Establish a connection to a TS Server in an Intranet Environment – RDP Client Establish a connection to a VM Host in an

Use case group	Use cases
	Intranet Environment – RDP Client Establish a connection using a TS Gateway – RDP Client Establish a connection to a TS Server in a TS **server farm** – RDP Client
Redirecting data from an RDP Client to a remote desktop or from a remote desktop to an RDP Client	Access local drives on an RDP Client – remote application Redirect clipboard data from a remote application – RDP Client Use a printer on an RDP Client – remote application Redirect smart card data from an RDP Client – remote application Access a plug and play device on an RDP Client – remote application Present content from a TS Server on an RDP Client – Media Player Access audio device on an RDP Client – remote application
Terminating a connection between an RDP Client and a TS Server	Log Off from a remote session – RDP Client Disconnect from a remote session– RDP Client

The following use case diagrams illustrate the separate groups of use cases described in this section.

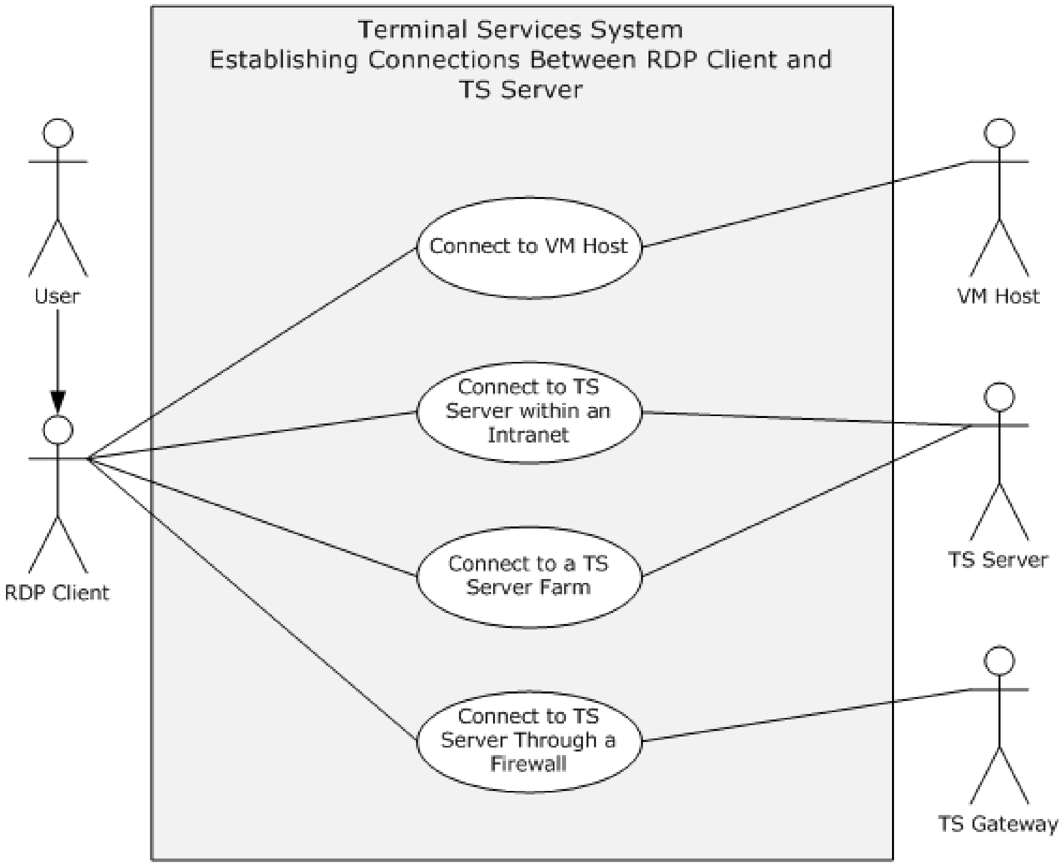

Terminal Services System
Establishing Connections Between RDP Client and TS Server
User
RDP Client
Connect to VM Host
Connect to TS Server within an Intranet
Connect to a TS Server Farm
Connect to TS Server Through a Firewall
VM Host
TS Server
TS Gateway

Figure 3: Use case group: Establishing a connection between an RDP Client and a TS Server

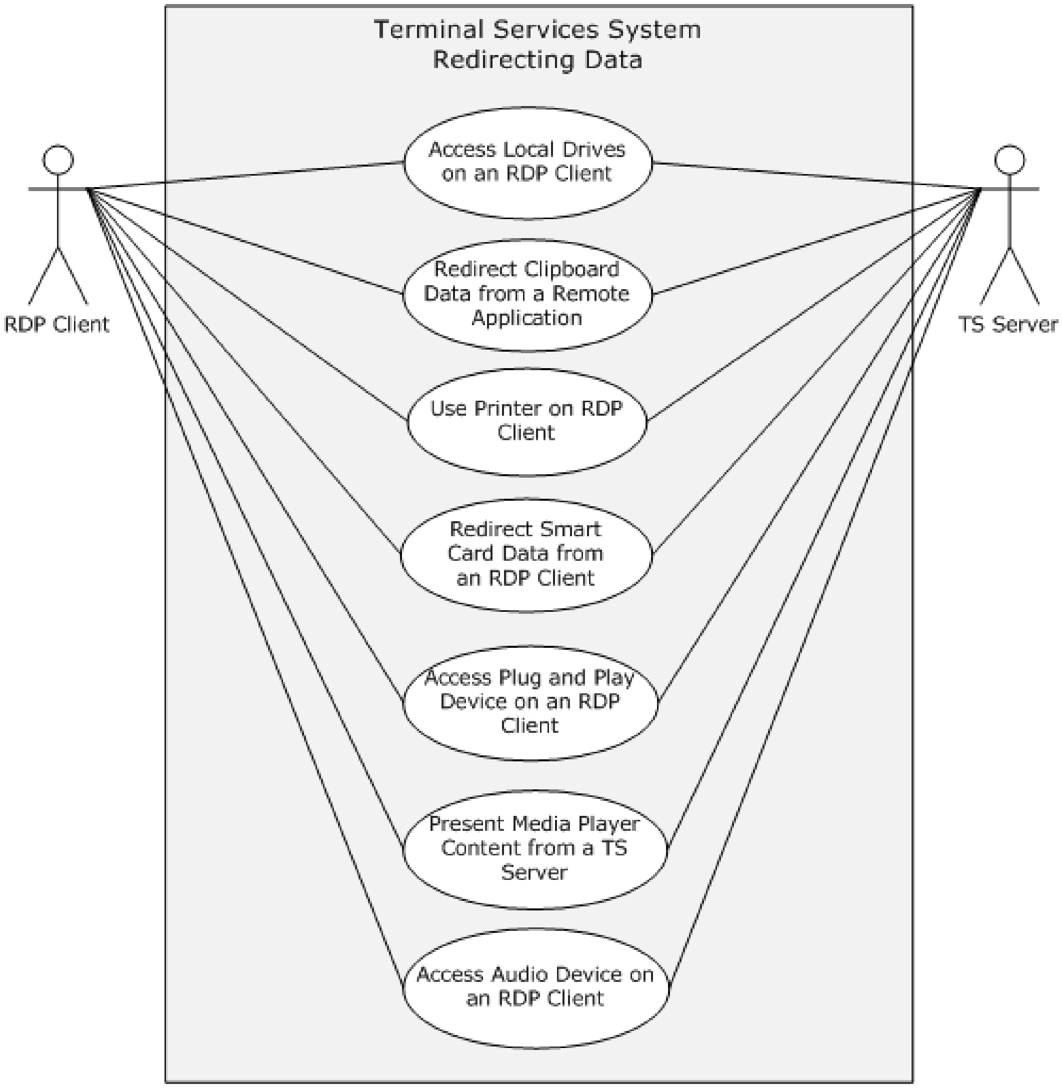

Terminal Services System
Redirecting Data
RDP Client
TS Server
Access Local Drives on an RDP Client
Redirect Clipboard Data from a Remote Application
Use Printer on RDP Client
Redirect Smart Card Data from an RDP Client
Access Plug and Play Device on an RDP Client
Present Media Player Content from a TS Server
Access Audio Device on an RDP Client

Figure 4: Use case group: Redirecting data from an RDP Client to a Remote Application

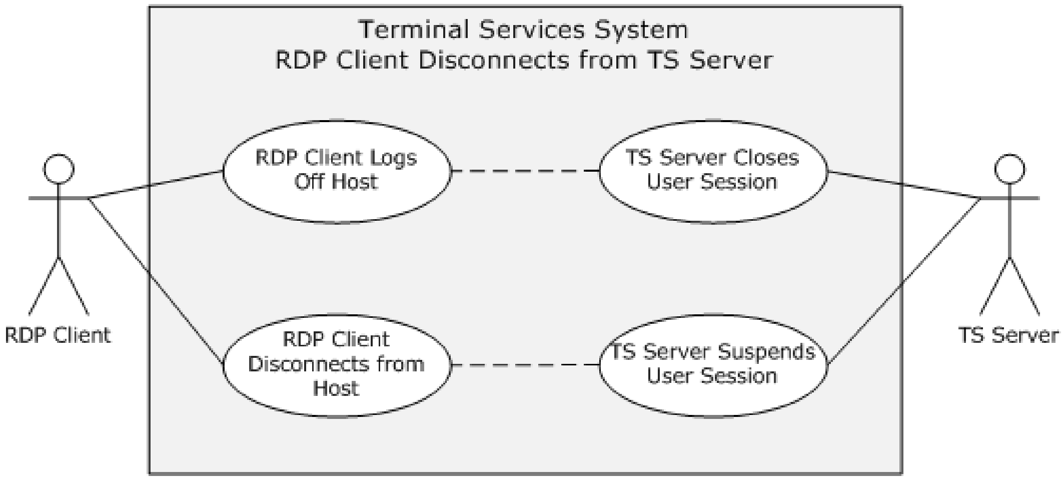

Terminal Services System
RDP Client Disconnects from TS Server
RDP Client Logs Off Host
TS Server Closes User Session
RDP Client Disconnects from Host
TS Server Suspends User Session
RDP Client
TS Server

Figure 5: Use case group: Terminating a connection between an RDP Client and a TS Server

3.3.4 Use Case Descriptions

3.3.4.1 Establish a Connection to a TS Server in an Intranet Environment — RDP Client

Goal: For an RDP Client to establish a connection with a TS Server.

Context of Use: A User launches the RDP Client to display and interact with a remote desktop or remote application. The RDP Client establishes a connection to a TS Server that is hosting the remote desktop or remote application. In this use case, the connection between the RDP Client and the TS Server is established in an intranet environment. The preconditions are met, and licensing, authentication, authorization, and **Domain Name System (DNS)** services are available.

Direct Actor: The direct actor is the RDP Client.

Primary Actor: The primary actor is the User.

Supporting Actors: The supporting actors are the TS Server, Licensing Services, Authentication System, and DNS.

Stakeholders and Interests:

- RDP Client: The RDP Client establishes a connection to the TS Server in order to interact with a remote desktop or remote application.
- TS Server: The TS Server is hosting the remote desktop or remote application to which the direct actor is connecting.

Preconditions: The TS Server is operational and listening for an RDP connect request on port 3389. If the direct actor is using the IPv6 protocol, then all supporting actors MUST support the IPv6 protocol.

Minimal Guarantees: The connection is prevented from being established.

Success Guarantee: The RDP Client will have a connection to the TS Server, allowing the RDP Client to display and interact with a remote desktop or remote application.

Trigger: The RDP Client initiates the connection when the User provides the name of the remote desktop to connect to.

Main Success Scenario:

1. The User provides the name of the remote desktop to the RDP Client.
2. The RDP Client resolves the name to an IP address using DNS.
3. The RDP Client establishes a secure RDP connection to port 3389 on the TS Server.
4. The RDP Client successfully negotiates use permissions with the TS Server using the client license.
5. The TS Server checks the User credentials and then creates a user session for the RDP Client.
6. The TS Server transmits the desktop of the user session to the RDP Client using the RDP protocol (presentation remoting) and the RDP Client displays the remote desktop.
7. The User of the RDP Client interacts with the remote desktop (RDP is used to transfer keyboard and mouse input).

Extensions: In Windows7 implementations, an RDP Client may connect to a virtual machine on a VM Host, rather than a user session on a TS Server.

3.3.4.2 Establish a Connection to a VM Host in an Intranet Environment — RDP Client

Goal: For an RDP Client to establish a connection with a VM Host.

Context of Use: A User launches the RDP Client to display and interact with a remote desktop or remote application. The RDP Client establishes a connection to a VM Host that is hosting the virtual machine running the remote desktop or remote application. In this use case, the connection between the RDP Client and the VM Host is established in an intranet environment. The preconditions are met, and licensing, authentication, authorization, and Domain Name System (DNS) services are available.

Direct Actor: The direct actor is the RDP Client.

Primary Actor: The primary actor is the same as the direct actor.

Supporting Actors: The supporting actors are the VM Host, Licensing Services, Authentication System, and DNS.

Stakeholders and Interests:

- RDP Client: The RDP Client establishes a connection to the VM Host in order to interact with a remote desktop or remote application.
- VM Host: The VM Host is hosting the virtual machine running the remote desktop or remote application to which the direct actor is connecting.

Preconditions: Windows 7 is being used. The VM Host is operational and listening for an RDP connect request on port 3389. If the direct actor is using the IPv6 protocol, then all supporting actors MUST support the IPv6 protocol.

Minimal Guarantees: The connection is prevented from being established.

Success Guarantee: The RDP Client will have a connection to the VM Host, allowing the RDP Client to display and interact with a remote desktop or remote application.

Trigger: The RDP Client initiates the connection when the User provides the name of the remote desktop to connect to.

Main Success Scenario:

1. The User provides the name of the remote desktop to the RDP Client.
2. The RDP Client resolves the name to an IP address using DNS.
3. The RDP Client establishes a secure RDP connection using a **session broker**. The session broker looks up the assigned virtual machine for the User, prepares the virtual machine to be run under the VM Host, and returns the IP address of the virtual machine to the RDP Client.
4. The RDP Client connects to port 3389 and the IP address of the virtual machine.
5. The virtual machine transmits the remote desktop to the RDP Client and the RDP Client displays the remote desktop.
6. The User of the RDP Client interacts with the remote desktop of the virtual machine.

Extensions: None.

3.3.4.3 Establish a Connection Using a TS Gateway – RDP Client

Goal: For an RDP Client to establish a connection with a TS Server.

Context of Use: The RDP Client is using the Internet to transport communication. The RDP Client is external to a firewall separating the desired TS Server from the Internet. The RDP Client uses a gateway to tunnel communication to the TS Server. The preconditions are met, and licensing, authentication, authorization, and Domain Name System (DNS) services are available.

Direct Actor: The direct actor is the RDP Client.

Primary Actor: The primary actor is the user.

Supporting Actors: The supporting actors are the TS Server, TS Gateway, Licensing Services, Authentication System, and DNS.

Stakeholders and Interests:

- RDP Client: The RDP Client establishes a connection to the TS Server in order to interact with a remote desktop or remote application.
- TS Gateway: The gateway tunnels communication from the RDP Client to a TS Server located behind a firewall.
- TS Server: The TS Server is hosting the remote desktop or remote application to which the direct actor is connecting.

Preconditions: The TS Gateway is operational and listening for a **remote procedure call (RPC)** connection request on a known port. The TS Gateway is capable of making RDP connections to the requested TS Server. The TS Server is operational and listening for an RDP connect request on port 3389. If the direct actor is using the IPv6 protocol, then all supporting actors MUST support the IPv6 protocol.

Minimal Guarantees: The connection is prevented from being established.

Success Guarantee: The RDP Client will have a connection to the TS Server using a TS Gateway, allowing the RDP Client to display and interact with a remote desktop or remote application.

Trigger: The RDP Client initiates the connection when a User provides the name of the remote desktop to connect to.

Main Success Scenario:

1. The User provides the name of the remote desktop to the RDP Client.
2. The RDP Client establishes a secure RDP connection to the TS Gateway through the RPC endpoint.
3. TS Gateway resolves the name to an IP address using DNS.
4. TS Gateway establishes an RDP connection to port 3389 on the TS Server.
5. The RDP Client successfully negotiates use permissions with the TS Server, using TS Gateway, using the client license.
6. The TS Server validates the client-provided certificates and client license and then creates a user session for the RDP Client.
7. The TS Server transmits the desktop of the user session to the RDP Client and the RDP Client displays the remote desktop (presentation remoting).
8. The User of the RDP Client interacts with the remote desktop (with keyboard and mouse through the RDP protocol).

Extensions: None.

3.3.4.4 Establish a Connection to a TS Server in a TS Server Farm – RDP Client

Goal: For an RDP Client to establish a connection to a TS Server within a server farm.

Context of Use: A user launches the RDP Client to display and interact with a remote desktop or remote application. An RDP Client initiates a connection to a TS Server that is hosting the remote desktop or remote application, and the connection may be redirected to another TS Server within the same server farm for the purposes of load balancing. In this use case, the connection between the RDP Client and the TS Server is established in an intranet environment. The preconditions are met, and session brokering, licensing, authentication, authorization, and Domain Name System (DNS) services are available.

Direct Actor: The direct actor is the RDP Client.

Primary Actor: The primary actor is the same as the direct actor.

Supporting Actors: The supporting actors are the TS Server, Session Broker, Licensing Services, Authentication System, and DNS services.

Stakeholders and Interests:

- RDP Client: The RDP Client establishes a connection to the TS Server in order to interact with a remote desktop or remote application.

- TS Server: The TS Server is hosting the remote desktop or remote application to which the direct actor is connecting.
- Session Broker: The Session Broker assigns the RDP Client to a TS Server within a server farm according to an algorithm to optimize load balancing.

Preconditions: The TS Server is operational and listening for an RDP connect request on port 3389. A session broker is available to redirect the RDP Client to an alternate TS Server. If the direct actor is using the IPv6 protocol, then all supporting actors must support the IPv6 protocol.

Minimal Guarantees: The connection is prevented from being established.

Success Guarantee: The RDP Client will have a connection to the TS Server. The RDP Client will display the remote desktop or a remote application. The RDP Client will be able to transmit input data to the remote desktop or remote application.

Trigger: The RDP Client initiates the connection when the User provides the name of the remote desktop to connect to.

Main Success Scenario:

1. The User provides the name of the remote desktop to the RDP Client.
2. The RDP Client resolves the server name to an IP address using DNS services.
3. The RDP Client establishes a secure RDP connection to port 3389 on the TS Server.
4. The target TS Server may redirect the RDP Client connection attempt using a session broker if the load on the server does not permit this new connection.
5. If redirected, the RDP Client opens a port on a different server in a farm and initializes an RDP connection to the alternate server.
6. The RDP Client successfully negotiates the client license with the TS Server.
7. The TS Server creates a user session for the RDP Client after credentials are checked.
8. The TS Server sends the session desktop to the RDP Client and the RDP Client displays the remote desktop.
9. The User of the RDP Client interacts with the remote desktop (keyboard and mouse through RDP).

Extensions: In Windows 7 implementations, an RDP Client may connect to a virtual machine on a VM Host, rather than a user session on a TS Server.

3.3.4.5 Access Local Drives on an RDP Client – Remote Application

Goal: For the remote application to access local drives on the RDP Client.

Context of Use: After an RDP Client establishes a connection to a TS Server, a remote application running on the TS Server can access local drives on the RDP Client.

Direct Actor: The direct actor is the remote application.

Primary Actor: The primary actor is the same as the direct actor.

Supporting Actors: The supporting actors are the TS Server, RDP Client, and local drives on the RDP Client.

Stakeholders and Interests:

- TS Server: The TS Server manages redirecting file system data from the local drives to the remote application.
- RDP Client: The RDP Client redirects file system data from local drives.
- Local Drives on the RDP Client: The local drives are accessible to the remote application or remote desktop after the RDP Connection is established.

Preconditions: The RDP Client is connected to the TS Server. The RDP Connection supports file system redirection. The remote desktop or remote application is running on the TS Server. The TS Server has permission to access the local drives on the RDP Client.

Minimal Guarantees: The remote application will be restricted from accessing the local drives.

Success Guarantee: Local drives will be available to the remote application.

Trigger: The remote application requests file system data from the local drives on the RDP Client.

Main Success Scenario: The remote application will have the ability to read from and write to the local drives on the RDP Client.

Extensions: None.

3.3.4.6 Redirect Clipboard Data from a Remote Application – RDP Client

Goal: To use the local clipboard of the RDP Client to perform clipboard operations on a remote application running on a TS Server.

Context of Use: The User of a remote application can copy data from a remote application and paste data to a remote application using the local clipboard when a static virtual channel supporting clipboard redirection is established during the initial RDP connection.

Direct Actor: The direct actor is the RDP Client.

Primary Actor: The primary actor is the same as the direct actor.

Supporting Actors: The supporting actors are the TS Server, remote application, the clipboard on the RDP Client, and the clipboard on the remote desktop.

Stakeholders and Interests:

- TS Server: The TS Server manages redirecting clipboard data from the remote application to the clipboard on the RDP Client.
- RDP Client: The RDP Client can copy data from a remote application or paste data to a remote application.
- Clipboard on the RDP Client: The clipboard on the RDP Client is accessible to the remote application or remote desktop session after the RDP Connection is established.
- Clipboard on the remote desktop: The clipboard on the remote desktop is synchronized with the clipboard on the RDP Client, providing the redirection functionality.

Preconditions: The RDP Client is connected to the TS Server. The RDP connection supports clipboard redirection. The remote desktop or remote application is running on the TS Server. The TS Server has permission to access the clipboard on the RDP Client.

Minimal Guarantees: The TS Server will be prevented from accessing the clipboard on the RDP Client.

Success Guarantee: The RDP Client is able to use the local clipboard to copy data from the remote application to a local directory on the RDP Client.

Trigger: The RDP Client attempts to use clipboard features to copy data from or paste data to a remote application.

Main Success Scenario: RDP Client application will be able to do clipboard operations between client applications.

Extensions: None.

3.3.4.7 Use Printer on RDP Client – Remote Application

Goal: For the remote application to send a print job to a printer on the RDP Client, which prints the job.

Context of Use: After an RDP Client establishes a connection to a TS Server, a remote application running on the TS Server can send a print job to the local printer on the RDP Client.

Direct Actor: The direct actor is the remote application.

Primary Actor: The primary actor is the same as the direct actor.

Supporting Actors: The supporting actors are the TS Server, the RDP Client, and the printer on the RDP Client.

Stakeholders and Interests:

- TS Server: The TS Server manages redirecting the print job from the remote application to the printer on the RDP Client.
- RDP Client: The RDP Client routes the print job to the local printer.
- Printer on the RDP Client: The printer on the RDP Client is accessible to the remote application or remote desktop session after the RDP Connection is established.

Preconditions: The RDP Client is connected to the TS Server. The RDP Connection supports printer redirection. The remote desktop or remote application is running on the TS Server. The TS Server has permission to access the local printer on the RDP Client.

Minimal Guarantees: The remote application will not have access to the local printer.

Success Guarantee: The local printer will be available to the remote application.

Trigger: The remote application sends a print job to the local printer on the RDP Client.

Main Success Scenario: The remote application running on the TS Server on behalf of the client will be able to print to the local printer on the RDP Client.

Extensions: None.

3.3.4.8 Redirect Smart Card Data from an RDP Client – Remote Application

Goal: For the remote application to access a smart card on the RDP Client.

Context of Use: After an RDP Client establishes a connection to a TS Server, a remote application running on the TS Server can access local drives on the RDP Client.

Direct Actor: The direct actor is the remote application.

Primary Actor: The primary actor is the same as the direct actor.

Supporting Actors: The supporting actors are the TS Server, the RDP Client, and the smart card on the RDP Client.

Stakeholders and Interests:

- TS Server: The TS Server manages redirecting smart card data from the RDP Client to the remote application.
- RDP Client: The RDP Client redirects smart card data to the remote application.
- Smart Card: The smart card is accessible to the remote application or remote desktop after the RDP Connection is established.

Preconditions: The RDP Client is connected to the TS Server. The RDP Connection supports smart card redirection. The remote desktop or remote application is running on the TS Server.

Minimal Guarantees: The remote application will be unable to access the smart card data.

Success Guarantee: Smart card data will be available to the remote application.

Trigger: The remote application requests smart card data from the smart card on the RDP Client.

Main Success Scenario: The remote application running on the TS Server will be able to access credential data on the smart card attached to the RDP Client computer during logon.

Extensions: None.

3.3.4.9 Access Plug and Play Device on an RDP Client – Remote Application

Goal: For the remote application to access a Plug and Play device on the RDP Client.

Context of Use: After an RDP Client establishes a connection to a TS Server, a remote application running on the TS Server can access a Plug and Play device on the RDP Client.

Direct Actor: The direct actor is the remote application.

Primary Actor: The primary actor is the same as the direct actor.

Supporting Actors: The supporting actors are the TS Server, the RDP Client, and the Plug and Play device on the RDP Client.

Stakeholders and Interests:

- TS Server: The TS Server manages redirecting data from the Plug and Play device to the remote application.
- RDP Client: The RDP Client redirects Plug and Play data to the remote application.

- Plug and Play device: The Plug and Play device is accessible to the remote application or remote desktop after the RDP Connection is established.

Preconditions: The RDP Client is connected to the TS Server. The RDP Connection supports Plug and Play redirection. The remote desktop or remote application is running on the TS Server. The TS Server has permission to access the Plug and Play device on the RDP Client.

Minimal Guarantees: The remote application will be unable to access the Plug and Play device.

Success Guarantee: The Plug and Play device will be available to the remote application.

Trigger: The remote application requests data from the Plug and Play device on the RDP Client.

Main Success Scenario: The remote application running on the TS Server will be able to access a Plug and Play device installed on the RDP Client computer.

Extensions: None.

3.3.4.10 Present Content from TS Server on an RDP Client – Media Player

Goal: To present content streamed from the Media Player running on the TS Server to the RDP Client.

Context of Use: In Windows 7, Media can be streamed from the Media Player running in the user session on the TS Server to the RDP Client running on the remote system.

Direct Actor: The direct actor is Media Player.

Primary Actor: The primary actor is the same as the direct actor.

Supporting Actors: The supporting actors are the TS Server and the RDP Client.

Stakeholders and Interests:

- TS Server: The TS Server manages streaming content from the Media Player to the RDP Client.
- RDP Client: The RDP Client receives and displays content from the Media Player on the TS Server.
- Media Player: The Media Player plays content on the TS Server that is streamed to the RDP Client for display.

Preconditions: Windows 7 is in use. The RDP Client is connected to the TS Server. The RDP Connection supports Media Player Redirection. Media Player is running on the TS Server.

Minimal Guarantees: The RDP Client will be unable to present remote Media Player content.

Success Guarantee: Media Player can present content on the RDP Client.

Trigger: Media Player begins streaming content to the RDP Client.

Main Success Scenario: The remote application running Media Player on the TS Server will be able to stream Media Player content to the RDP Client.

Extensions: None.

3.3.4.11 Access Audio Device on an RDP Client – Remote Application

Goal: For the remote application to access an audio device on the RDP Client.

Context of Use: After an RDP Client establishes a connection to a TS Server, a remote application running on the TS Server can access an audio device on the RDP Client.

Direct Actor: The direct actor is the remote application.

Primary Actor: The primary actor is the same as the direct actor.

Supporting Actors: The supporting actors are the TS Server, the RDP Client, and the Audio Device on the RDP Client.

Stakeholders and Interests:

- TS Server: The TS Server manages redirecting audio data from the remote application to the Audio Device on the RDP Client.
- RDP Client: The RDP Client routes audio content from the remote application to the local Audio Device.
- Audio Device on the RDP Client: The Audio Device plays audio content that is sent from the remote application.

Preconditions: The RDP Client is connected to the TS Server. The RDP Connection supports **audio redirection**. The remote desktop or remote application is running on the TS Server. The TS Server has permission to access the audio device on the RDP Client.

Minimal Guarantees: The remote application will be prevented from accessing the audio device on the RDP Client.

Success Guarantee: The audio device will be available to the remote application.

Trigger: The remote application sends audio content to the RDP Client to play on the local audio device.

Main Success Scenario: The remote application running on a TS Server will have the ability to stream audio content to an audio device connected to the RDP Client.

Extensions: None.

3.3.4.12 Log Off from a Remote Session – RDP Client

Goal: The User of the RDP Client logs off from a TS Server, causing the user session on the TS Server to be closed.

Context of Use: The User of the RDP Client wants to terminate an RDP connection.

Direct Actor: The direct actor is the RDP Client.

Primary Actor: The primary actor is the same as the direct actor.

Supporting Actors: The supporting actors are the TS Server, the RDP Client, the Administrator, and the User.

Stakeholders and Interests:

- RDP Client: The RDP Client enables the User to log off the user session.
- TS Server: The TS Server must close a user session and clean up associated resources after a User logs off or an Administrator forces the user session to close.
- Administrator: An administrator may need to force a user session closed using an administrative tool.
- User: The User of the RDP Client wants to close the assigned user session on the TS Server.

Preconditions: Licensing, authentication, authorization, and DNS services are available. A connection exists between the RDP Client and TS Server.

Minimal Guarantees: The user session remains on the TS Server if the logoff is not successful.

Success Guarantee: The user session is closed on the TS Server and associated resources are cleaned up.

Trigger: The User attempts to log off the remote desktop.

Main Success Scenario: After logging off the remote desktop, the TS Server terminates the user session and cleans up resources associated with the session.

Extensions: An Administrator may force a user session to be terminated using an administrative tool.

3.3.4.13 Disconnect From a Remote Session – RDP Client

Goal: The RDP Client disconnects from a TS Server, but the user session remains in a suspended mode for possible later use.

Context of Use: An RDP Client may become disconnected from a TS Server because of network problems, or because the RDP Client is shut down prior to the User logging off the assigned session. When this occurs, the user session remains on the TS Server for a certain amount of time, depending on the configuration of the TS Server. This allows a User to reconnect to the existing session.

Direct Actor: The direct actor is the RDP Client.

Primary Actor: The primary actor is the same as the direct actor.

Supporting Actors: The supporting actors are the TS Server and the User.

Stakeholders and Interests:

- RDP Client: The RDP Client enables the user to disconnect the user session.
- TS Server: The TS Server handles the disconnect request and keeps the session intact for a timeout period configured by the administrator.
- Administrator: An administrator of the TS Server configures the session timeout values for a specified period in which reconnection to the disconnect session is feasible.

Preconditions: Licensing, authentication, authorization, and DNS services are available. A connection exists between the RDP Client and TS Server.

Minimal Guarantees: The RDP Client will be able to terminate the user session.

Success Guarantee: The RDP Client will be able to disconnect from a user session while the TS Server preserves the user session in a disconnected state.

Trigger: None.

Main Success Scenario: The RDP Client will be able to gracefully disconnect the user session so that reconnection to the same user session will be possible at a later time.

Extensions: None.

4 System Context

This section describes the relationship between this system and its environment.

4.1 System Environment

A TS Server depends on a number of prerequisites for it to be configured, applied and used by RDP Clients. There are core networking protocols and services that MUST be open, running and configured to correctly query and respond in order for policy to be applied. Refer to section 4.3.1 for an illustration of the system context.

The network must be capable of supporting TCP/IP traffic such as DNS and LDAP communication to support the lookup, transport and transfer of services and policy data. Additionally the network must also support authentication and authorization. As part of this protocol access, any host based firewalls residing on the RDP Client and TS Servers must have open **Transmission Control Protocol (TCP)** ports for each of these services in addition to port 3389, Remote Desktop Protocol (RDP) port.

For provisioning a remote user, the Terminal Services System depends on services supporting User Profiles and directory services. For effective management of a remote desktop and remote applications running on a virtual machine in Windows 7 implementations, suitable image management services and virtualization services are needed.

4.2 System Assumptions and Preconditions

The Terminal Services System assumes that:

- The RDP Client and TS Server have network connectivity over TCP/IPv4 or IPV6.
- The RDP Client initiating the connection is using an implementation of the Remote Desktop Protocol (RDP).
- The TS Server is configured and any firewall between the RDP Client and TS Server is configured to allow RDP traffic.
- The TS Server is actively listening for RDP Client connections on a registered port.

4.3 System Relationships

4.3.1 Black Box Relationship Diagram

The following figure illustrates the Terminal Services System and components that interact with it.

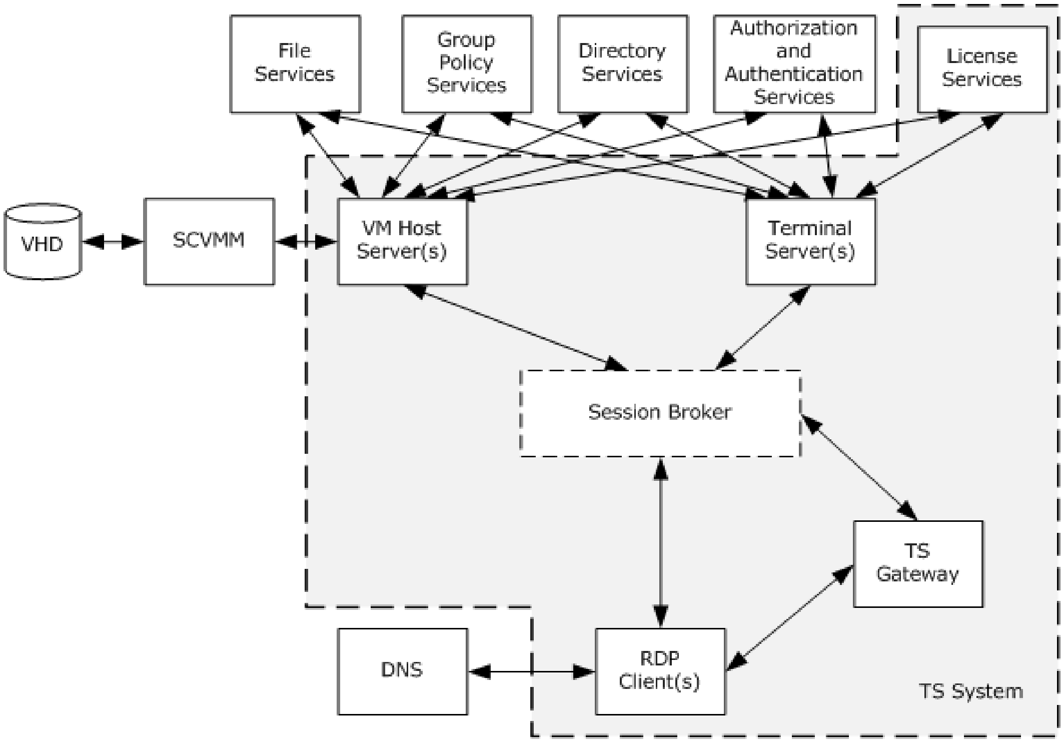


Figure 6: Terminal Services System overview

TS Servers support the use of external systems, such as **directory services (DS)**, licensing services, domain services, and security services. These services use protocols that are not a part of the Terminal Services System.

The Session Broker component, shown with a dashed line in the previous diagram, is an optional component that is not necessary for an RDP Client to connect to a TS Server. Session Broker services assign users of RDP Clients to user sessions on TS Servers and can use an algorithm that balances the work load between TS Servers.

4.3.2 System Dependencies

The Terminal Services System relies on the following:

- Basic TCP/IP network connectivity between the system sub-roles and dependent components as shown in the previous figure.
- Authentication and authorization services by domain controllers or Kerberos systems. Authentication services might depend on a certificate infrastructure to support SSL.
- DNS for address resolution.
- Directory Services or other components (such as a Session Broker, or a System Center Virtual Machine Manager (SCVMM)) for assigning user sessions or virtual machines.
- File Services for Terminal Services publication of remote desktops and remote applications.

4.3.3 System Influences

There are a variety of factors that can affect how the Terminal Services System can be deployed. Some examples of these factors are as follows:

- The Group Policy system can alter capabilities of the remote experience in the Terminal Services System through user-specific policies. A profile server that stores user-specific profiles (mandatory and roaming) can alter the capabilities of the remote experience. System Center for Virtual Machine Management (SCVMM) is used for desktop image management in a large Remote Desktop Virtualization environment. Internet Access Gateway (IAG) influences the TS Gateway in restricting what an RDP Client can access.
- Administrative policies can influence the remote experience and capabilities of the RDP Client and TS Server.
- Cluster services (such as server farms) can ensure high availability Terminal Services System configuration.

4.4 System Applicability

Remote Desktop Services is applicable in server based centralized computing scenarios, where users can access their desktop or application remotely across a TCP/IP network through rich or thin clients.

4.5 System Versioning and Capability Negotiation

The Terminal Services System provides capability-based services, as described in [MS-RDPBCGR]. The capabilities and requirements of a client requesting a connection are established during the Remote Desktop Protocol (RDP) handshake. Information exchanged about capabilities includes data such as supported **protocol data units (PDU)** and **drawing orders**, desktop dimensions, allowed color depths, input device support, and cache structures. The client and terminal server perform merge operations between their capabilities so that all RDP communication is consistent with negotiated expectations and can be processed by each party.

4.6 System Vendor-Extensible Fields

The Terminal Services System does not define any vendor-extensible fields beyond those described in the specifications of the protocols supported by the system.

5 System Architecture

This section describes the basic structure of the system and the interrelationships among its parts, consumers, and dependencies.

5.1 Abstract Data Model

The abstract data model for the Terminal Services System is a union of the data models for the various components of the system: TS Server, RDP Client, TS Gateway, and other commonly used components described in this section. Not all components of the Terminal Services System are used all the time. See the component protocol documentation for more detailed information on all abstract data models covered in the Terminal Services System.

The following table lists shared high-level data elements described within this section. For more granular elements, please refer to [MS-RDPBCGR], and [MS-TSGU], as well as [MS-GPSO].

Shared data elements

Data elements shared between ADMs	ADMs sharing elements	Persisted at	Documents describing protocols using the elements
User Credential	TS Server and RDP Client and TS Gateway	Active Directory	[MS-RDPBCGR] [MS-TSGU]
Client Access License	TS Server and RDP Client	RDP Client	[MS-RDPELE]
Client Graphic Capability	TS Server and RDP Client	RDP Client	[MS-RDPBCGR]
Redirected Device List	TS Server and RDP Client	RDP Client	[MS-RDPBCGR]
Redirected Drive List	TS Server and RDP Client	RDP Client	[MS-RDPEFS]
User Policy	TS Server and TS Gateway	TS Server and Active Directory	[MS-GPSO]
TargetServerName	RDP Client and TS Gateway	TS Gateway	[MS-TSGU]
Client machine name	RDP Client and TS Gateway	RDP Client	[MS-TSGU] [MS-RDPBCGR]

5.1.1 TS Server

The abstract data model of the TS Server is illustrated as follows.

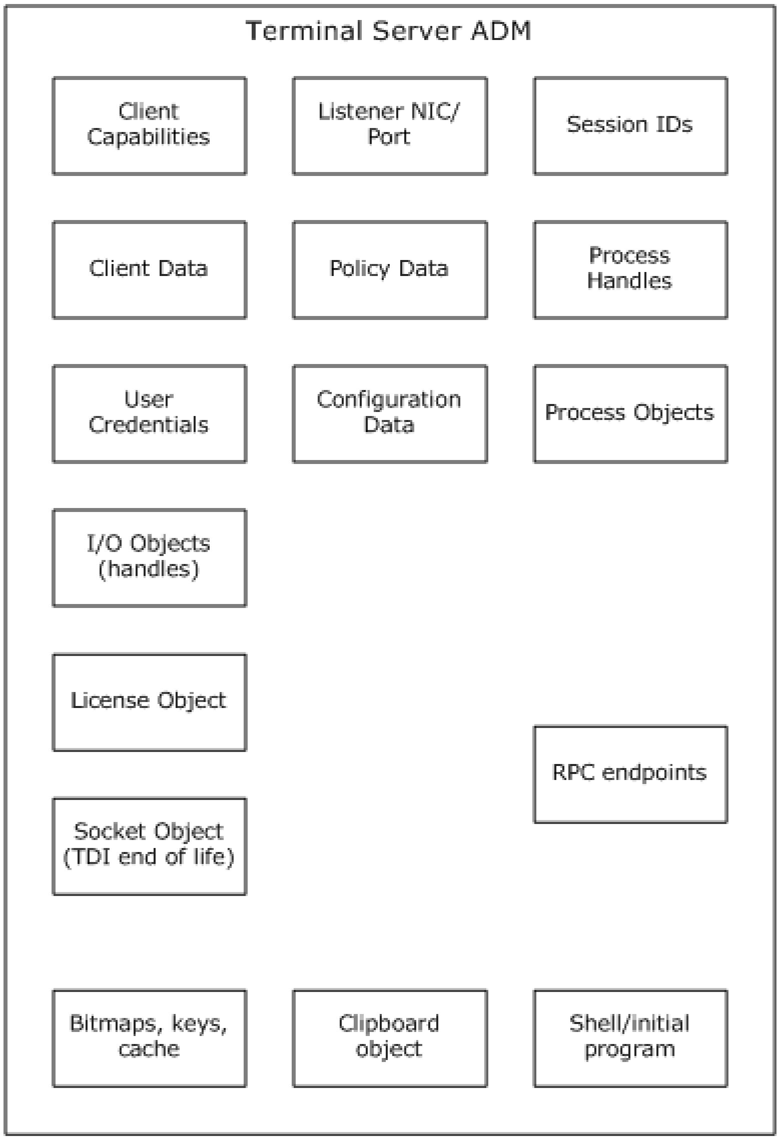


Figure 7: TS Server abstract data model

The elements in the TS Server abstract data model are described as follows:

Client Capabilities: The Client Capabilities store contains capability sets received from the RDP Client in the Confirm Active protocol data unit (PDU) (see [MS-RDPBCGR] section 2.2.1.13.2). The TS Server can only send data supported by the RDP Client. For example, if the RDP Client does not support Fast-Path output, the TS Server will only send Slow-Path output PDUs. In effect, the TS Server ensures that all of the RDP traffic which it sends on the wire is consistent with the expectations of the RDP Client as described by the data held in the Client Capabilities store. Client Capabilities include Client Licensing Capabilities encoded in the Client Licensing Encryption Ability store. This store determines whether the RDP Client has the ability to handle encrypted licensing packets when using RDP security mechanisms. This information is communicated to the server as part of the Security Exchange PDU (see [MS-RDPBCGR] section 2.2.1.10).

Client Data: Configuration data from the RDP Client which states the preferences of the RDP Client or user, for example, color depth, screen resolution, etc.

Client Identity : The TS Server assigns a unique identity string to each RDP Client connection.

User Credentials: User name, password.

I/O Channel IDs: This store contains the **Multipoint Communication Service (MCS)** channel identifiers for I/O communication through virtual channels.

I/O Objects (handles): Handles for mouse and keyboard display.

License Object: Stores licensing related data for license validation.

Connection State: Stores current state of session-client relationship.

Listener NIC/Ports: Stores the port/NIC mapping on which a listener thread waits for a connection request.

Server Policy: Configuration policy applied to all listeners, connections, sessions.

User Policy: Configuration policy applied to a connection or session after user is logged on.

Registration with Local Session Manager (COM): An instance of a relationship or interface between a TS Server and a Local Session Manager (LSM).

RPC Endpoints: Name or entry for TS Server remote procedure call (RPC) APIs.

Pluggable Protocol List: A stored list of protocols to load during service startup.

Session Broker (member of): Optional session broker membership (FQDN of Session broker).

Timer Objects: Timers to enforce various policies (such as disconnect an idle session after some number of minutes).

Events: Global events signaled by TS Server to broadcast a state, such as TS Server is ready.

Terminal Object: Collection of several of the previously mentioned objects which are treated as an instance of a terminal to which a session will be bound by the LSM.

Bitmap, keys, cache:

- Bitmap: Portion of a drawing that is rendered as a bitmap (instead of drawn through Graphics Device Interface (GDI) primitives).
- Keys: When the server sends a bitmap to the client it can first check the Persistent Bitmap Keys store to determine whether the client already has the bitmap in a local bitmap cache and save on bandwidth.
- Cache: Similar to bitmap keys, the Pointer Image Cache contains a collection of pointer images sent to the client in Color Pointer Updates and New Pointer Updates.
- Clipboard Object: An object in a session space to help with clipboard redirection.
- Shell or Initial Program: A shell or initial program is a stored item which controls which application starts after user logon. Typically it is the Windows Explorer shell for a full-desktop, but it could be some other application.

5.1.2 RDP Client

The abstract data model for the RDP Client is illustrated as follows.

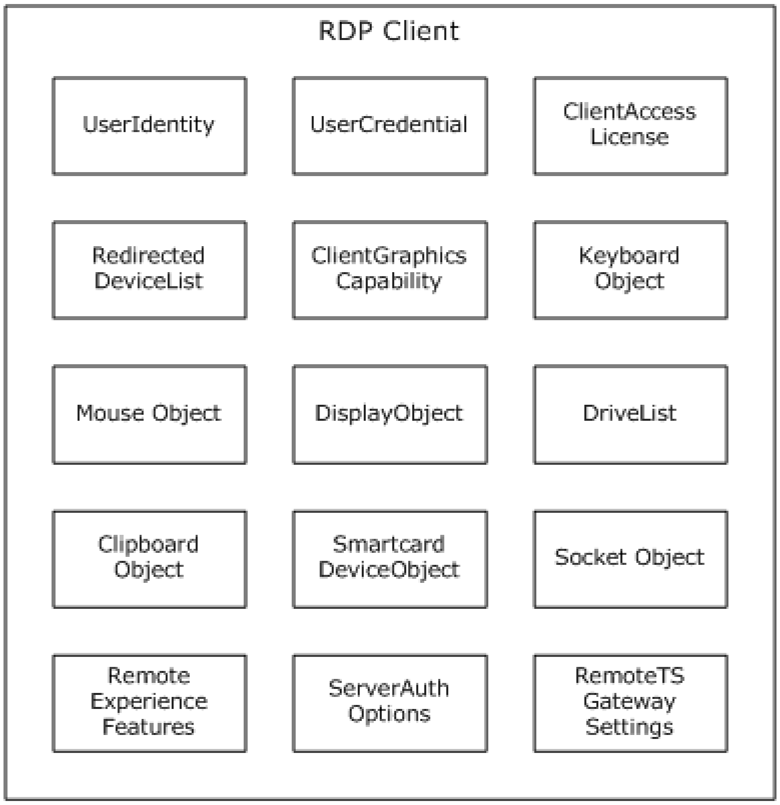


Figure 8: Abstract data model of RDP Client

The elements in the abstract data model of the RDP Client are:

UserIdentity: The RDP Client must provide user identity to the TS Server to create a remote session for the user for authentication purposes. This takes the form of **domain** name and username. User identity is acquired through a trusted authorization system on the client.<1>

UserCredential: The user credential is in the form of a password or smart card PIN that will be used for authentication purposes while establishing a remote connection. The user credential is again acquired through a trusted security component.

ClientAccessLicense: The client access license is a certificate that is either configured by the administrator of a remote server or acquired by the RDP Client when a remote connection is established for the first time, with a TS Server to be provided to the remote server when creating a remote session.

RedirectedDeviceList: A list of device objects that will be redirected to the remote session. This data element will be used in constructing the initial Remote Desktop Protocol (RDP) capability handshake with the remote session.

ClientGraphicsCapability: The client specified or configured screen resolution and color depth information used in the client capability part of RDP protocol.

KeyboardObject: An input device object that captures keyboard input before sending it to the remote session using the RDP protocol.

Mouse Object: An input device object that captures mouse input before sending it to the remote session using the RDP protocol.

DisplayObject: An output device object representing the display device to which graphics data will be written. Graphics data is received using the RDP protocol from the remote session.

DriveList: The user specified hard disk drives that are made available for a remote file system.

ClipboardObject: The element to store clipboard data during clipboard operation.

SmartcardDeviceObject: The device object that will be used to redirect a smart card device.

SocketObject: Sockets for network connection for communicating with the server.

RemoteExperienceFeatures: These features include desktop background, font smoothing, desktop composition, menu and window animation, themes, and bitmap caching.

ServerAuthOptions: Options for TS Server authentication that will be used during connection establishment time.

RemoteTSGatewaySettings: Settings used for establishing a connection through TS Gateway such as automatic TS Gateway detection, TS Gateway name, and log on settings.

5.1.3 TS Gateway

This section describes a conceptual model of TS Gateway. The data elements illustrate the required state information that is needed on the TS Gateway to facilitate establishing an remote procedure call (RPC) tunnel (see [MS-RPCH]) from the RDP Client to the TS Gateway. After establishing this tunnel, a communication channel is created for Remote Desktop Protocol (RDP) data travelling from the RDP Client to the TS Server.

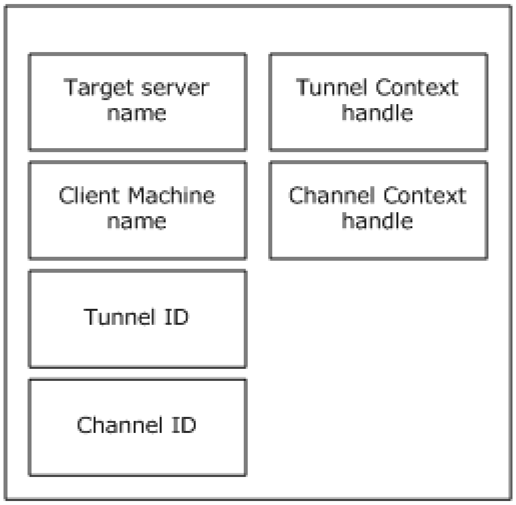


Figure 9: Abstract data model of TS Gateway

The elements in the abstract data model of a TS Gateway are as follows:

Target Server Name: A string of Unicode characters that cannot exceed 512 bytes, including the NULL terminator. The server name applies to the computer that the TS Gateway server connects to. This is the computer name which is returned by running **gethostname** on the computer.

Client Machine Name: A string of Unicode characters that cannot exceed 513 bytes, including the NULL terminator. The Client Machine name refers to the computer that runs the TS Gateway client. It is possible for the Client Machine name to be the same as the server name (in value) if the client and the server run on the same physical computer. The Client Machine name refers to the computer name only as determined by **gethostname**.

Tunnel ID: Represents the tunnel identifier for tracking purposes on the TS Gateway server. It may be used by the TS Gateway server as an index to access the tunnel context object. During the process of connection, the client calls **TsProxyCreateTunnel** to create a tunnel to the gateway. The Tunnel ID, which is then generated on the server, is stored and may later be used for subsequent tunnel-related operations. After the tunnel is set up, the client calls **TsProxyCreateChannel** to create a channel to the target server name. The Channel ID, which is then generated on the server, is stored and can later be used for subsequent channel-related calls.

Channel ID: Represents the channel identifier for tracking purposes on the TS Gateway server. It can be used by the TS Gateway server as an index to access the channel context object.

Tunnel Context handle: An RPC context handle for the TS Gateway client to TS Gateway server connection represented by an array of 20 bytes on the TS Gateway server.

Channel Context handle: An RPC context handle for the TS Gateway client to TS Gateway target server connection represented by an array of 20 bytes on the TS Gateway server.

5.1.4 Abstract Data Model Supporting Virtual Channels

Static virtual channels are established when an RDP Client connects to a TS Server (see [MS-RDPBCGR]). One example of a static virtual channel is a dynamic virtual channel (DVC) (see [MS-RDPEDYC]). This type of virtual channel is called dynamic because the endpoints do not have to use component protocols of the Terminal Services System, but exists on both the TS Server and the RDP Client. The endpoints can be connected or disconnected any time after the DVC is established during the initial connection sequence between the RDP Client and the TS Server.

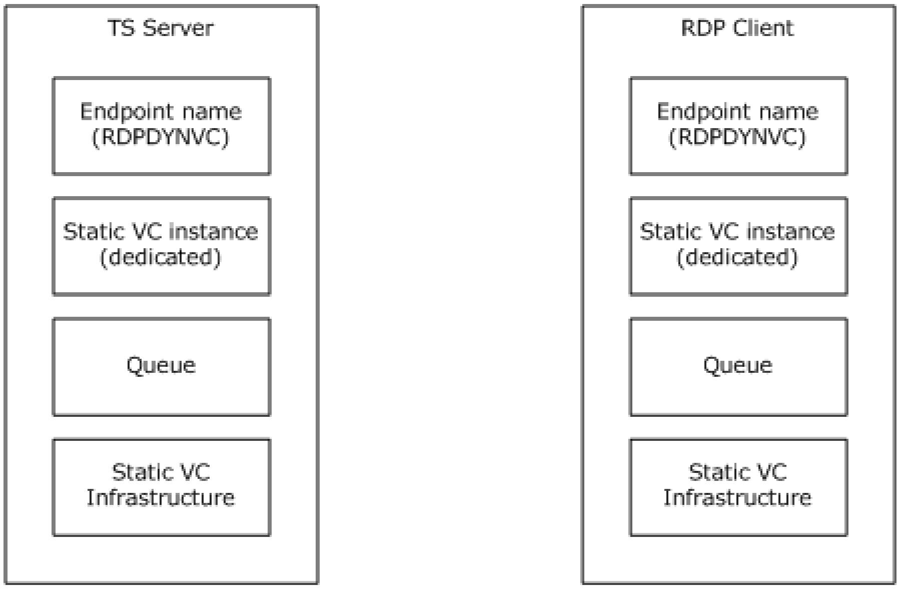


Figure 10: TS Server and RDP Client ADM supporting dynamic virtual channels

The elements of the abstract data model supporting DVCs are the same as those supporting static virtual channels. These elements are as follows:

Endpoint Name: The predefined static name of a static virtual channel used as a communication path through which arbitrary dynamic channels are negotiated and through which the appropriate data packets are sent.

Static Virtual Channel Instance: The data abstraction for the static virtual channels used.

Queue: A queue data structure used for managing traffic of DVC packets through the dedicated static virtual channel.

Static Virtual Channel Infrastructure: All the data and routines for handling communication for static virtual channel.

5.2 White Box Relationships

[MS-RDPBCGR] is the core protocol used to communicate between a Terminal Server Client and a Terminal Server or Remote Desktop Virtual Host. [MS-TSGU] describes how a Terminal Server Gateway tunnels RDP wire traffic. A Remote Desktop Session Broker redirects [MS-RDPBCGR] traffic without inspecting or tampering with any of the data. [MS-RDPBCGR] messages are described in [MS-RDPBCGR], Message Syntax (section 2.2).

To support additional functionality, extensions to [MS-RDPBCGR] can be implemented. Protocol extensions are implemented by the addition of protocol packets (such as [MS-RDPEPS]) or tunneled virtual channel protocols (such as [MS-RDPEFS]). See Member Protocol Roles section 5.3.1, for a description of the relationships between the family of RDP protocols.

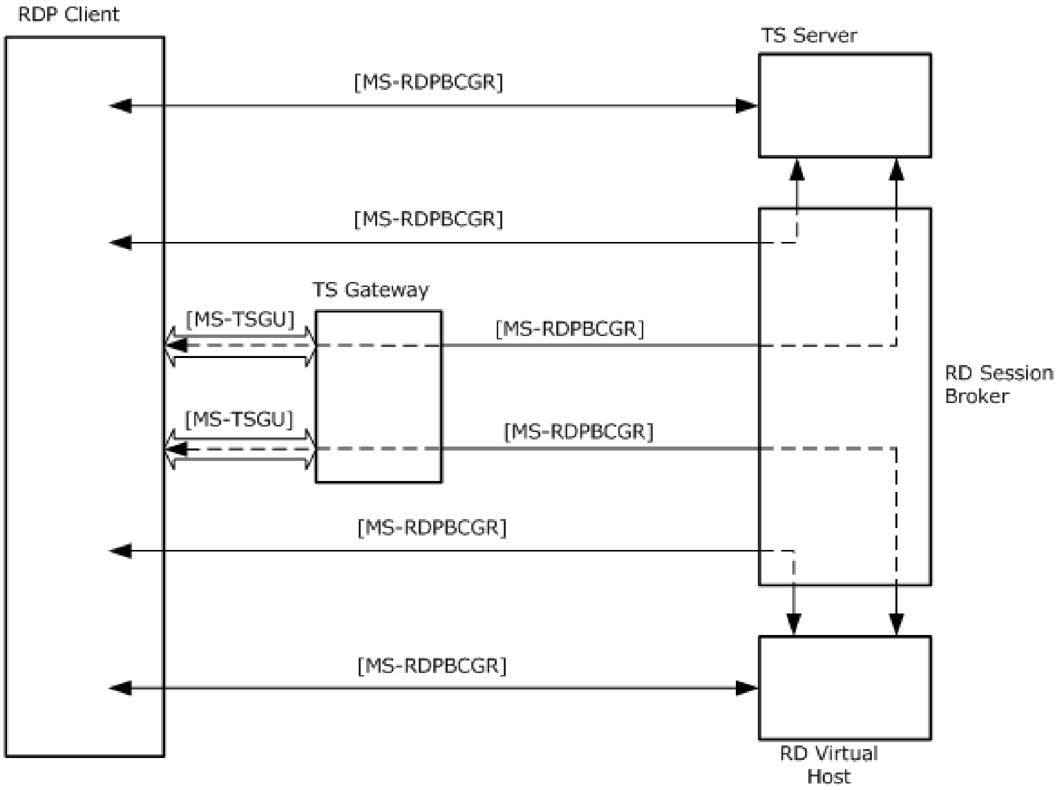


Figure 11: Basic Terminal Services System components and the core protocols used between them

5.2.1 Data Exchange

The Remote Desktop Protocol (RDP) is used to tunnel graphical data, input data, and device data (and other communication) between an RDP Client and a TS Server. RDP also defines an extensible virtual channel mechanism. Each virtual channel acts as an independent data stream. The RDP Client and TS Server examine the data received on each virtual channel and route the data stream to the appropriate endpoint for further processing. The necessary static virtual channels are opened at the start of the session during handshaking, and remain open until the session is closed.

One virtual channel protocol, described in [MS-RDPEDYC], allows the use of a dynamic virtual channel (DVC). A DVC is opened when a connection is established between the RDP Client and the TS Server, but the endpoints can be dynamically connected after the initial connection sequence. For example, a Plug and Play device can be connected to an RDP Client while the RDP Client is connected to a TS Server, and the Plug and Play device can subsequently be accessed by the TS Server without re-establishing the RDP connection to the RDP Client.

5.3 Member Protocol Functional Relationships

5.3.1 Member Protocol Roles

The component protocols of the Terminal Services System have the roles listed in this section. The first three roles are accomplished by the main protocol on which the Terminal Services System is based (the Remote Desktop Protocol (RDP) as described in [MS-RDPBCGR]). Member protocol roles are as follows:

- **Establishing a connection between an RDP Client and a TS Server**: Protocols performing this role are described in [MS-RDPBCGR] and [MS-TSGU].
- **Transporting graphical information from a Remote Desktop or remote application running on the TS Server to an RDP Client**: Protocols performing this role are described in [MS-RDPBCGR] and [MS-RDPEGDI].
- **Transporting user input from an RDP Client to a TS Server**: The protocol performing this role is described in [MS-RDPBCGR].
- **Supporting the use of Remote Applications**: The protocol performing this role is described in [MS-RDPERP].
- **Supporting licensed use of a TS Server**: The protocol performing this role is described in [MS-RDPELE].
- **Transporting device data or resource data between an RDP Client and a TS Server**: The protocols performing this role are described in [MS-RDPECLIP], [MS-RDPEFS], [MS-RDPESP], [MS-RDPEPC], [MS-RDPESC], [MS-RDPERP], [MS-RDPEMC], [MS-RDPEA], [MS-RDPEDYC], [MS-RDPEPNP], [MS-RDPCR2], [MS-RDPNSC], [MS-RDPRFX], [MS-RDPEUSB], and [MS-RDPEXPS].
- **Configuring and Managing a TS Server**: The protocol performing this role is [MS-TSTS].

The following figure illustrates the relationships between the protocols that are used for communication between an RDP Client and a TS Server. Implementations of RDP may support a combination of these protocol extensions, or just RDP itself as described in [MS-RDPBCGR].

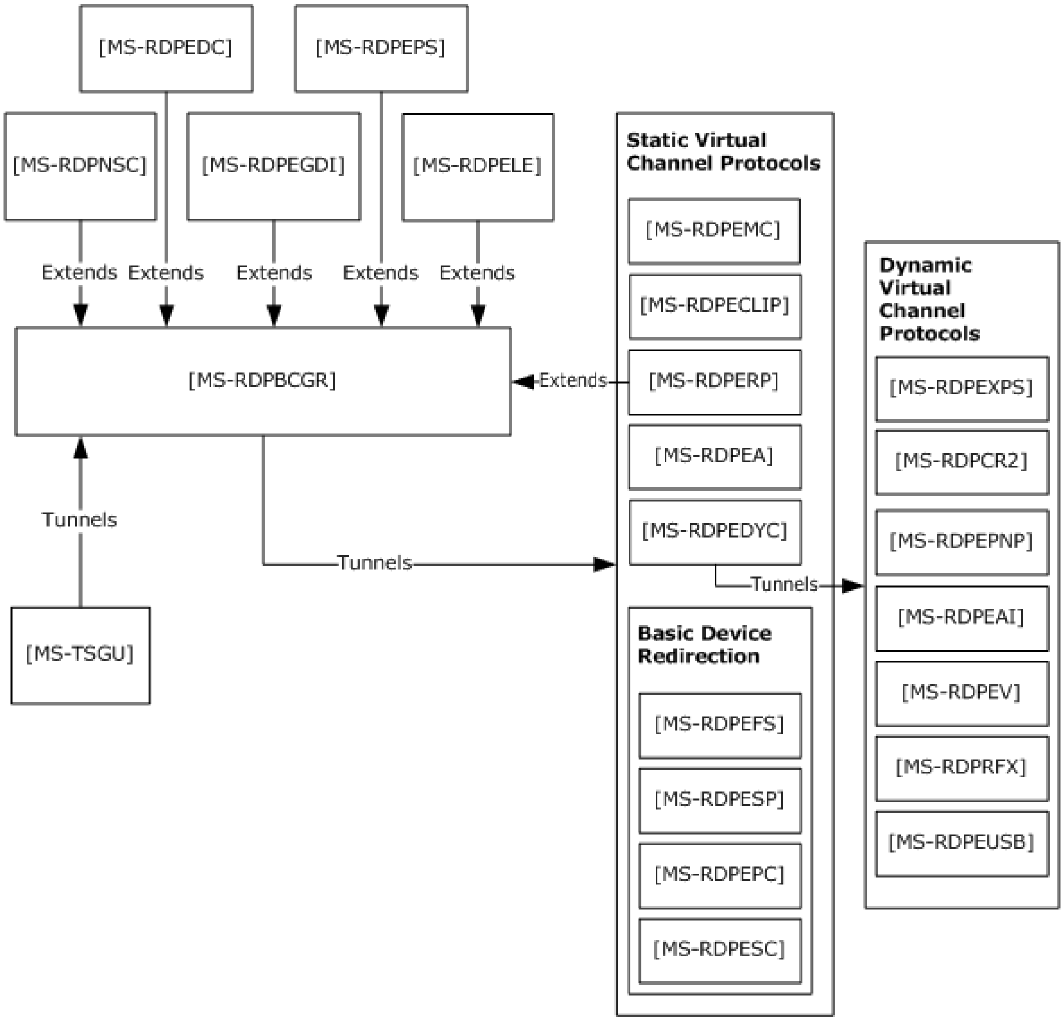


Figure 12: Terminal Services System protocols overview and relationships

[MS-RDPBCGR] is based on the ITU (International Telecommunication Union) T.120 series of protocols. The T.120 standard is composed of a suite of communication and application-layer protocols that enable implementers to create compatible products and services for real-time, multipoint data connections and conferencing.

The following protocols are tunneled within an [MS-RDPBCGR] static virtual channel ([MS-RDPBCGR] section 1.3.3):

- Multiparty Virtual Channel Extension [MS-RDPEMC]
- Clipboard Virtual Channel Extension [MS-RDPECLIP]
- Audio Output Virtual Channel Extension [MS-RDPEA]
- Remote Programs Virtual Channel Extensions [MS-RDPERP]
- Dynamic Channel Virtual Channel Extension [MS-RDPEDYC]

- File System Virtual Channel Extension [MS-RDPEFS]
- Serial Port Virtual Channel Extension [MS-RDPESP]
- Print Virtual Channel Extension [MS-RDPEPC]
- Smart Card Virtual Channel Extension [MS-RDPESC]

The following protocols are tunneled within an [MS-RDPEDYC] extended dynamic virtual channel:

- XPS Printing Virtual Channel Extension [MS-RDPEXPS]
- Plug and Play Devices Virtual Channel Extension [MS-RDPEPNP]
- Video Virtual Channel Extensions [MS-RDPEV]
- Audio Input Virtual Channel Extension [MS-RDPEAI]
- Composited Remoting V2 Extensions [MS-RDPCR2]

The following protocols extend [MS-RDPBCGR]:

- Licensing Extensions [MS-RDPELE]
- Session Selection Extension [MS-RDPEPS]
- Graphics Device Interface (GDI) Acceleration Extensions [MS-RDPEGDI]
- Remote Programs Virtual Channel Extension [MS-RDPERP]

The following protocol tunnels [MS-RDPBCGR]:

- Gateway Server Protocol [MS-TSGU]

5.3.2 Member Protocol Groups

Some of the functionality of the Terminal Services System requires that protocols are used as groups. A wide permutation of protocol groupings can be used, depending on the configuration of the RDP Client and the TS Server. When a connection is established between an RDP Client and TS Server capabilities of and requirements are exchanged, and channels supporting necessary protocols are established. The member protocol groups of the Terminal Services System are as follows:

- **Establishing a secure connection between an RDP Client and a TS Server**. Although the Remote Desktop Protocol (RDP) allows an RDP Client to securely connect to a TS Server, in order to connect across a domain boundary an RDP Client MUST use a VPN or the protocol described in [MS-TSGU] to first connect to a TS Gateway server.
- **Transporting graphical information from a remote desktop or remote application running on the TS Server to an RDP Client.** Although RDP supports drawing primitives, more efficient transport of graphical data (in the form of drawing orders) is provided by using RDP in combination with the protocol described in [MS-RDPEGDI].
- **Transporting known device data or resource data between an RDP Client and a TS Server**. To transport data using known endpoints, static virtual channels can be initialized when an RDP Client establishes a connection with a TS Server by using RDP in combination with any or all of the following protocols: [MS-RDPECLIP], [MS-RDPEFS], [MS-RDPESP], [MS-RDPEPC], [MS-RDPESC], [MS-RDPERP], [MS-RDPEMC], and [MS-RDPEA].

- **Transporting unknown device data or resource data between an RDP Client and a TS Server**. To transport data between endpoints that are dynamically established after a connection is established between an RDP Client and a TS Server, a static virtual channel supporting dynamic virtual channels (DVC) MUST first be established during the initial connection in anticipation of this need. The protocol described in [MS-RDPEDYC] is used in combination with RDP to create a DVC. Protocols that are also part of this group or those defined in the Terminal Services System to support Plug and Play devices are described in [MS-RDPEPNP], and XPS Printing as described in [MS-RDPEXPS]. However, DVCs can also support protocols that are not formally a part of the Terminal Services System, as long as the protocols are supported by both the RDP Client and the TS Server.

5.4 System Internal Architecture

5.4.1 TS Server

To establish the core functionality and capabilities of a TS Server, the connection to an RDP Client proceeds through various stages. This sequence provides the necessary data for the TS Server to construct an abstract data object representing a remote terminal, with I/O channels and attributes. After the remote terminal object is constructed, the TS Server creates a uniquely identifiable session with a graphics desktop attached to the remote terminal object. After the session is created, the TS Server redirects graphical data from various graphics objects through the remote terminal object to the RDP Client using the Remote Desktop Protocol (RDP). A TS Server also exposes remote procedure call (RPC) endpoints for manageability and configuration. A TS Server also accepts policy data that prohibits or limits various functions. When a connection between an RDP Client and TS Server is closed, appropriate cleanup steps are taken.

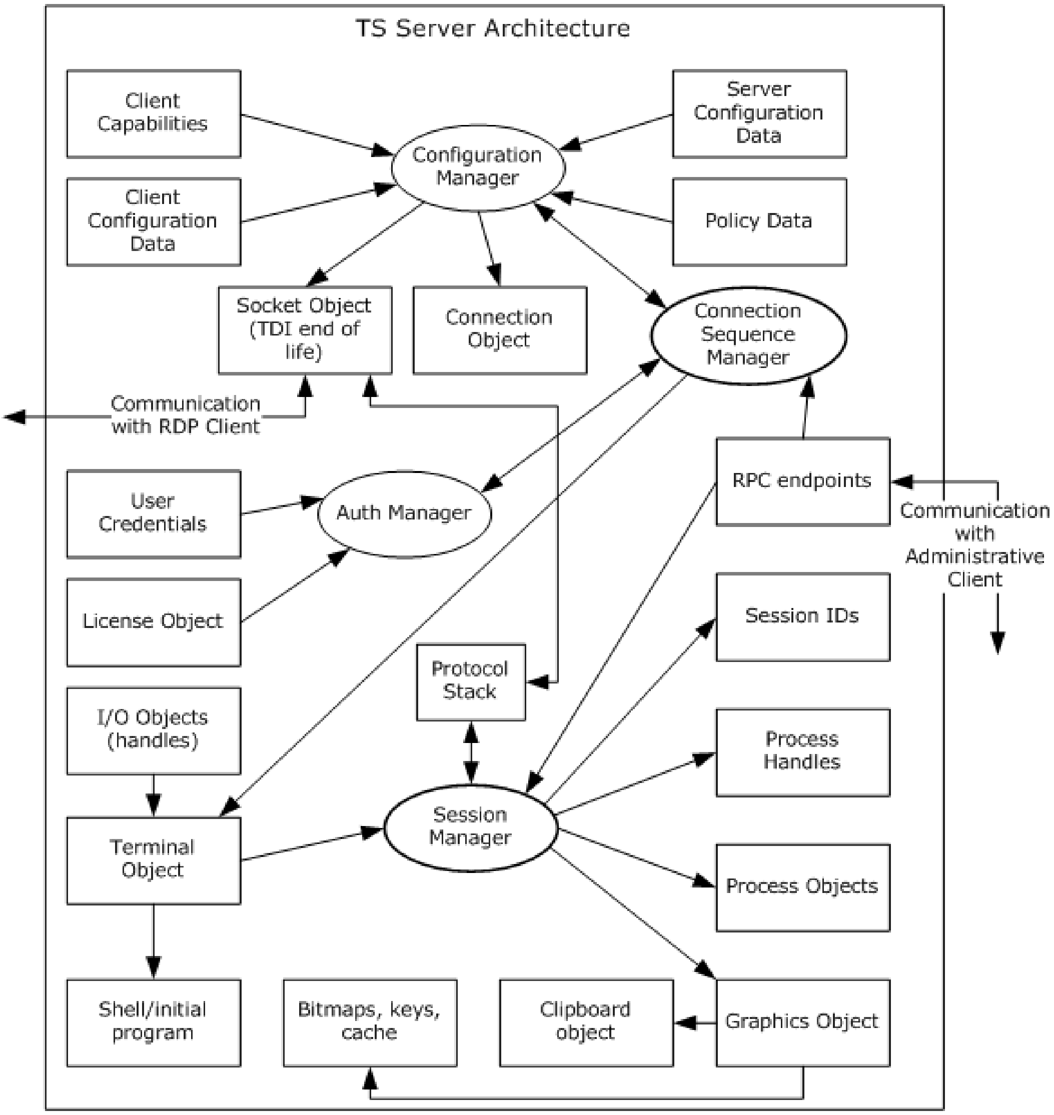


Figure 13: TS Server architecture

5.4.2 RDP Client

The abstract RDP Client architecture is illustrated in the following figure. The primary architectural components are the Remote Desktop Protocol (RDP) Protocol Engine, RDP Client UI and display, and the Keyboard/Mouse Redirection component that interacts with the RDP Protocol Engine. The RDP Protocol Engine leverages platform capabilities such as networking, security, and support for device enumeration. When an RDP Client starts up, it initializes shared data elements by reading through the UI or a configuration file. The RDP Client obtains a handle to the display, Keyboard and Mouse, and other devices that are to be redirected to the remote session on the TS Server. To establish a connection, the RDP Protocol Engine initiates an X.224 exchange to the specified remote endpoint on the TS Server through the TCP/IP transport stack. Once the connection is established, the RDP Protocol Engine exchanges User Identity and User Credentials for authentication. After the user is authenticated, a license (ClientAccessLicense) is provided to the TS Server to validate authorization. At the end of this exchange, the connection state moves into an established state. Then the graphical elements of the remote session are redirected to the RDP Client and the RDP Protocol Engine routes the elements to the designated graphics device handle. Similarly, user input on the RDP Client is read from the keyboard/mouse devices and sent over the connection to the remote session on the TS Server.

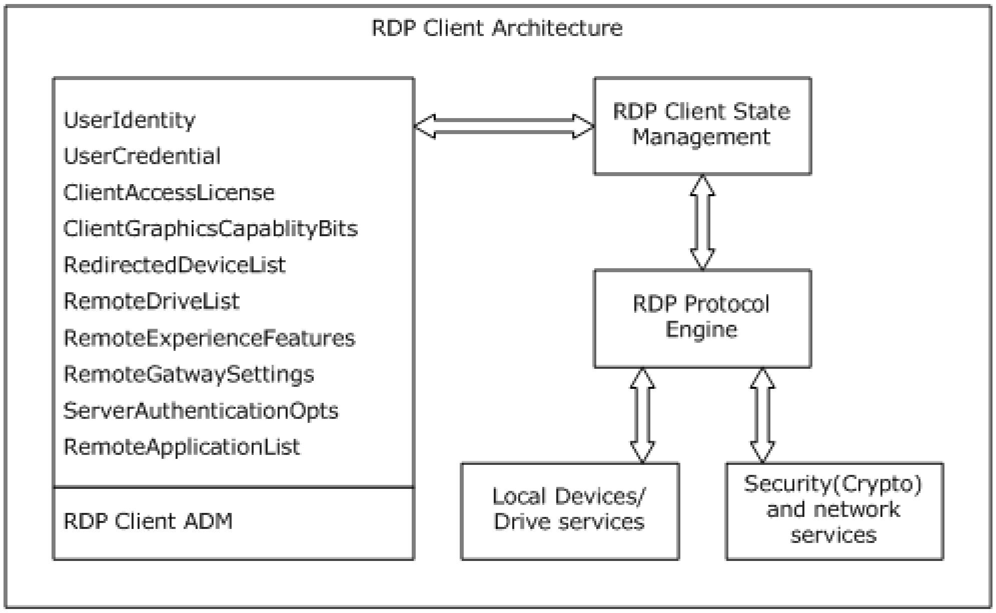


Figure 14: RDP Client component architecture

The following diagram shows the protocol layering within the RDP Client architecture.

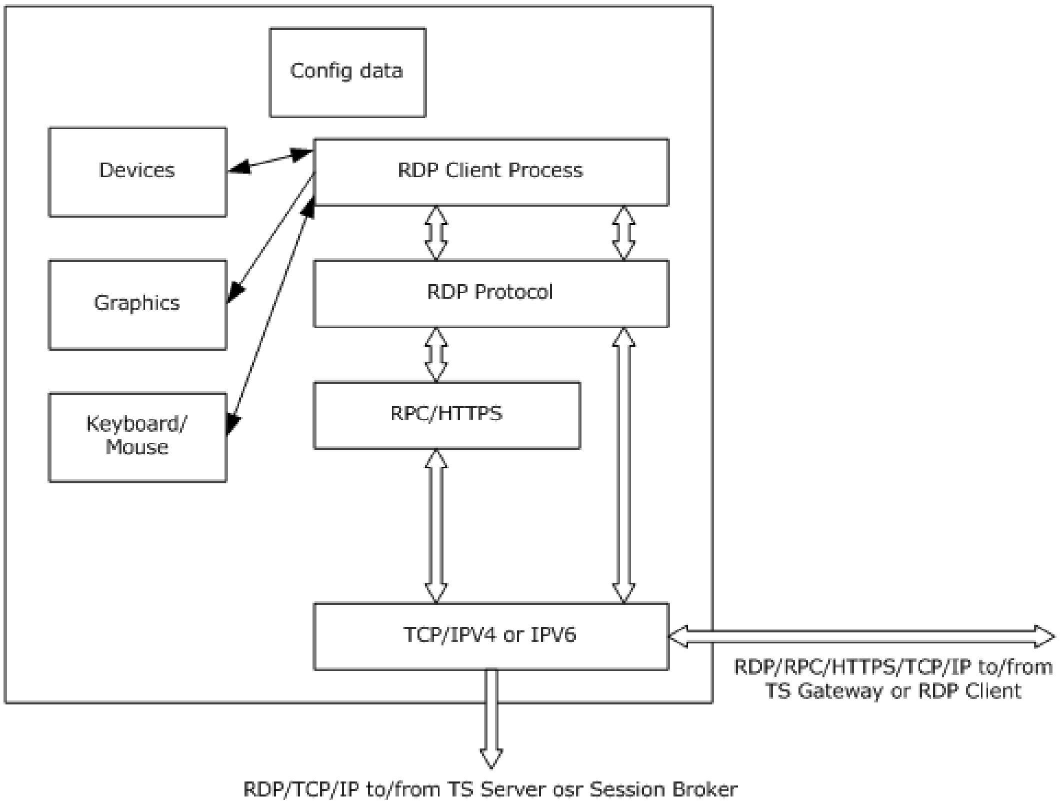


Figure 15: Protocol layering within the architecture of the RDP Client

5.4.3 TS Gateway

The TS Gateway is a proxy server that establishes a tunnel utilizing a remote procedure call (RPC) interface over HTTP as transport between the RDP Client and the TS Gateway. The TS Gateway establishes a channel between the TS Gateway and the TS Server using an RPC interface. A channel is established for every remote connection between the RDP Client and the TS Server via the TS Gateway.

The RDP Client initiates a connection using an RPC call that uses a **tunnelSetup** call over HTTPS to the TS Gateway. A certificate provided by the RDP Client is used to check remote access policy before the tunnel is established.

Once the tunnel is established, a channel is created between the TS Gateway and the TS Server. RDP data, tunneled through the RDP/HTTP channel, is sent by proxy to the TS Server using the channel handle.

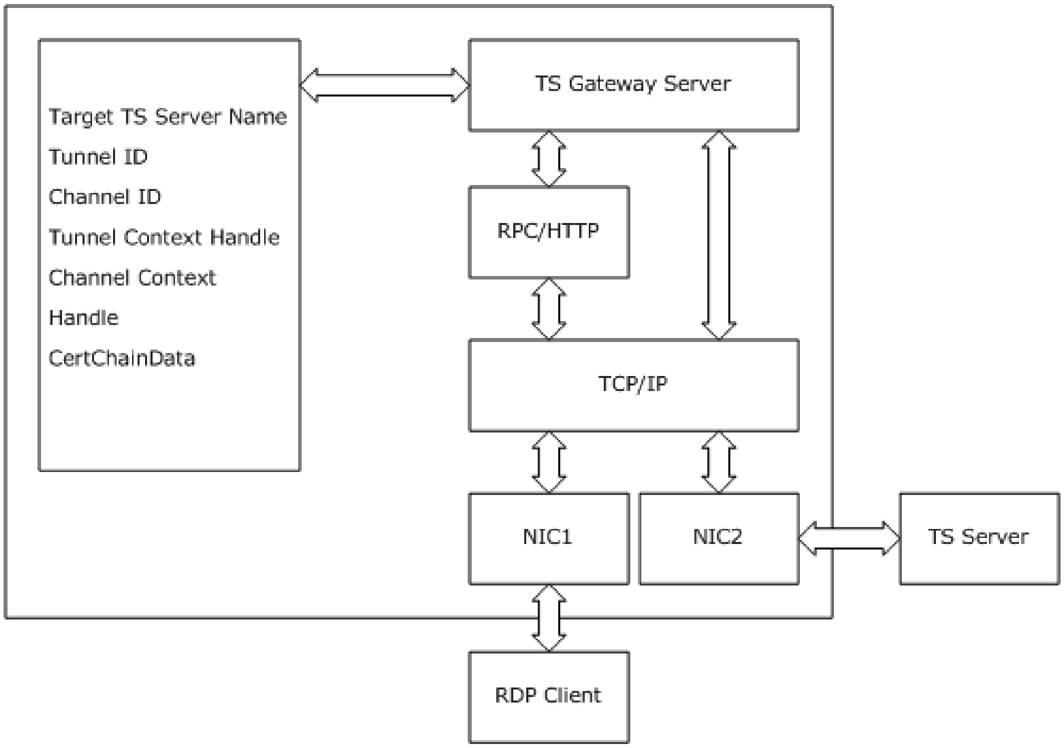


Figure 16: TS Gateway component architecture

5.4.4 Dynamic Virtual Channel

A dynamic virtual channel (DVC) is a channel that is tunneled within a static virtual channel with a predefined name (for example, RDPDYNVC in the following figure) that schedules arbitrary packets tagged for arbitrary source/destination associations. In a DVC, such packets can be routed from the correct source endpoint to the correct destination endpoint. After a static virtual channel is designated to act as a DVC, the static virtual channel listens for a request to create a virtual channel abstraction managed by unique tags defining the source, the destination and any packet association. After the virtual channel abstraction is created, communication can take place between an endpoint on the RDP Client and an endpoint on the TS Server. When a DVC is closed, appropriate cleanup steps are taken.

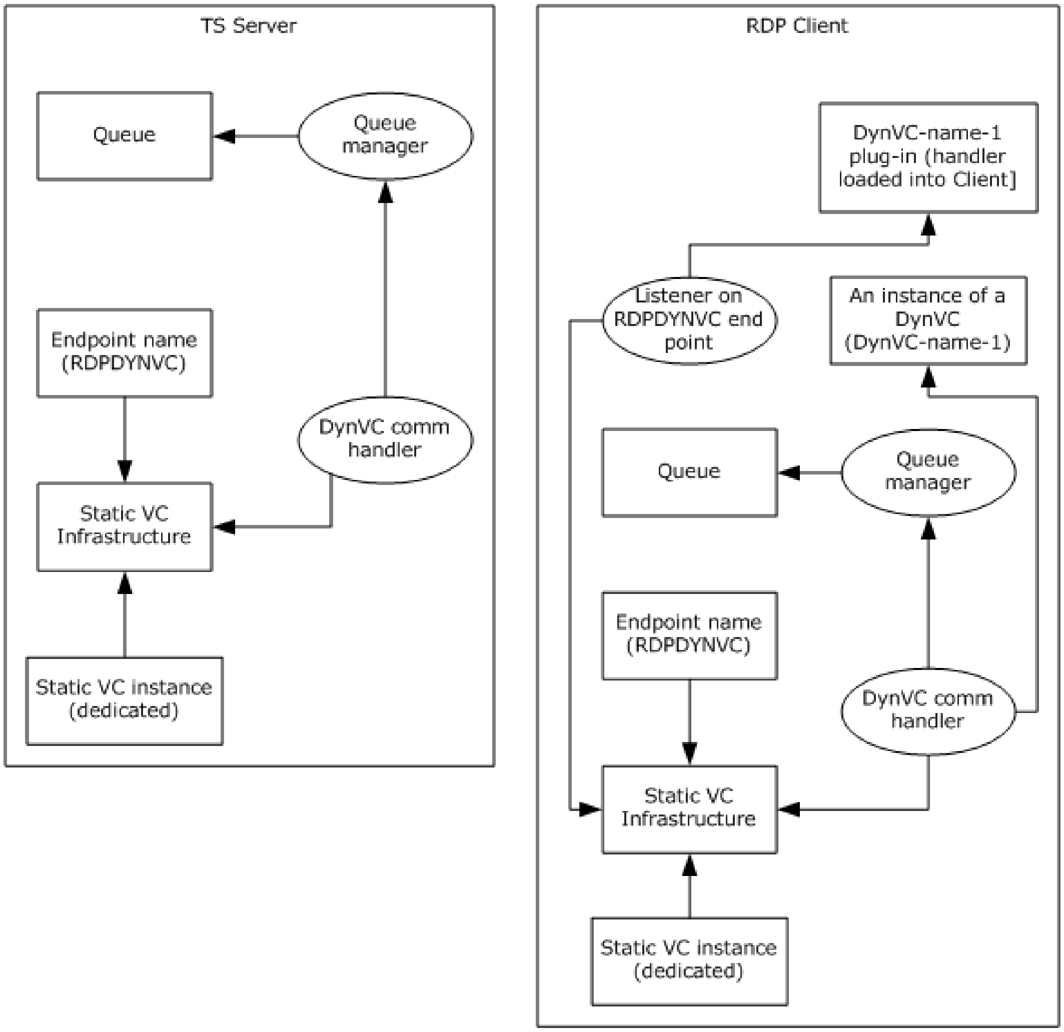


Figure 17: Dynamic Virtual Channel component architecture

5.4.5 Plug and Play

The Plug and Play architecture supporting redirection in the Terminal Services System is based on several core operating system components and services. On the TS Server, a dynamic virtual channel (DVC) is established to a Virtual Function Driver endpoint that communicates device requests to a Virtual Plug and Play component endpoint on the RDP Client. This virtual component on the RDP Client redirects I/O to the actual function driver already present on the RDP Client. On the TS Server, the Plug and Play and I/O manager route requests to the TS Server's Virtual Function Driver based on which devices the driver is registered for handling.

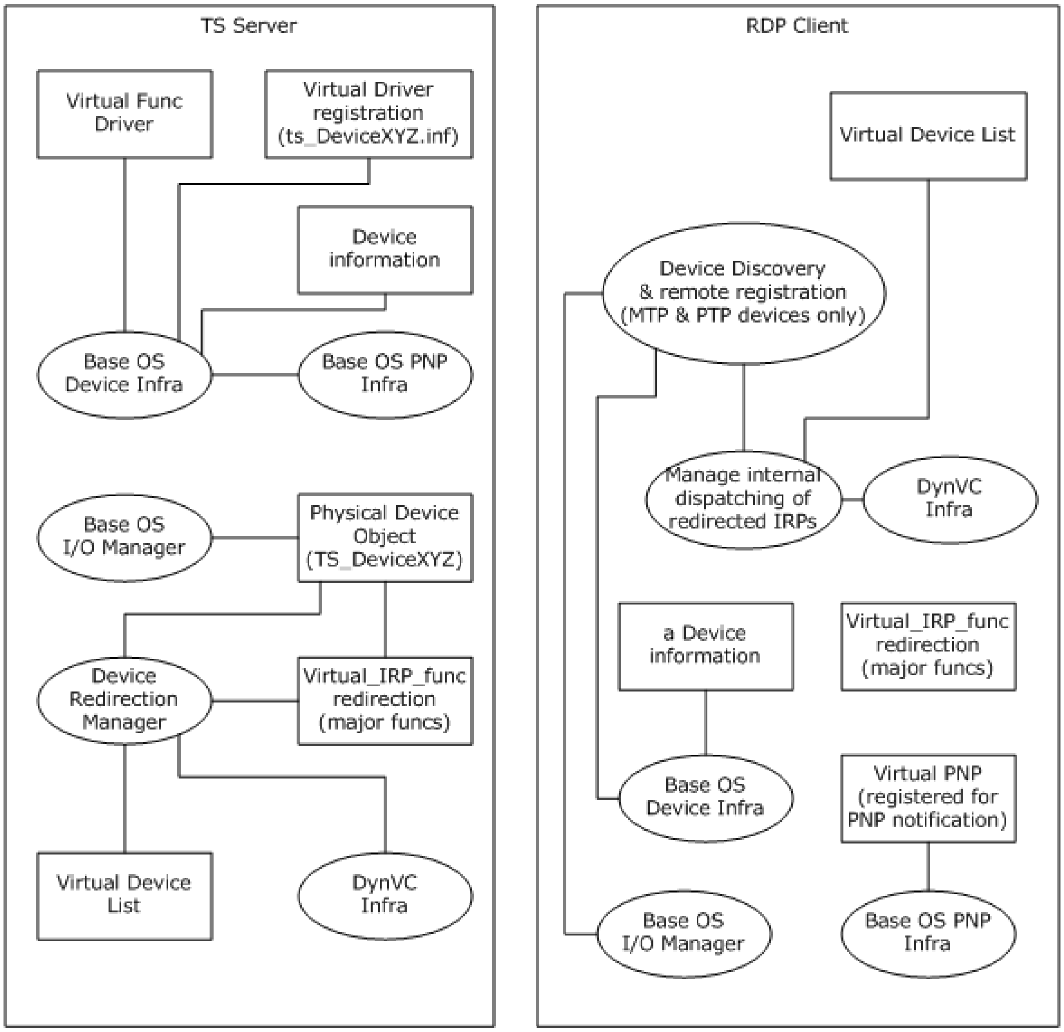
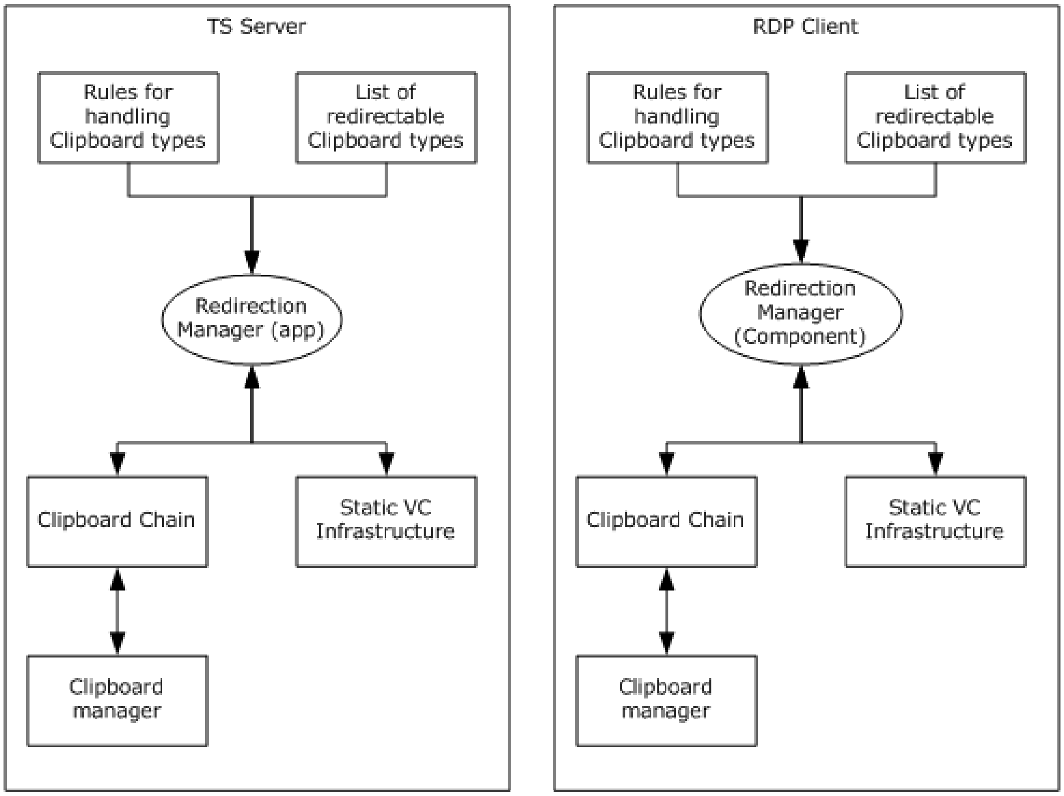

TS Server
Virtual Func Driver
Virtual Driver registration (ts_DeviceXYZ.inf)
Device information
Base OS Device Infra
Base OS PNP Infra
Base OS I/O Manager
Physical Device Object (TS_DeviceXYZ)
Device Redirection Manager
Virtual_IRP_func redirection (major funcs)
Virtual Device List
DynVC Infra
RDP Client
Virtual Device List
Device Discovery & remote registration (MTP & PTP devices only)
Manage internal dispatching of redirected IRPs
DynVC Infra
a Device information
Virtual_IRP_func redirection (major funcs)
Base OS Device Infra
Virtual PNP (registered for PNP notification)
Base OS I/O Manager
Base OS PNP Infra

Figure 18: Architecture of a Dynamic Virtual Channel supporting Plug and Play redirection

5.4.6 Clipboard Redirection

To enable redirection of **Clipboard** data, both the RDP Client and the TS Server mirror each other and pass **Clipboard** data in both directions. After a Remote Desktop Protocol (RDP) connection is established, a **Clipboard** manager application is started, registering various handlers for different types of **Clipboard** formats (such as text and bitmaps). Because some **Clipboard** formats cannot be redirected (depending on the capabilities of the RDP Client and TS Server), a list of allowed formats and rules for transport is exchanged between the RDP Client and TS Server. When the **Clipboard** manager finds a new **Clipboard** object, the manager obtains a serialized copy of the data and uses the **Clipboard** virtual channel to transport the data.



TS Server
Rules for handling Clipboard types
List of redirectable Clipboard types
Redirection Manager (app)
Clipboard Chain
Static VC Infrastructure
Clipboard manager
RDP Client
Rules for handling Clipboard types
List of redirectable Clipboard types
Redirection Manager (Component)
Clipboard Chain
Static VC Infrastructure
Clipboard manager

Figure 19: Architecture of static virtual channel supporting Clipboard redirection

5.5 Failure Scenarios

5.5.1 Connection Time Errors

When an RDP Client attempts to connect to a TS Server, the attempt can fail due to several issues described as follows:

Causes:

- DNS issues in resolving FQDN server name.
- TS Server is not listening on port 3389 at the time the connect request arrives at the TS Server.
- The server on which the TS Gateway is running is not listening on RPCE, port 3388.
- There is an Authentication failure at the TS Server or the TS Gateway.
- There are other network layer issues, such as the inability to reach a TS Server.

Consequences: Failure of the Remote Desktop connection.

Recognition: Client will not be able to connect to the Remote Desktop.

Task recovery: None.

Consequences of recovery: None.

5.5.2 Post Connection Time Errors

After a connection is initially established, the connection may fail due to one of the following:

Causes:

- Network connectivity issues or session timeouts.
- Server resource issues resulting in packet drops.

Consequences: Remote Desktop connection will be disconnected after the timeout.

Recognition: Client will lose the Remote Desktop connection.

Task recovery: None.

Consequences of recovery: None.

6 System Details

This section contains the details that complete the descriptions in earlier sections of the document. These details are needed to understand and implement this system.

6.1 Architectural Details

This section provides more information about the component architectures described in section 5 by detailing the effect of protocol messages on the states of terminal server components. To simplify the illustration of these effects, details are covered under the following scenarios. While these scenarios are not exhaustive use cases of protocol extensions, they do demonstrate the fundamentals of protocol extensions leveraging either static or dynamic virtual channels (DVC) as described in detail in [MS-RDPBCGR]. The examples are:

- Connecting from an RDP Client to a TS Server
- Connecting from an RDP Client to a TS Server through a TS Gateway
- Establishing a DVC for transporting data

The type of virtual channel shown in this example is for redirecting data from a Plug and Play device attached to the RDP Client.

- Establishing a static virtual channel for transporting data

The type of virtual channel shown in this example is for redirecting clipboard data from the TS Server to the local clipboard of the RDP Client.

- Disconnecting an RDP Client from a TS Server

Two examples are shown for disconnection: a break in the communication between the RDP Client and TS Server, and logging off from the TS Server.

6.1.1 Connecting from an RDP Client to a TS Server

This section discusses the process of establishing a secure connection between an RDP Client and a TS Server. To illustrate the connection process, the state transitions of the RDP Client are shown, as well as the TS Server activity and the message flow between the RDP Client and TS Server.

The steps for an RDP Client initiating a connection to the TS Server are specified in section 6.1.1.3. The process uses X.224 protocol data units (PDU) to establish the connection and provide secure communications, and Multipoint Communication Service (MCS) PDUs for the transfer of data between the client and server. The RDP Client proceeds to join the user channel, I/O channel, and all virtual channels by using multiple MCS Channel Join Request PDUs. Data sent between the RDP Client and the TS Server is wrapped in MCS Data PDUs as well as an X.224 Data PDU.

If standard security mechanisms and encryption are used, the subsequent RDP traffic is then encrypted and a security header is included with the data. The security header follows the X.224 and MCS Headers and indicates whether the attached data is encrypted.

The RDP Client and TS Server exchange licensing-related packets that are defined by the licensing mechanisms employed by the TS Server.

The RDP Client and TS Server send PDUs to finalize the connection details. The PDUs exchanged may be sent concurrently as long as the sequencing in either direction is maintained. After the RDP Client receives the Font Map PDU it can start sending mouse and keyboard input to the TS Server.

After the TS Server receives the Font List PDU, the TS Server can start sending graphics output to the RDP Client.

6.1.1.1 RDP Client State Model

The RDP Client state model for a basic connection scenario is illustrated in the following figure. In this scenario, an RDP Client connects to a TS Server in an intranet environment where no gateway is used. The high-level state diagram that follows shows the connection states as the RDP Client transitions from an initial state to the state of an established connection.

After the RDP Client has initialized elements in its ADM, the connection process continues as follows:

1. The RDP Client acquires the destination IP address of the TS Server using domain services.
2. The RDP Client initiates the sequence to establish an Remote Desktop Protocol (RDP) connection as described in [MS-RDPBCGR], starting with an X.224 exchange. If the connection attempt fails due to authentication issues, the flow reverts to the initial state as shown in the following figure.
3. If the X.224 exchange is successful, the RDP Client supplies capability and license information to the TS Server.
4. Once the license is validated, the user session moves to an established state. While in this state, keyboard and mouse input is sent from the RDP Client to the remote endpoint on the TS Server while graphics data is received from an established graphics channel and is sent to the display adapter of the RDP Client.

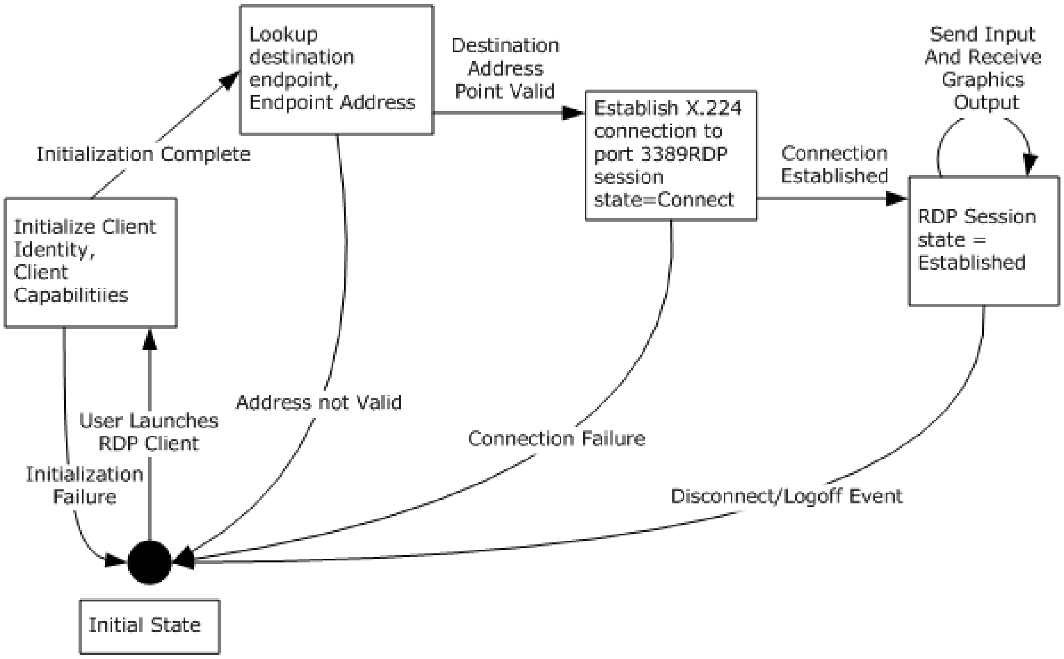

Lookup destination endpoint, Endpoint Address
Destination Address Point Valid
Send Input And Receive Graphics Output
Establish X.224 connection to port 3389RDP session state=Connect
Initialization Complete
Connection Established
RDP Session state = Established
Initialize Client Identity, Client Capabilitiies
Address not Valid
User Launches RDP Client
Connection Failure
Initialization Failure
Disconnect/Logoff Event
Initial State

Figure 20: State model of RDP Client during the connection sequence

6.1.1.2 TS Server Activity

A TS Server starts listening for an incoming connection request after internal initialization (an implementation-specific process). When an RDP Client attempts to establish a connection with a TS Server, the TS Server starts processing the request by going through a sequence of steps:

1. The TS Server passes configuration and policy data to the RDP Client.
2. The TS Server requests information about the capability of the RDP Client.
3. The TS Server queries for data from the RDP Client that will be overridden by the configuration and policy data of the TS Server.
4. The TS Server will then start a licensing sequence, requesting a license from the RDP Client and attempting to validate the license. If a new or updated license is required, the TS Server will use licensing services to obtain a new or updated license and then will send the license back to the RDP Client. If the TS Server is configured in a per-user licensing mode, the TS Server will establish a connection without validating the license provided by the RDP Client.
5. The TS Server requests the user's credentials and if the user is a domain user, the TS Server will attempt to authorize and validate the user using directory services. If the user is not allowed to log on to the TS Server, the connection request will be terminated with an appropriate error message.
6. If the user is allowed to log on, the TS Server will query for the handles to the I/O objects and will construct a terminal object. The TS Server binds the terminal object to the session object, fully establishing the connection and allowing the RDP Client to display the remote desktop or remote application.

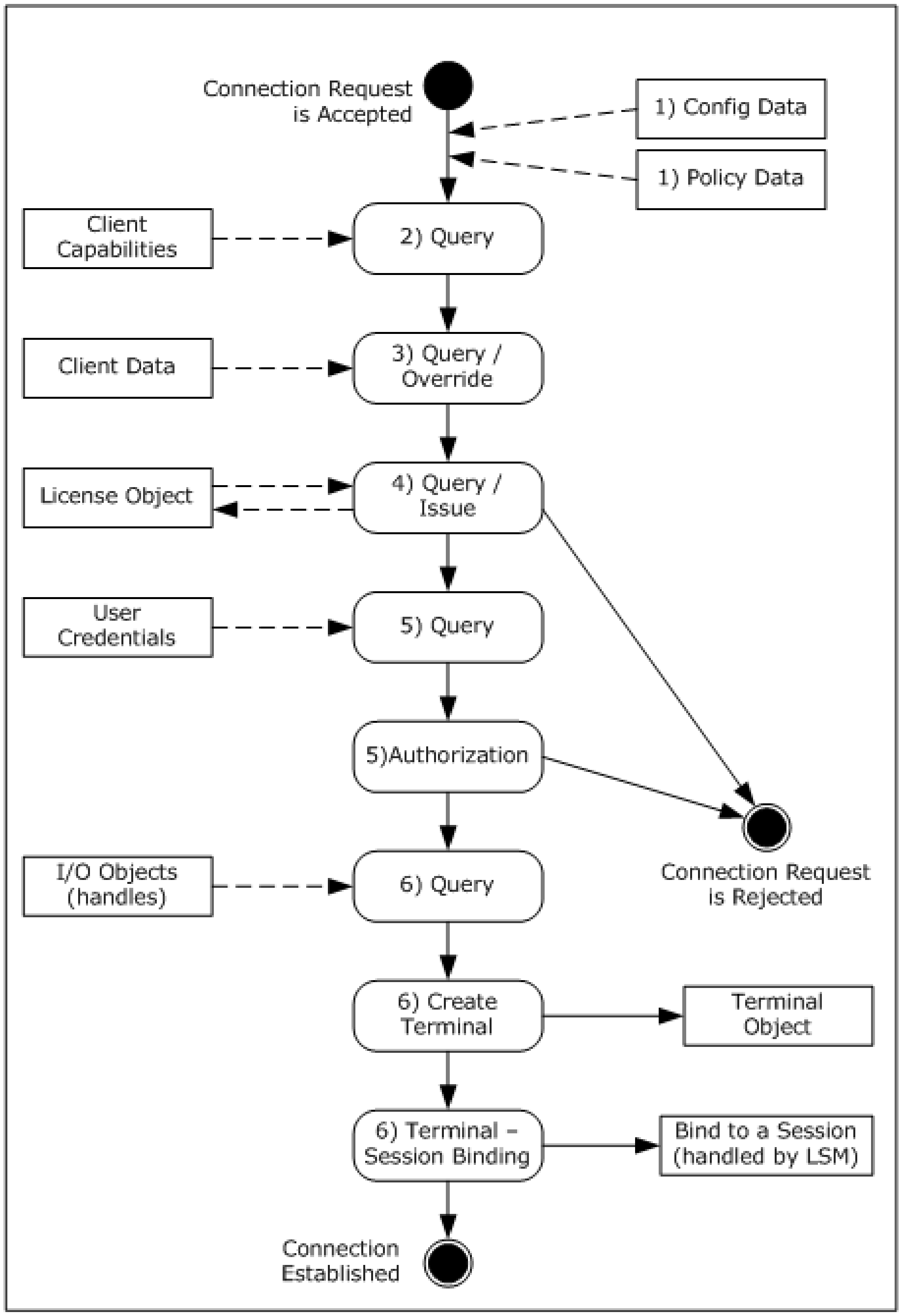


Figure 21: Activity of a TS Server when connecting to an RDP Client

6.1.1.3 Connection Sequence between an RDP Client and a TS Server

The diagram that follows illustrates one example of the messages that are exchanged between an RDP Client and a TS Server in an environment where no intermediary gateway is used.

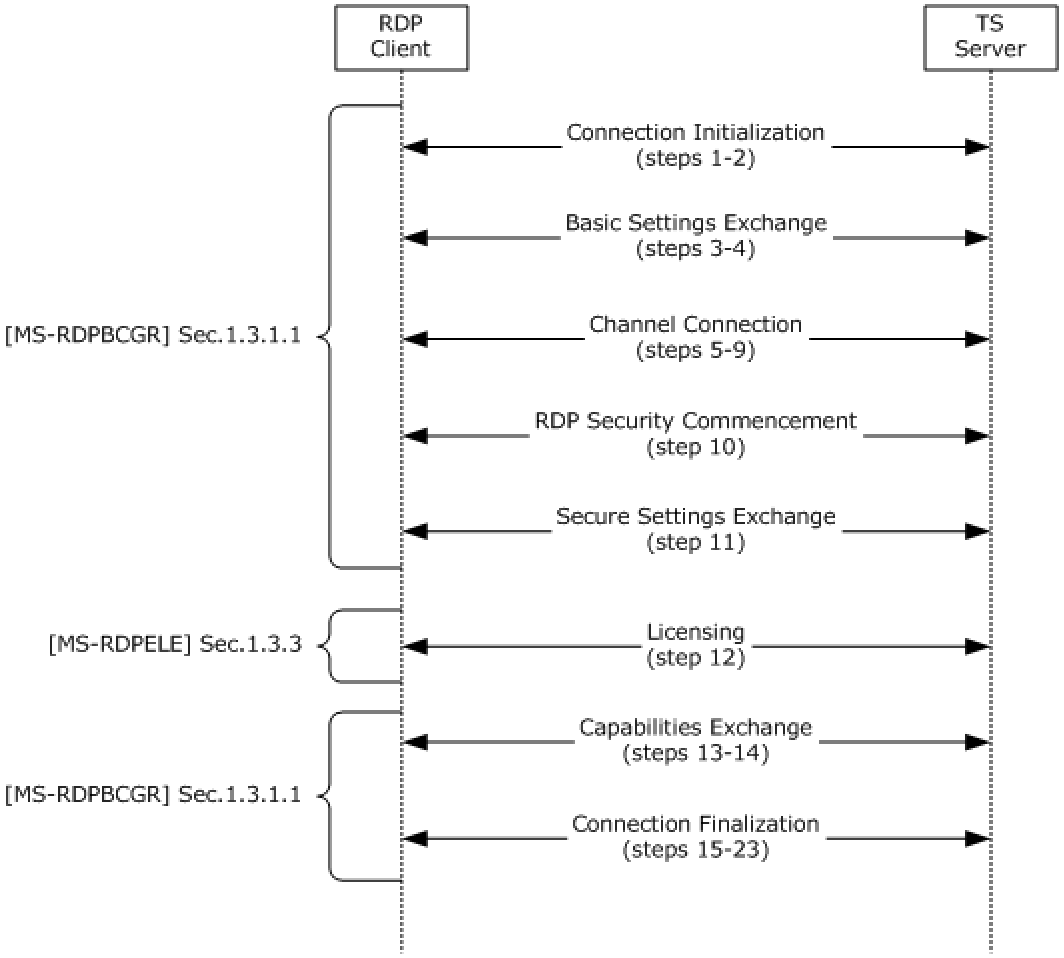

RDP Client
TS Server
Connection Initialization (steps 1-2)
Basic Settings Exchange (steps 3-4)
Channel Connection (steps 5-9)
RDP Security Commencement (step 10)
Secure Settings Exchange (step 11)
[MS-RDPBCGR] Sec.1.3.1.1
Licensing (step 12)
[MS-RDPELE] Sec.1.3.3
Capabilities Exchange (steps 13-14)
Connection Finalization (steps 15-23)
[MS-RDPBCGR] Sec.1.3.1.1

Figure 22: Sequence illustrating RDP Client connecting to a TS Server

The connection sequence is described in the following steps.

1. The RDP Client initiates a connection to the TS Server by sending an X.224 Connection Request protocol data unit (PDU), as described in [MS-RDPBCGR] section 1.3.1.1.
2. The TS Server responds with an X.224 Connection Confirm PDU.
3. The RDP Client sends a Multipoint Communication Service (MCS) Connect Initial PDU with GCC Conference Create Request.
4. The TS Server responds with an MCS Connect Response PDU with GCC Conference Create Response.
5. The RDP Client sends an MCS Erect Domain Request PDU.
6. The RDP Client sends an MCS Attach User Request PDU.
7. The server responds with an MCS Attach User Confirm PDU.
8. The RDP Client sends multiple (in this case six) MCS Channel Join Request PDUs.

9. The TS Server sends multiple (in this case six) MCS Channel Join Confirm PDUs.

10.The RDP Client sends a Security Exchange PDU.

11.The RDP Client sends a Client Info PDU.

12.The TS Server sends a License Error PDU – Valid Client. Note that there are several possible scenarios for the licensing negotiations (as described in [MS-RDPELE] section 1.3.3). For this scenario it is assumed that a valid, non-expired license exists for the client on the License Server.

13.The TS Server sends a Demand Active PDU.

14.The RDP Client responds with a Confirm Active PDU.

15.The RDP Client sends a Synchronize PDU.

16.The RDP Client sends a Control PDU – Cooperate.

17.The RDP Client sends a Control PDU – Request Control.

18.The RDP Client sends zero or more Persistent Key List PDUs. In this case zero PDUs are sent.

19.The RDP Client sends a Control PDU – Font List PDU.

20.The TS Server sends a Synchronize PDU.

21.The TS Server sends a Control PDU – Cooperate.

22.The TS Server sends a Control PDU – Granted Control.

23.The TS Server sends a Font Map PDU.

Post Connection:

After the client receives the Font Map PDU, it can start sending mouse and keyboard input to the server, and upon receipt of the Font List PDU the server can send graphics output to the client.

6.1.2 Connecting from an RDP Client to a TS Server through a TS Gateway

When an RDP Client outside a domain boundary needs to establish a connection to a TS Server across a domain boundary, the RDP Client can use a TS Gateway that is a part of the domain. The TS Gateway provides a secure method for allowing an external RDP Client to access internal resources across a firewall.

6.1.2.1 TS Gateway State Model

While communicating with the TS Gateway, the RDP Client first follows a series of steps to set up a remote procedure call (RPC) /HTTP tunnel, and then establishes an Remote Desktop Protocol (RDP) connection to the TS Server. The detailed call phases are documented in [MS-TSGU] (section 1.3.1).

The TS Gateway proxies any messages between the client and the terminal server. The data from the target server is sent by the TS Gateway server to the TS Gateway client via socket calls. In order to proxy this data from the TS Gateway server to the TS Gateway client, the Terminal Services Gateway Server Protocol utilizes RPC out pipes. The data from the RDP Client is sent to the target server via the TS Gateway Server using RPC calls. In order to send this data from the RDP Client to the TS Gateway Server, the RDP Client uses the **TsProxySendToServer** RPC call.

The state transition for the TS Gateway itself is shown in the following diagram.

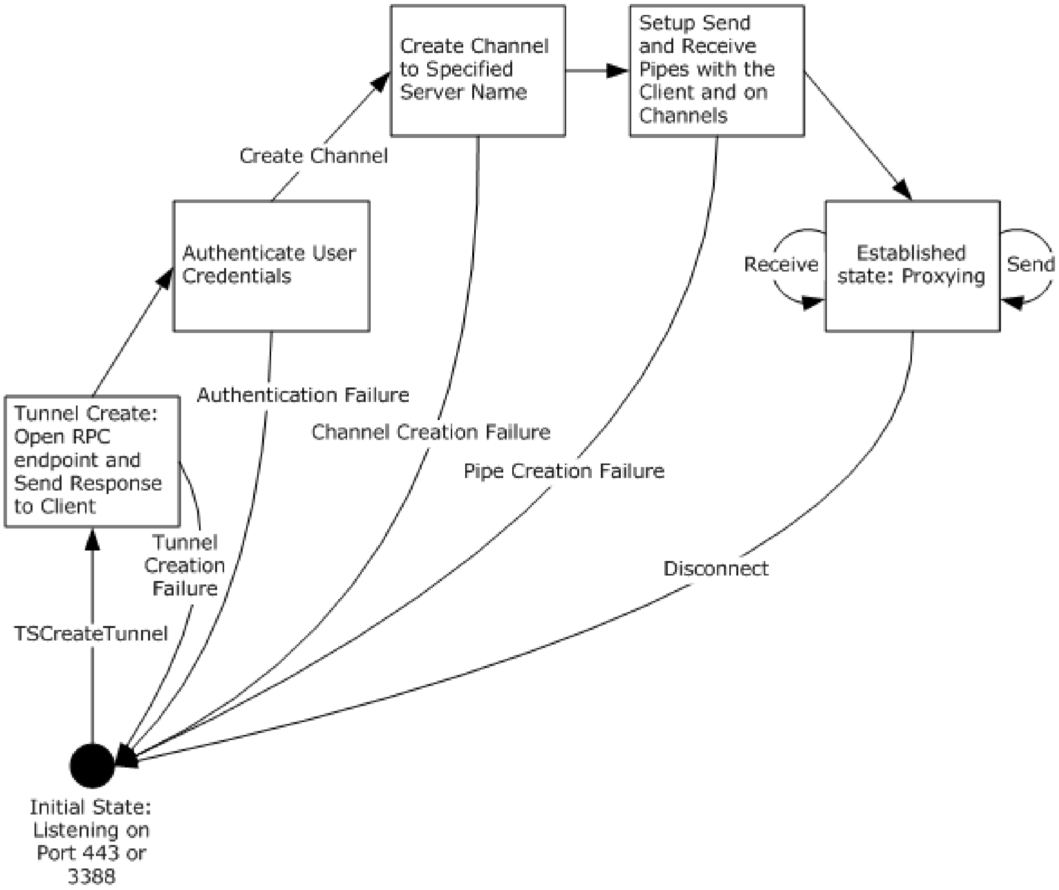

Create Channel to Specified Server Name
Setup Send and Receive Pipes with the Client and on Channels
Create Channel
Authenticate User Credentials
Receive
Established state: Proxying
Send
Authentication Failure
Channel Creation Failure
Tunnel Create: Open RPC endpoint and Send Response to Client
Pipe Creation Failure
Tunnel Creation Failure
Disconnect
TSCreateTunnel
Initial State: Listening on Port 443 or 3388

Figure 23: State model of TS Gateway during the connection sequence

6.1.2.2 Connection Sequence Using a TS Gateway

Communication messages within the Terminal Services System are illustrated in the following message flow diagram for establishing a connection between an RDP Client and TS Server using a TS Gateway. The messages are documented in more detail in the component protocol documentation [MS-RDPBCGR] and [MS-TSGU]. In the description of the connection sequence using a TS Gateway at the end of this section, steps 1-5 describe the process to create an RPC/HTTP tunnel. The data transfer phase referenced in [MS-TSGU] refers to steps 6-14.

Significant messages in the following illustration are:

1. The RDP Client initiates a connection after acquiring the remote endpoint name.
2. The RDP Client reads the configuration data elements and if the TS Gateway component is configured, begins establishing a connection to a TS Gateway. These steps are shown in the first figure below.
   - [MS-TSGU] protocol messages are used to establish a tunnel.
   - The TS Gateway establishes a channel to the TS Server.

3. After the TS Gateway proxy connection is established, the RDP Client initiates a Remote Desktop Protocol (RDP) connect request to the TS Server by tunneling the RDP requests through the established tunnel.

4. The connection sequence then follows the RDP Client to TS Server connection sequence detailed in [MS-RDPBCGR], as shown in the second figure below.

After authenticating the user, the TS Server creates a user session. The user's remote desktop is rendered in this session context and is written to the RDP stack using the established connection. The commands for rendering graphics are sent to the RDP Client via the TS Gateway and are then rendered locally by the RDP Client.

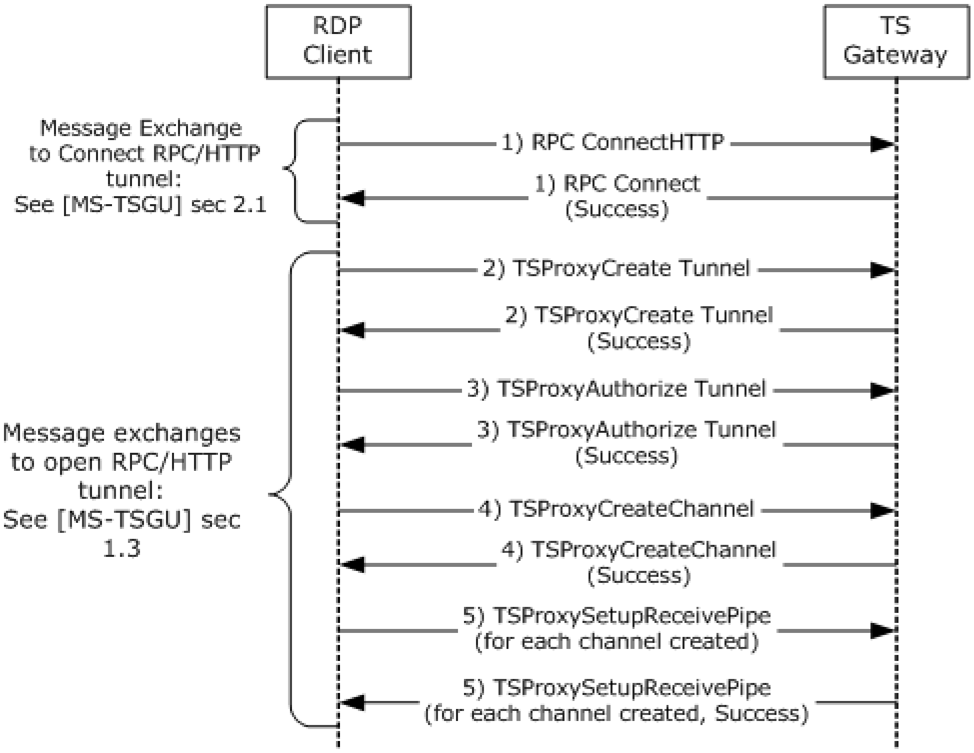

RDP Client
TS Gateway
Message Exchange to Connect RPC/HTTP tunnel: See [MS-TSGU] sec 2.1
1) RPC ConnectHTTP
1) RPC Connect (Success)
Message exchanges to open RPC/HTTP tunnel: See [MS-TSGU] sec 1.3
2) TSProxyCreate Tunnel
2) TSProxyCreate Tunnel (Success)
3) TSProxyAuthorize Tunnel
3) TSProxyAuthorize Tunnel (Success)
4) TSProxyCreateChannel
4) TSProxyCreateChannel (Success)
5) TSProxySetupReceivePipe (for each channel created)
5) TSProxySetupReceivePipe (for each channel created, Success)

Figure 24: Creating an RPC over HTTP (RPC/HTTP) tunnel

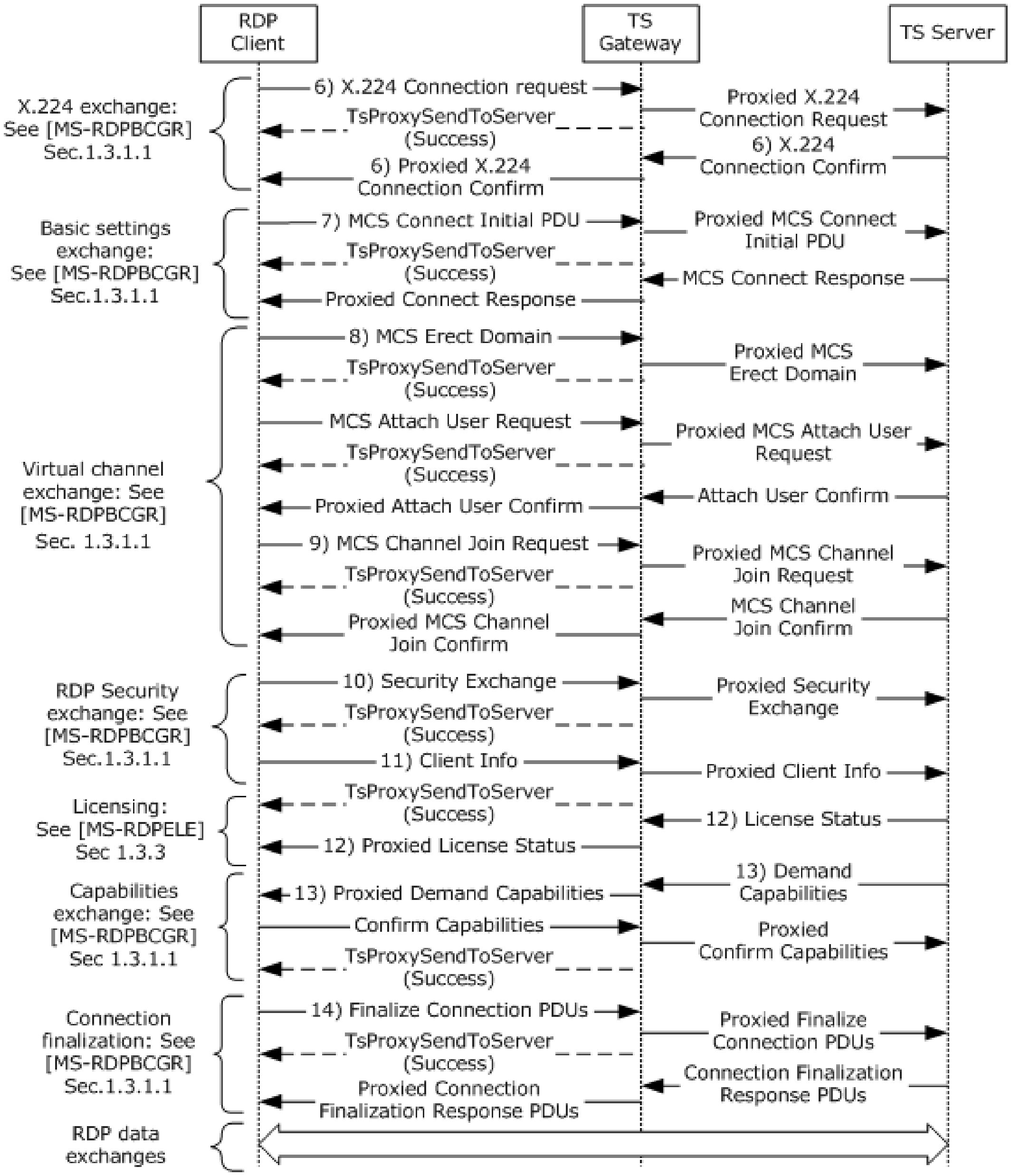

RDP Client
TS Gateway
TS Server
X.224 exchange: See [MS-RDPBCGR] Sec.1.3.1.1
6) X.224 Connection request
TsProxySendToServer (Success)
Proxied X.224 Connection Request
6) X.224 Connection Confirm
6) Proxied X.224 Connection Confirm
Basic settings exchange: See [MS-RDPBCGR] Sec.1.3.1.1
7) MCS Connect Initial PDU
Proxied MCS Connect Initial PDU
TsProxySendToServer (Success)
MCS Connect Response
Proxied Connect Response
Virtual channel exchange: See [MS-RDPBCGR] Sec. 1.3.1.1
8) MCS Erect Domain
Proxied MCS Erect Domain
TsProxySendToServer (Success)
MCS Attach User Request
Proxied MCS Attach User Request
TsProxySendToServer (Success)
Attach User Confirm
Proxied Attach User Confirm
9) MCS Channel Join Request
Proxied MCS Channel Join Request
TsProxySendToServer (Success)
MCS Channel Join Confirm
Proxied MCS Channel Join Confirm
RDP Security exchange: See [MS-RDPBCGR] Sec.1.3.1.1
10) Security Exchange
Proxied Security Exchange
TsProxySendToServer (Success)
11) Client Info
Proxied Client Info
Licensing: See [MS-RDPELE] Sec 1.3.3
TsProxySendToServer (Success)
12) License Status
12) Proxied License Status
Capabilities exchange: See [MS-RDPBCGR] Sec 1.3.1.1
13) Demand Capabilities
13) Proxied Demand Capabilities
Confirm Capabilities
Proxied Confirm Capabilities
TsProxySendToServer (Success)
Connection finalization: See [MS-RDPBCGR] Sec.1.3.1.1
14) Finalize Connection PDUs
Proxied Finalize Connection PDUs
TsProxySendToServer (Success)
Connection Finalization Response PDUs
Proxied Connection Finalization Response PDUs
RDP data exchanges

Figure 25: Creating an RDP connection

A description of the connection sequence using a TS Gateway is as follows:

1. The RDP Client sends an RPC Connect HTTP Request to the TS Gateway. The TS Gateway responds with a RPC Connect Response. This sequence is described in [MS-TSGU] (section 2.1).

2. The RDP Client sends a TSProxyCreateTunnel Request to the TS Gateway to request that a tunnel be created. The TS Gateway responds with a TSProxyCreateTunnel Response. This sequence is described in [MS-TSGU] (section 1.3).

3. The RDP Client sends a TSProxyAuthorizeTunnel Request to the TS Gateway to authorize the tunnel from the previous step. The TS Gateway responds with a TSProxyAuthorizeTunnel Response. This sequence is described in [MS-TSGU] (section 1.3).

4. The RDP Client sends a TSProxyCreateChannel Request to the TS Gateway to create a channel. The TS Gateway responds with a TSProxyCreateChannel Response. This sequence is described in [MS-TSGU] (section 1.3).

5. For each channel, the RDP Client sends a TSProxySetupReceivePipe Request to the TS Gateway to establish a pipe for data transfer. The TS Gateway responds with a TSProxySetupReceivePipe Response. This sequence is described in [MS-TSGU] (section 1.3).

6. By proxy, The RDP Client initiates a connection to the TS Server by sending an X.224 Connection Request protocol data unit (PDU), as described in [MS-RDPBCGR] (section 1.3.1.1). The server responds with an X.224 Connection Confirm PDU. All subsequent data sent between the RDP Client and TS Server is wrapped in an X.224 Data PDU.

7. By proxy, basic settings are exchanged between the RDP Client and TS Server using the Multipoint Communication Service (MCS) Connect Initial and MCS Connect Response PDUs, as described in [MS-RDPBCGR] (section 1.3.1.1).

8. By proxy, the RDP Client sends an MCS Erect Domain Request PDU, followed by an MCS Attach User Request PDU to attach the primary user identity to the MCS domain, as described in [MS-RDPBCGR] (section 1.3.1.1). The server responds with an MCS Attach User Confirm PDU containing the user channel ID.

9. By proxy, the RDP Client proceeds to join the user channel, I/O channel, and all virtual channels by using multiple MCS Channel Join Request PDUs, as described in [MS-RDPBCGR] (section 1.3.1.1). The TS Server confirms each channel with an MCS Channel Join Confirm PDU. All subsequent data sent from the RDP Client to the TS Server is wrapped in an MCS Send Data Request PDU, while data sent from the TS Server to the RDP Client is wrapped in an MCS Send Data Indication PDU. This is in addition to the data being wrapped by an X.224 Data PDU.

10. If Standard RDP security mechanisms and encryption are being used, which they are for this example, the RDP Client sends a Security Exchange PDU containing an encrypted 32-byte random number to the TS Server, by proxy, as described in [MS-RDPBCGR] (section 1.3.1.1). All subsequent RDP traffic is then encrypted and a security header is included with the data if encryption is in force. The security header follows the X.224 and MCS Headers and indicates whether the attached data is encrypted.

11. By proxy, the RDP Client sends secure client data (such as username, password, and auto-reconnect cookie) to the server using the Client Info PDU, as described in [MS-RDPBCGR] (section 1.3.1.1).

12. By proxy, the RDP Client and TS Server exchange licensing-related packets that are defined by the licensing mechanisms employed by the TS Server, as described in [MS-RDPBCGR] (section 1.3.1.1). Different licensing scenarios are possible and are covered in [MS-RDPELE] (section 1.3.3). For this scenario it is assumed that a valid, non-expired, license exists for the client on the License Server.

13. By proxy, the TS Server sends the set of capabilities it supports to the RDP Client in a Demand Active PDU, as described in [MS-RDPBCGR] (section 1.3.1.1). The RDP Client responds with its capabilities by sending a Confirm Active PDU.

14. By proxy, the RDP Client and TS Server send PDUs to finalize the connection details, as described in [MS-RDPBCGR] (section 1.3.1.1). The PDUs exchanged may be sent concurrently as long as the sequencing in either direction is maintained. After the RDP Client receives the Font Map PDU

it can start sending mouse and keyboard input to the TS Server. After the TS Server receives the Font List PDU, the TS Server can start sending graphics output to the RDP Client.

6.1.3 Establishing a Dynamic Virtual Channel for Plug and Play Device Redirection

One common scenario for using a dynamic virtual channel (DVC) involves redirecting data from a Plug and Play device that is connected to an RDP Client after a user session has been established. After the Remote Desktop Protocol (RDP) connection has been made with the DVC, components of the RDP Client and TS Server find the device and redirect the data as follows:

1. The RDP Client starts listening on a known static virtual channel that is being used for DVC communications.

2. A Virtual Plug and Play module on the RDP Client registers itself to be the handler of redirected Plug and Play communication. The module also registers itself to handle events related to Plug and Play devices located on the RDP Client.

3. On the TS Server, the registration of a device that can be redirected has already been established. When an application attempts to communicate with a redirected device, the I/O manager on the TS Server creates a link between the application on the TS Server and the Virtual Function Driver on the TS Server.

4. The Virtual Function Driver on the TS Server uses a DVC to the Virtual Function Driver on the RDP Client.

5. The application on the TS Server sends and receives I/O to and from the Virtual Function Driver representing the device.

6. The Virtual Function Driver routes the I/O to the Virtual Function Driver on the RDP Client.

7. The Virtual Plug and Play module on the RDP Client routes the I/O to the real Function Driver registered on the RDP Client.

The following figures illustrate the process of establishing a DVC that is then used for redirecting data from a Plug and Play device located on the RDP Client.

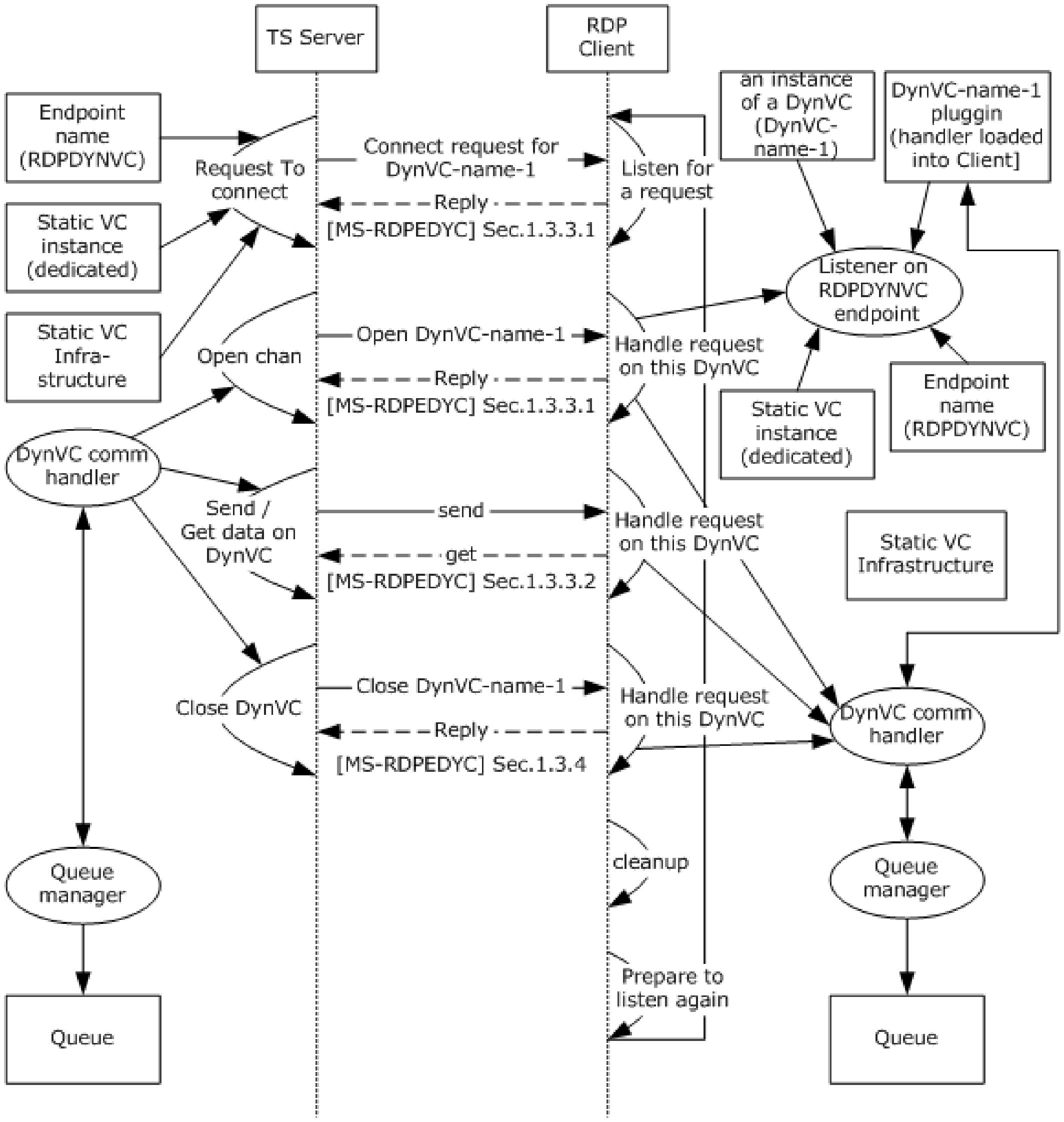


Figure 26: Sequence establishing a dynamic virtual channel for Plug and Play redirection

The following steps describe the sequence of establishing a DVC.

1. The TS Server sends a Capabilities protocol data unit (PDU) that indicates the maximum supported version level as well as any capability information that is relevant for the supported version.
2. The RDP Client responds with a Capabilities Response PDU that states the maximum version level that it supports.
3. The TS Server and RDP Client exchange Create Request and Create Response PDUs to establish the DVC for Plug and Play redirection.
4. The RDP Client sends data from the Plug and Play device to the TS Server (as described in the next portion of this section).

5. To close the channel, the TS Server sends a Close Request PDU for the DVC.

6. The RDP Client responds with a Close Response PDU.

7. After the RDP connection sequence has begun and a DVC has been established, data from a Plug and Play device is redirected as shown in the following diagram.

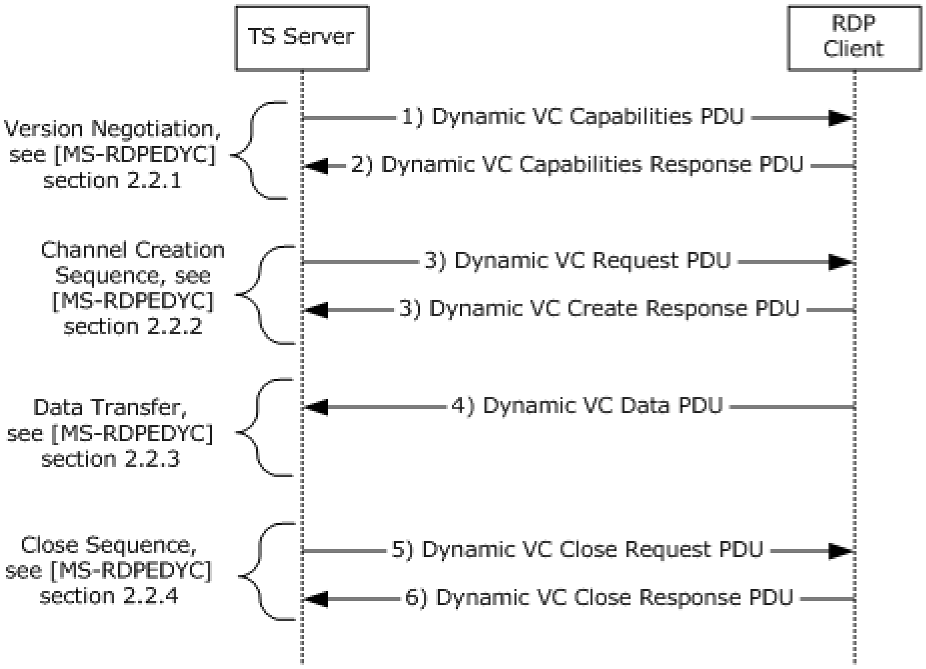

TS Server
RDP Client
Version Negotiation, see [MS-RDPEDYC] section 2.2.1
1) Dynamic VC Capabilities PDU
2) Dynamic VC Capabilities Response PDU
Channel Creation Sequence, see [MS-RDPEDYC] section 2.2.2
3) Dynamic VC Request PDU
3) Dynamic VC Create Response PDU
Data Transfer, see [MS-RDPEDYC] section 2.2.3
4) Dynamic VC Data PDU
Close Sequence, see [MS-RDPEDYC] section 2.2.4
5) Dynamic VC Close Request PDU
6) Dynamic VC Close Response PDU

Figure 27: Detailed sequence establishing dynamic virtual channel

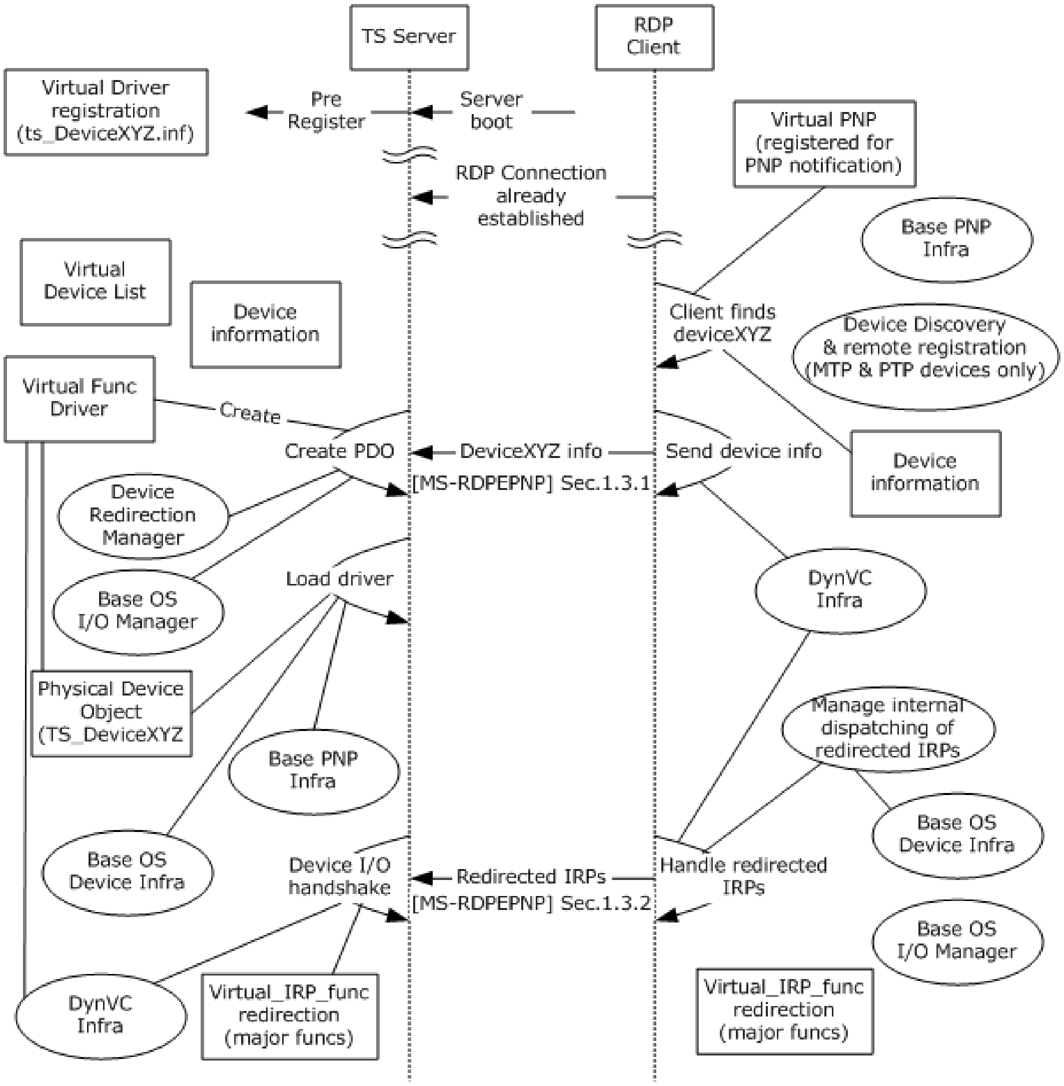


Figure 28: Sequence for redirecting Plug and Play device data

The following steps describe the sequence of adding a Plug and Play device to the RDP Client, transferring data to and from the device, and removing the device from the RDP Client.

1. The TS Server exposes its capabilities and version information to the RDP Client after the RDP connection is initialized.
2. The RDP Client responds by sending its capabilities and version information.
3. When the Plug and Play device is physically added to the RDP Client, the TS Server sends an Authenticated Client message to the RDP Client.
4. The RDP Client responds with a Client Device Additions message.
5. The TS Server sends a Capabilities Request message to the RDP Client.
6. The RDP Client responds by exposing its capabilities.

7. Depending on the Plug and Play device added, device IO messages such as File Create/Read/Write/IO Control Request messages can be sent from the TS Server to the RDP Client.

8. The RDP Client responds with File Create/Read/Write/IO Control Response messages.

9. The RDP Client can send Custom Event messages to the TS Server with details.

10. Depending on the RDP Client message, the TS Server will send a Specific IO Cancel Request message.

11. When the Plug and Play device is physically removed from the RDP Client, the TS Server will send the Authenticated Client message to the RDP Client.

12. The RDP Client responds by sending a Client Device Removal message.

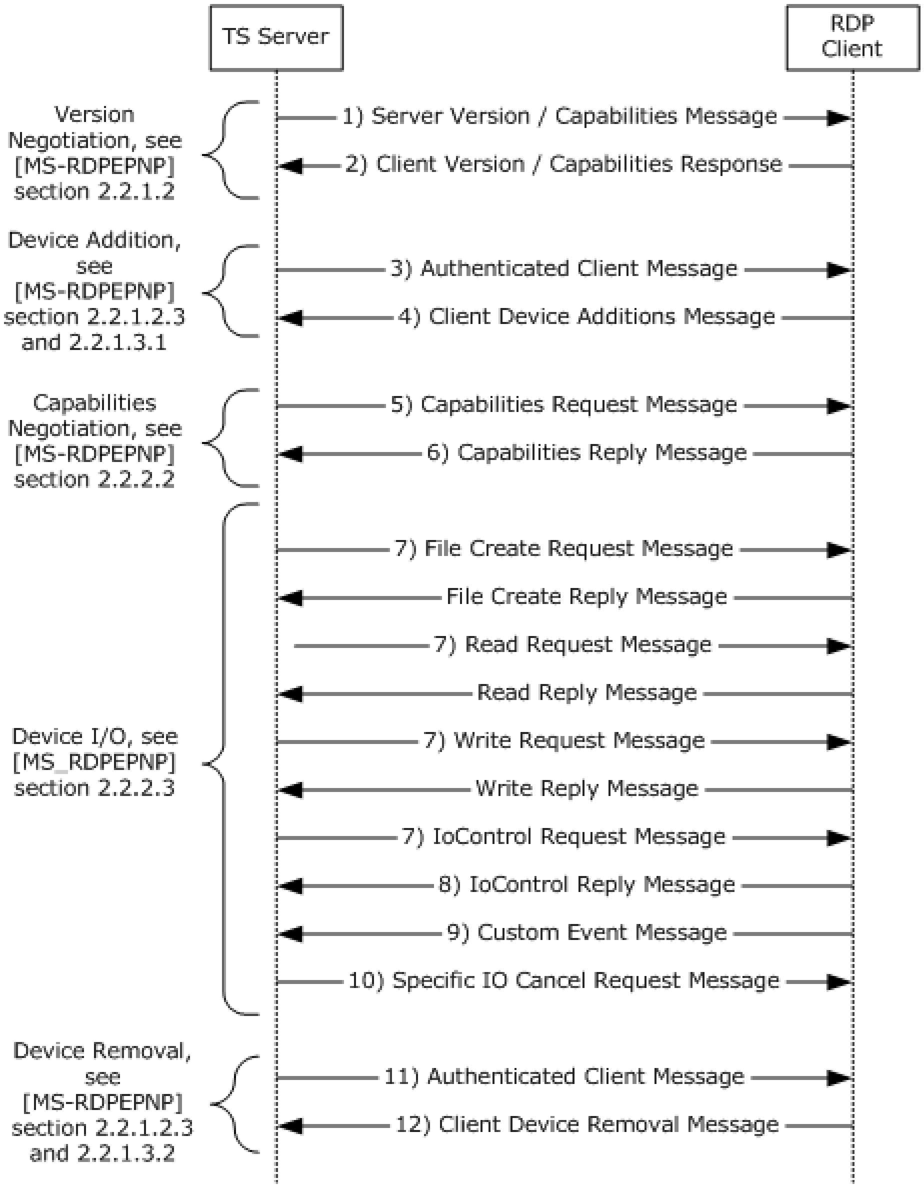

TS Server
RDP Client
Version Negotiation, see [MS-RDPEPNP] section 2.2.1.2
1) Server Version / Capabilities Message
2) Client Version / Capabilities Response
Device Addition, see [MS-RDPEPNP] section 2.2.1.2.3 and 2.2.1.3.1
3) Authenticated Client Message
4) Client Device Additions Message
Capabilities Negotiation, see [MS-RDPEPNP] section 2.2.2.2
5) Capabilities Request Message
6) Capabilities Reply Message
Device I/O, see [MS_RDPEPNP] section 2.2.2.3
7) File Create Request Message
File Create Reply Message
7) Read Request Message
Read Reply Message
7) Write Request Message
Write Reply Message
7) IoControl Request Message
8) IoControl Reply Message
9) Custom Event Message
10) Specific IO Cancel Request Message
Device Removal, see [MS-RDPEPNP] section 2.2.1.2.3 and 2.2.1.3.2
11) Authenticated Client Message
12) Client Device Removal Message

Figure 29: Detailed sequence for redirecting data from a Plug and Play device

6.1.4 Redirecting Clipboard Data

A static virtual channel is used to redirect Clipboard data. This channel is established during the initial RDP connection sequence.

The protocol described in [MS-RDPECLIP] is not dependent on a user being logged on to a TS Server.<2> After a user logs on, a logon notification message MAY be sent by a terminal server to the RDP Client, depending on the implementation, but such a message is not required and does not determine whether the clipboard redirection succeeds or fails.

The following diagrams illustrate how a Clipboard channel is initialized and subsequently used to transfer data.

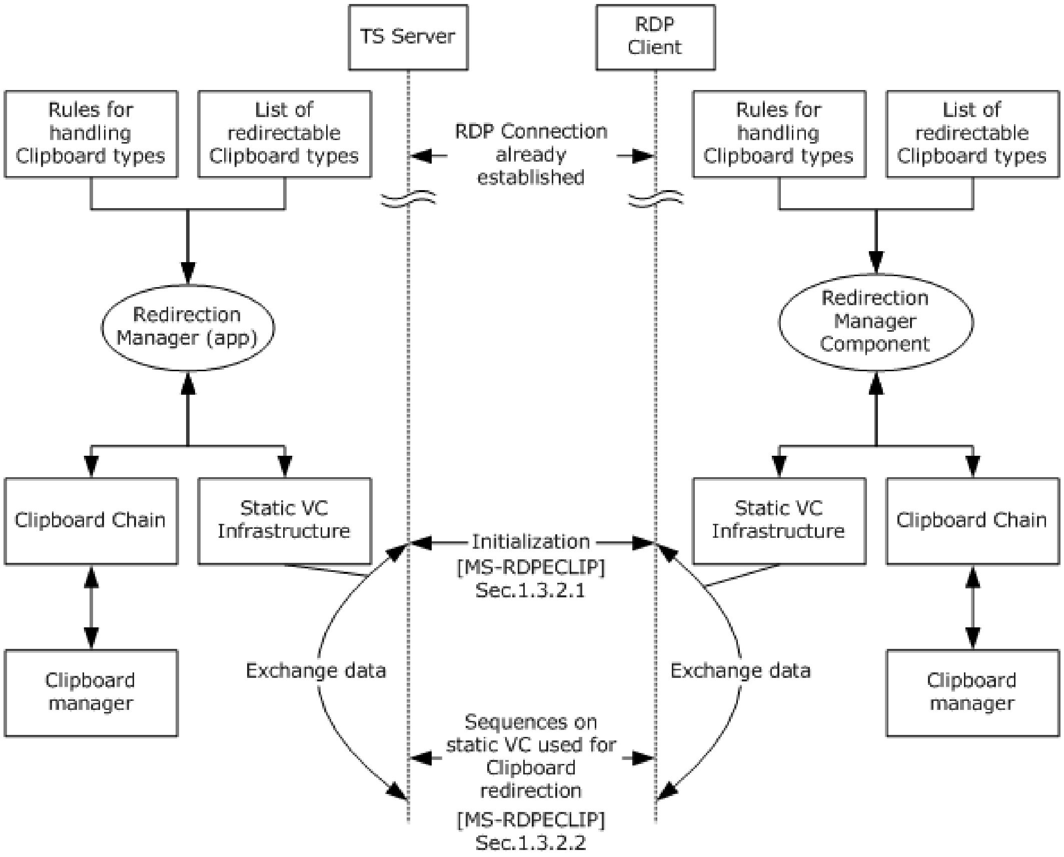

TS Server
RDP Client
Rules for handling Clipboard types
List of redirectable Clipboard types
RDP Connection already established
Rules for handling Clipboard types
List of redirectable Clipboard types
Redirection Manager (app)
Redirection Manager Component
Clipboard Chain
Static VC Infrastructure
Initialization [MS-RDPECLIP] Sec.1.3.2.1
Static VC Infrastructure
Clipboard Chain
Clipboard manager
Exchange data
Exchange data
Clipboard manager
Sequences on static VC used for Clipboard redirection [MS-RDPECLIP] Sec.1.3.2.2

Figure 30: Establishing a static virtual channel for redirecting Clipboard data

A more detailed view of the Clipboard initialization sequence and the Clipboard data transfer sequence is illustrated in the following diagrams:

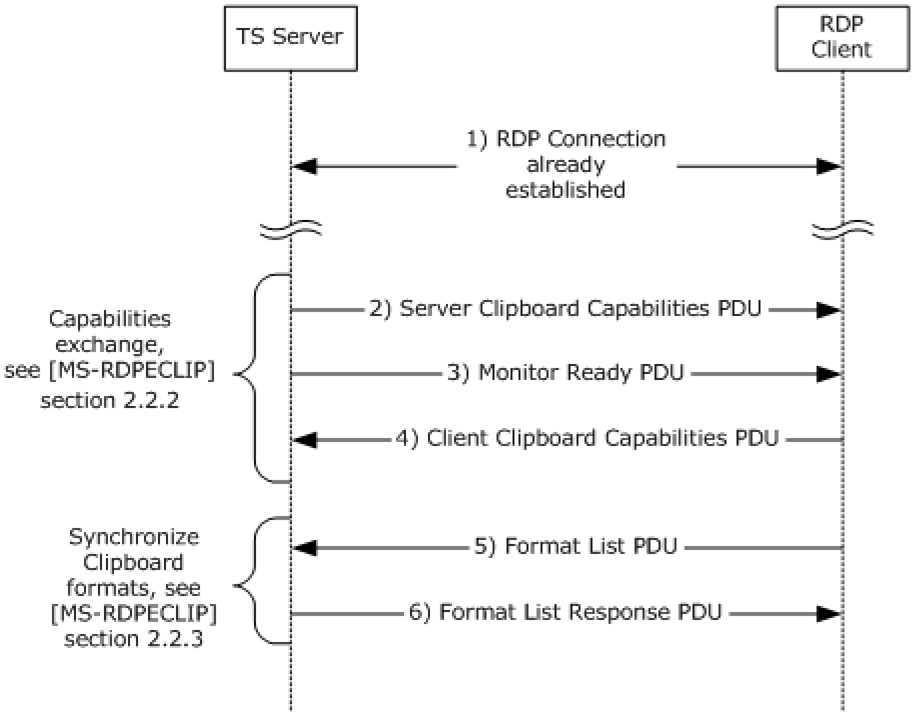

TS Server
RDP Client
1) RDP Connection already established
Capabilities exchange, see [MS-RDPECLIP] section 2.2.2
2) Server Clipboard Capabilities PDU
3) Monitor Ready PDU
4) Client Clipboard Capabilities PDU
Synchronize Clipboard formats, see [MS-RDPECLIP] section 2.2.3
5) Format List PDU
6) Format List Response PDU

Figure 31: Clipboard initialization sequence

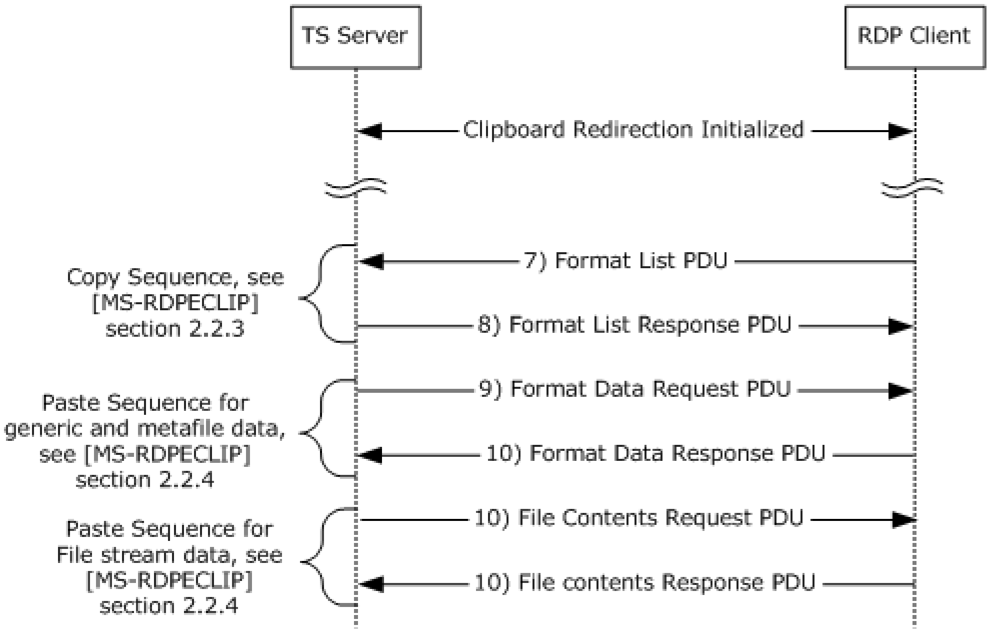

TS Server
RDP Client
Clipboard Redirection Initialized
Copy Sequence, see [MS-RDPECLIP] section 2.2.3
7) Format List PDU
8) Format List Response PDU
Paste Sequence for generic and metafile data, see [MS-RDPECLIP] section 2.2.4
9) Format Data Request PDU
10) Format Data Response PDU
Paste Sequence for File stream data, see [MS-RDPECLIP] section 2.2.4
10) File Contents Request PDU
10) File contents Response PDU

Figure 32: Data transfer sequence

The steps of the Clipboard initialization sequence and data transfer sequence are as follows:

1. The RDP Client establishes an Remote Desktop Protocol (RDP) connection with the TS Server.

2. The TS Server sends a Clipboard Capabilities protocol data unit (PDU) to the RDP Client to advertise the capabilities that it supports.

3. The TS Server sends a Monitor Ready PDU to the RDP Client.

4. Upon receiving the Monitor Ready PDU, the RDP Client transmits its capabilities to the TS Server by using a Clipboard Capabilities PDU.

5. The final stage of the Initialization Sequence involves synchronizing the Clipboard formats on the TS Server Clipboard and the RDP Client. This is accomplished by effectively mimicking a copy operation on the RDP Client by forcing it to send a Format List PDU to the TS Server.

6. The TS Server responds with a Format List Response PDU.

7. To copy the data, the RDP Client sends a Format List PDU to the TS Server.

8. The TS Server responds with a Format List Response PDU.

9. The TS Server sends a Format Data Request PDU in case of generic or metafile data or File Contents Request PDU in case of file stream data.

10. The client sends Format Data Response PDU/File Contents Response PDU (with the data) accordingly in response to the request sent by the TS Server.

6.1.5 Disconnection Sequence

There are two ways an RDP Client may leave a connection with a TS Server:

- The RDP Client is disconnected from a TS Server.

- The user of the RDP Client logs off from the TS Server.

6.1.5.1 RDP Client Disconnects from TS Server

An RDP Client may be disconnected from the TS Server due to the RDP Client being turned off because of network problems, or for other reasons. In these cases, the user session established on the TS Server remains active for a certain amount of time, depending on how the TS Server is configured. If the RDP Client reconnects to the TS Server within this timeframe, the RDP Client will be reconnected to the same user session.

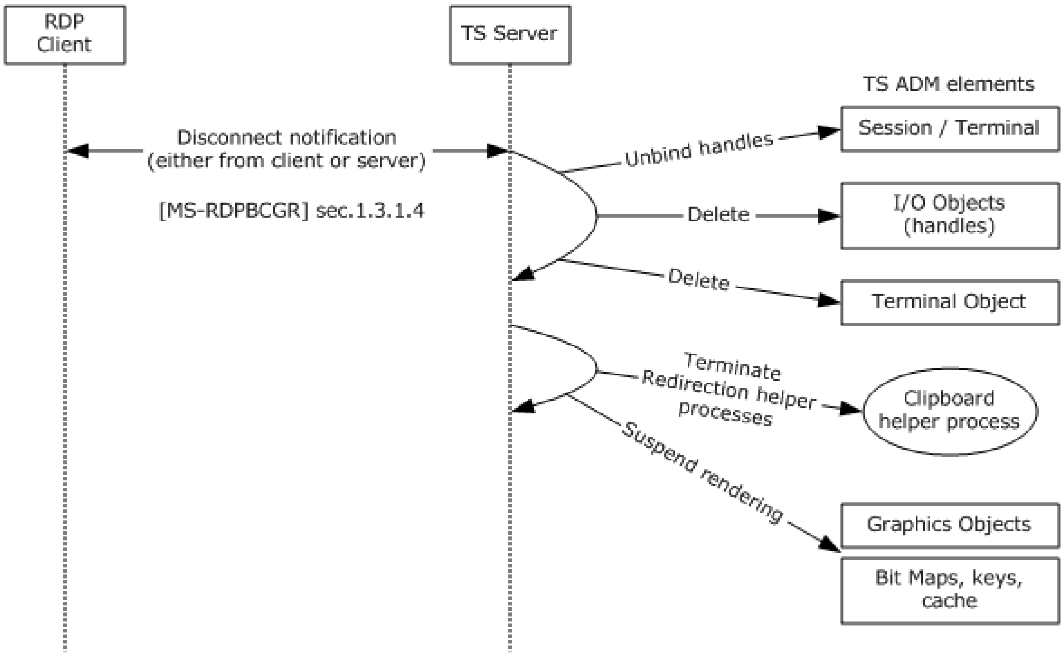


Figure 33: Sequence of RDP Client disconnecting from a TS Server

6.1.5.2 RDP Client Logoff from TS Server

If the user of an RDP Client logs off from the TS Server, or an administrator forces the user logoff using an administrative tool, the user session on the TS Server is closed and the TS Server performs accompanying clean-up work.

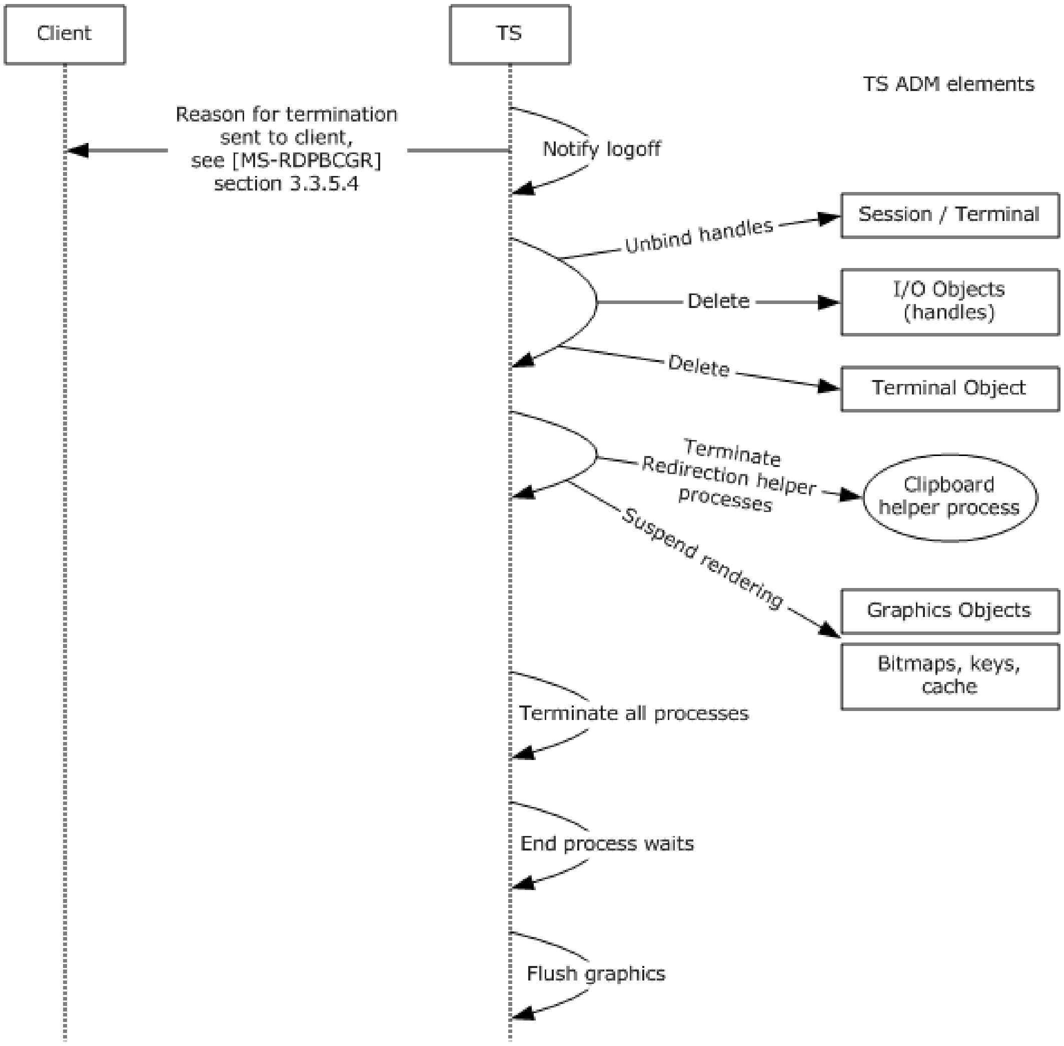


Figure 34: Sequence of RDP Client Logging off from TS Server

6.2 Communication Details

The system does not define any communication constraints or additional message types beyond those described in the specifications of the protocols supported by the system, as listed in section 2.2.

6.3 Transport Requirements

RDP Client to TS Server communication requires Transmission Control Protocol (TCP) transport over either an IPV4 or IPV6 network.

6.4 Timers

The Session Disconnect timer is the primary timer that is used by the TS Server to disconnect an idle session. The TS Server disconnects the session on a timeout event by initiating a protocol disconnect event.

6.5 Non-Timer Events

The Terminal Services System depends on platform facilities for creating a remote session for the remote user. As such, there are several error events possible during this operation arising from several subsystems such as authentication, authorization, group policy, and licensing. The Terminal Services System handles these errors and appropriately transmits the errors using the protocol described in [MS-RDPBCGR].

To enable connection shutdowns by administrators, the Terminal Services System provides management functions to enumerate sessions and terminate all or specific sessions.

6.6 Initialization and Re-initialization Procedures

The initialization of the RDP Client, TS Server, and TS Gateway are described in this section. Other components of the Terminal Services System are initialized as described in the component protocol documentation. Virtual channels, including dynamic virtual channels (DVC), are initialized as a part of the connection sequence between the RDP Client and the TS Server.

6.6.1 RDP Client

The RDP Client can be initialized in multiple ways. The data elements such as user identity and user credentials can be either preconfigured in a configuration file (Remote Desktop Protocol (RDP) file) or can be acquired interactively through the RDP user interface. This is true for the client properties such as color bitmap, resolution, authentication behavior, and device redirection options.

6.6.2 Terminal Server

The TS Server has a specific user interface through which TS Server properties can be configured. The data elements are initialized primarily through registry, group policies, and user interface when the TS Server is started. The user profile services, session broker services, TS Gateway, and license services data elements are initialized with appropriate names or IP addresses via the user interface or a configuration file.

6.6.3 TS Gateway

The TS Gateway component data elements such as certificates configured are initialized when the TS Gateway services are started. The configuration elements such as policy data and TS Server name are read through a persistent store located either locally or remotely. Once the TS Gateway service is initialized, the TS Gateway listens on port 443 or port 3388 for incoming remote procedure call (RPC) requests.

6.7 Status and Error Returns

The Terminal Services System does not define any error handling requirements beyond those described in the specifications of the protocols supported by the system, as listed in section 2.2.

Various kinds of error conditions may impact one or more protocols that comprise the system. Such error conditions and the resulting protocol semantics are described in section 3 of component protocol documentation.

7 Security

This section documents system-wide security issues that are not otherwise described in the Technical Documents (TDs) for the Member Protocols. It does not duplicate what is already in the Member Protocol TDs unless there is some unique aspect that applies to the system as a whole.

The Terminal Services System includes security features for creating secure end-to-end connections between mutually authenticated RDP Clients and TS Servers. The Terminal Services System also includes security features to ensure the privacy and integrity of data exchanged using encryption. The security mechanisms that provide secure end-to-end communication for basic connections and virtual channels are described in [MS-RDPBCGR] (section 5) . In addition, there are general implementation-specific restrictions relating to some of the components of the Terminal Services System as detailed in the following sections.

7.1 RDP Client

The RDP Client implementation will ensure that user credentials are not locally stored in clear text form. In the Microsoft implementation of the Terminal Server System, the Windows security system is leveraged when handling user passwords.

7.2 TS Server

The configuration data elements of the TS Server that are persisted either in database or registry hives require administrator privileges to be accessible. In addition, management objects that interact with remote sessions are protected and require administrator privileges or local system access privileges to be accessible.

Administrators interacting with TS Servers use the Terminal Services Terminal Server Runtime Interface protocol, described in [MS-TSTS].

7.3 TS Gateway

The data elements that configure the TS Gateway, such as to policies regarding remote access and device redirection, are limited to administrator access.

8 Appendix A: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include released service packs:

- Microsoft Windows NT® Server 4.0 operating system, Terminal Server Edition
- Microsoft Windows NT® Server 4.0 operating system, Terminal Server Edition with Service Pack 4
- Microsoft Windows® 2000 Server operating system
- Windows® XP operating system
- Windows Server® 2003 operating system
- Windows Vista® operating system
- Windows Vista® operating system Ultimate
- Windows Vista® operating system Business
- Windows Server® 2008 operating system
- Windows® 7 operating system
- Windows Server® 2008 R2 operating system

Exceptions, if any, are noted below. If a service pack or Quick Fix Engineering (QFE) number appears with the product version, behavior changed in that service pack or QFE. The new behavior also applies to subsequent service packs of the product unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms SHOULD or SHOULD NOT implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that the product does not follow the prescription.

<1> Section 5.1.2: A trusted authorization system could be, for example, a logon component in Windows.

<2> Section 6.1.4: In the Windows implementation of this scenario, the clipboard redirection will not happen until the user logs on.

Release: Friday, February 4, 2011

9 Change Tracking

This section identifies changes that were made to the [MS-TSSO] protocol document between the January 2011 and February 2011 releases. Changes are classified as New, Major, Minor, Editorial, or No change.

The revision class **New** means that a new document is being released.

The revision class **Major** means that the technical content in the document was significantly revised. Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements or functionality.
- An extensive rewrite, addition, or deletion of major portions of content.
- The removal of a document from the documentation set.
- Changes made for template compliance.

The revision class **Minor** means that the meaning of the technical content was clarified. Minor changes do not affect protocol interoperability or implementation. Examples of minor changes are updates to clarify ambiguity at the sentence, paragraph, or table level.

The revision class **Editorial** means that the language and formatting in the technical content was changed. Editorial changes apply to grammatical, formatting, and style issues.

The revision class **No change** means that no new technical or language changes were introduced. The technical content of the document is identical to the last released version, but minor editorial and formatting changes, as well as updates to the header and footer information, and to the revision summary, may have been made.

Major and minor changes can be described further using the following change types:

- New content added.
- Content updated.
- Content removed.
- New product behavior note added.
- Product behavior note updated.
- Product behavior note removed.
- New protocol syntax added.
- Protocol syntax updated.
- Protocol syntax removed.
- New content added due to protocol revision.
- Content updated due to protocol revision.
- Content removed due to protocol revision.
- New protocol syntax added due to protocol revision.

- Protocol syntax updated due to protocol revision.
- Protocol syntax removed due to protocol revision.
- New content added for template compliance.
- Content updated for template compliance.
- Content removed for template compliance.
- Obsolete document removed.

Editorial changes are always classified with the change type **Editorially updated.**

Some important terms used in the change type descriptions are defined as follows:

- **Protocol syntax** refers to data elements (such as packets, structures, enumerations, and methods) as well as interfaces.
- **Protocol revision** refers to changes made to a protocol that affect the bits that are sent over the wire.

The changes made to this document are listed in the following table. For more information, please contact protocol@microsoft.com.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
3.3.4.1 Establish a Connection to a TS Server in an Intranet Environment — RDP Client	59864 Changed "must" to "MUST" in the preconditions description.	N	Content updated.
3.3.4.2 Establish a Connection to a VM Host in an Intranet Environment — RDP Client	59866 Changed "must" to "MUST" in the preconditions description.	N	Content updated.
3.3.4.3 Establish a Connection Using a TS Gateway – RDP Client	59878 Changed "must" to "MUST" in the preconditions description.	N	Content updated.
4.1 System Environment	59880 Changed "must" to "MUST" in the opening paragraph.	N	Content updated.
5.3.2 Member Protocol Groups	59884 Changed "an RDP Client must use a VPN" to "an RDP Client MUST use a VPN".	N	Content updated.
5.3.2 Member Protocol Groups	59890 Changed "a static virtual channel supporting dynamic virtual channels (DVC) must first be established" to "a static virtual channel supporting	N	Content updated.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
	dynamic virtual channels (DVC) MUST first be established".		

10 Index