[MS-RA]:
Remote Assistance Protocol Specification

Revision Summary

Date	Revision History	Revision Class	Comments
02/22/2007	0.01		MCPP Milestone 3 Initial Availability
06/01/2007	1.0	Major	Updated and revised the technical content.
07/03/2007	1.0.1	Editorial	Revised and edited the technical content.
07/20/2007	1.1	Minor	Updated the technical content.
08/10/2007	1.2	Minor	Updated the technical content.

Date	Revision History	Revision Class	Comments
09/28/2007	1.3	Minor	Updated the technical content.
10/23/2007	1.3.1	Editorial	Revised and edited the technical content.
11/30/2007	1.4	Minor	Updated the technical content.
01/25/2008	1.4.1	Editorial	Revised and edited the technical content.

Table of Contents

1 Introduction

This document describes the Remote Assistance Protocol available in Windows Vista and Windows XP. This protocol is used after a **remote assistance connection** is established between two computers. The protocol used to establish the remote assistance connection is specified in the Microsoft Remote Assistance Initiation Protocol. After the remote assistance connection is established, this protocol is used to support communications and control between the two computers. The functions supported by the Remote Assistance Protocol are session initialization, file transfer, chat (text message exchange), share control, and Voice-over-IP (**VoIP**) control.

1.1 Glossary

Expert: The side of a **remote assistance connection** that can view the remote screen of the other computer to provide help.

Novice: The side of a **remote assistance connection** that shares its screen with the other computer to receive help.

RDP: See **Remote Desktop Protocol**.

Remote Assistance (RA): A feature of the operating system that allows screen, keyboard, and mouse sharing so that a computer user can be assisted by a remote helper.

Remote Assistance Connection: A communication framework that is established between two computers to facilitate **remote assistance (RA)**.

Remote Assistance Session: A **remote assistance connection** that is initialized and allows the **expert** to view the screen of the **novice**.

Remote Desktop Protocol: Enables the exchange of client and server settings; also enables negotiation of common settings to use for the duration of the connection, so that input, graphics, and other data can be exchanged and processed between client and server.

Terminal Services (TS): The capability to host multiple, simultaneous client sessions on Windows servers. Remote users establish a session on a machine, log in, and run applications on a server. The server transmits the graphical user interface (GUI) of the program to the client. The client then returns keyboard and mouse clicks to be processed by the server.

Virtual Channel: A static transport used for lossless communication between two computers over a main data connection, in 1600-byte chunks, that is created once a **remote assistance connection** is established. A virtual channel allows data messages and control messages to be exchanged between the two computers. The virtual channel that is used by the Remote Assistance Protocol is specified in [MS-RDPBCGR].

VoIP: Voice over Internet Protocol.

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as specified in [RFC2119]. All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

1.2 References

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-RAI] Microsoft Corporation, "Remote Assistance Initiation Protocol Specification", June 2007.

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

[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

1.2.2 Informative References

[MSDN-RTC] Microsoft Corporation, "RTC Overview", October 2004, http://msdn2.microsoft.com/en-us/library/ms775938.aspx

1.3 Protocol Overview (Synopsis)

The Remote Assistance Protocol is used after a remote assistance connection is established to facilitate different capabilities used during the connection. This protocol supports five capabilities: session initialization, file transfer, chat, share control, and VoIP control.

The Remote Assistance Protocol uses **virtual channels** (a **Terminal Services (TS)** feature) as its underlying transport to accomplish these capabilities. There are four virtual channels used by the Remote Assistance Protocol:

- The session initialization virtual channel is created when the remote assistance connection is made, and persists through the duration of the remote assistance connection. This channel is used to do initial setup and configuration of the remote assistance connection.
- The file transfer virtual channel is created on demand to transfer file data.
- The chat virtual channel is created when the remote assistance connection is first established, and persists through the duration of the remote assistance connection.
- The last virtual channel is used for share control, and to initialize VoIP and file transfer.

1.3.1 Session Initialization

The session initialization capability supported by the Remote Assistance Protocol allows control messages to be exchanged between the **novice** and the **expert**. This exchange has to be completed successfully for the **remote assistance session** to be established.

Once the remote assistance session is established, the expert can view the novice's screen, and other **Remote Assistance (RA)** capabilities can be initiated.

1.3.2 File Transfer

The file transfer capability supported by the Remote Assistance Protocol enables files to be copied from one computer to another. Both computers have to be in a remote assistance connection to transfer files. The Remote Assistance Protocol supports the transfer of one file at a time. File transfers can occur in either direction (from expert to novice or from novice to expert). File transfers are originated by the sender (expert or novice) side and the receiver accepts the file to complete the file transfer.

A file transfer virtual channel is created dynamically to transfer the file. Once the virtual channel is established, control messages and data messages are sent through the virtual channel to complete the transfer. The data messages contain the data that is in the file, and the control messages synchronize the file transfer between the two computers and confirm successful transfer.

1.3.3 Session Control

The session control capability supported by the Remote Assistance Protocol is used to control and synchronize the state of the remote assistance connection between two computers. When a remote assistance connection is first established, it is in a view-only state, and the expert can view the screen of the novice's computer. To change state to the share-control state, the expert must request for control, and control sharing must be granted by the novice. The session control capability is used to enable state change and synchronize the remote assistance connection state between the two machines.

A session control virtual channel is created when the remote assistance connection is established, and it is used to exchange control messages between the two computers. The session control virtual channel persists for the duration of the remote assistance connection. Only control messages are sent through the session control virtual channel.

1.3.4 Chat

The chat capability supported by the Remote Assistance Protocol allows the exchange of text messages between two machines that are in a remote assistance connection. Both computers have to be in a remote assistance connection to exchange chat messages. A chat virtual channel is created when the remote assistance connection is established and persists through the duration of the remote assistance connection.

Once the chat virtual channel is created, text messages can be transported in a duplex manner between the two computers. All the messages that are sent through a chat virtual channel are text messages. The chat virtual channel does not have any control messages.

1.3.5 VoIP Control

The VoIP (Voice over Internet Protocol) capability is used to control the audio communications between two computers in a remote assistance connection. The VoIP control virtual channel is created dynamically if VoIP is attempted during a remote assistance connection. The virtual channel is used to negotiate and control the VoIP connection. Voice data flow is an independent peer-to-peer communication and does not use the established virtual channel.

Remote assistance (RA) uses the Real-Time Communications (RTC) client to enable peer-to-peer audio communication using VoIP.<1> For more information, see [MSDN-RTC].

1.4 Relationship to Other Protocols

The Remote Assistance Protocol assumes that a remote assistance connection between two computers has been established using the Remote Assistance Initiation Protocol and the Remote Desktop Protocol: Basic Connectivity and Graphics Remoting Specification.

The Remote Assistance Protocol assumes that an underlying protocol, specifically the Remote Desktop Protocol: Basic Connectivity and Graphics Remoting Specification [MS-RDPBCGR], will be available to transport the protocol messages.

No other protocol is dependent on the Remote Assistance Protocol.

1.5 Prerequisites/Preconditions

The Remote Assistance Protocol assumes that a remote assistance connection between two computers has been established.

1.6 Applicability Statement

The Remote Assistance Protocol can only be used between two computers after a remote assistance connection has been established. This protocol is used to initialize the remote assistance (RA) session and accomplish file transfer, session control, chat, and VoIP control.

1.7 Versioning and Capability Negotiation

The Remote Assistance Protocol uses the REMOTEDESKTOP_CTL_VISTA_EXPERT_PACKET message to announce that the expert is running Windows Vista.<2>

1.8 Vendor-Extensible Fields

There are no vendor-extensible fields in the Remote Assistance Protocol.

1.9 Standards Assignments

The Remote Assistance Protocol does not use any standards assignments.

2 Messages

The following sections specify how the Remote Assistance Protocol messages are encapsulated on the wire and common data types.

2.1 Transport

When the remote assistance connection is started, it MUST create three virtual channels:

- The session initialization virtual channel MUST be named "RC_CTL", and is used to initialize the remote assistance connection.
- A second virtual channel named "70" MUST be created, and is used to exchange chat messages.
- A third virtual channel named "71" MUST be created, and is used for share control and for the initialization of file transfer and VoIP control.

These three virtual channels MUST persist for the duration of the remote assistance session.

A separate virtual channel for file transfer SHOULD be created dynamically when needed, and SHOULD be deleted after the file transfer is completed.

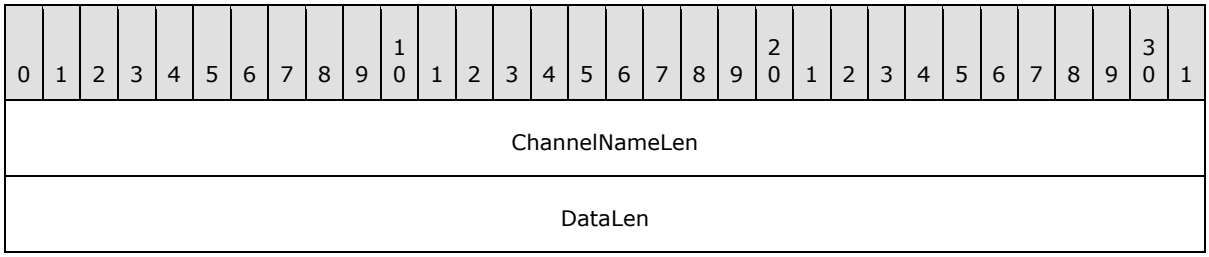
2.2 Message Syntax

2.2.1 Session Initialization Messages

All of these messages MUST be sent and received over the session initialization (RC_CTL) virtual channel.

2.2.1.1 REMOTEDESKTOP_CHANNELBUFHEADER

The REMOTEDESKTOP_CHANNELBUFHEADER packet provides information about the size of a remote assistance (RA) virtual channel packet. This data structure is at the top of all channel packets. Channel name and message data immediately follow.

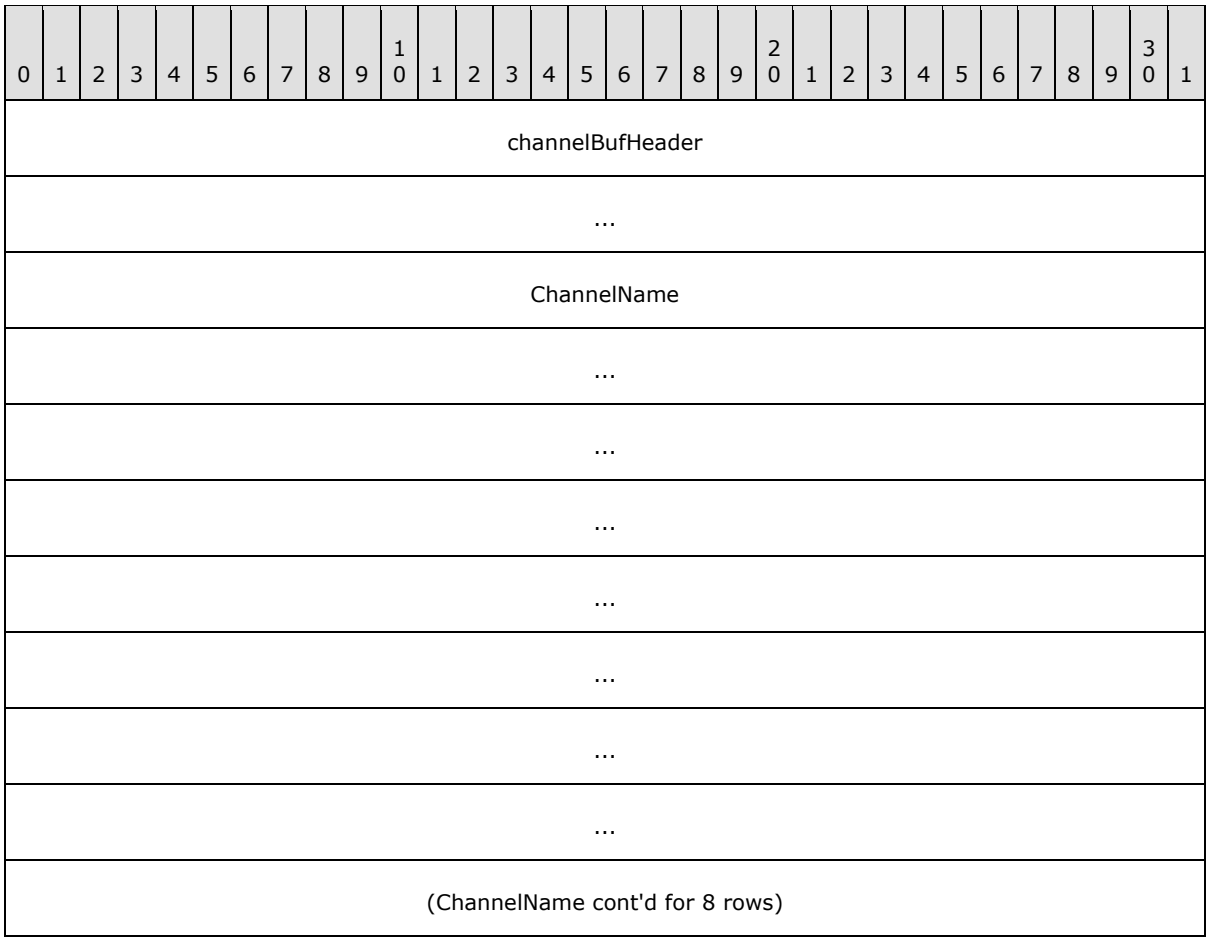


ChannelNameLen (4 bytes): Length of the virtual channel name in **bytes**. This is a DWORD.

DataLen (4 bytes): Length in **bytes** of the packet data. This is a **DWORD**.

2.2.1.2 REMOTEDESKTOP_CTL_PACKETHEADER

The REMOTEDESKTOP_CTL_PACKETHEADER is the control message packet header.



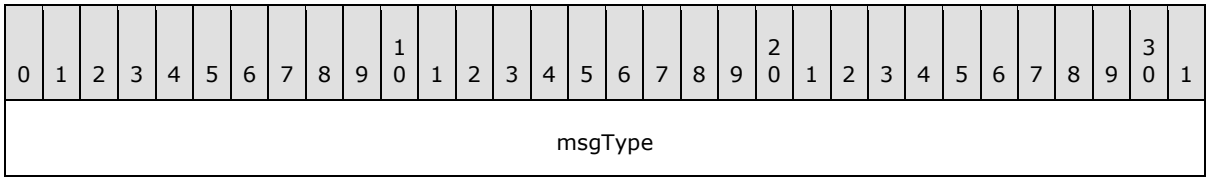
channelBufHeader (8 bytes): This is of type REMOTEDESKTOP_CHANNELBUFHEADER.

ChannelName (64 bytes): Null-terminated Unicode name of the virtual channel for which the packet is intended. ChannelName length can vary with the maximum length being 64 bytes. The **ChannelName** field is 64 bytes long.

Data immediately follows the **ChannelName** field and is transferred as a Unicode string.

2.2.1.3 REMOTEDESKTOP_CTL_BUFHEADER

The REMOTEDESKTOP_CTL_BUFHEADER describes the type of a remote assistance channel message.



msgType (4 bytes): A DWORD, in which the value is one of the following:

Name	Value
REMOTEDESKTOP_CTL_REMOTE_CONTROL_DESKTOP	1
REMOTEDESKTOP_CTL_RESULT	2
REMOTEDESKTOP_CTL_AUTHENTICATE	3
REMOTEDESKTOP_CTL_SERVER_ANNOUNCE	4
REMOTEDESKTOP_CTL_DISCONNECT	5
REMOTEDESKTOP_CTL_VERSIONINFO	6
REMOTEDESKTOP_CTL_ISCONNECTED	7
REMOTEDESKTOP_CTL_VERIFY_PASSWORD	8
REMOTEDESKTOP_CTL_VISTA_EXPERT	9
REMOTEDESKTOP_CTL_RANOVICE_NAME	10
REMOTEDESKTOP_CTL_RAEXPERT_NAME	11

2.2.1.4 REMOTEDESKTOP_CTL_AUTHENTICATE_PACKET

The REMOTEDESKTOP_CTL_AUTHENTICATE_PACKET is the expert authentication request packet. The expert sends this packet that includes the remote assistance connection string to the novice requesting authentication.<3>

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
packetHeader																															
...																															
...																															
...																															
...																															
...																															
...																															
...																															
(packetHeader cont'd for 10 rows)																															
bufHeader																															
raConnectionString (variable)																															
...																															
expertBlob (variable)																															
...																															

packetHeader (72 bytes): The REMOTEDESKTOP_CTL_PACKETHEADER part of the packet. The virtual channel name SHOULD be set to "RC_CTL".

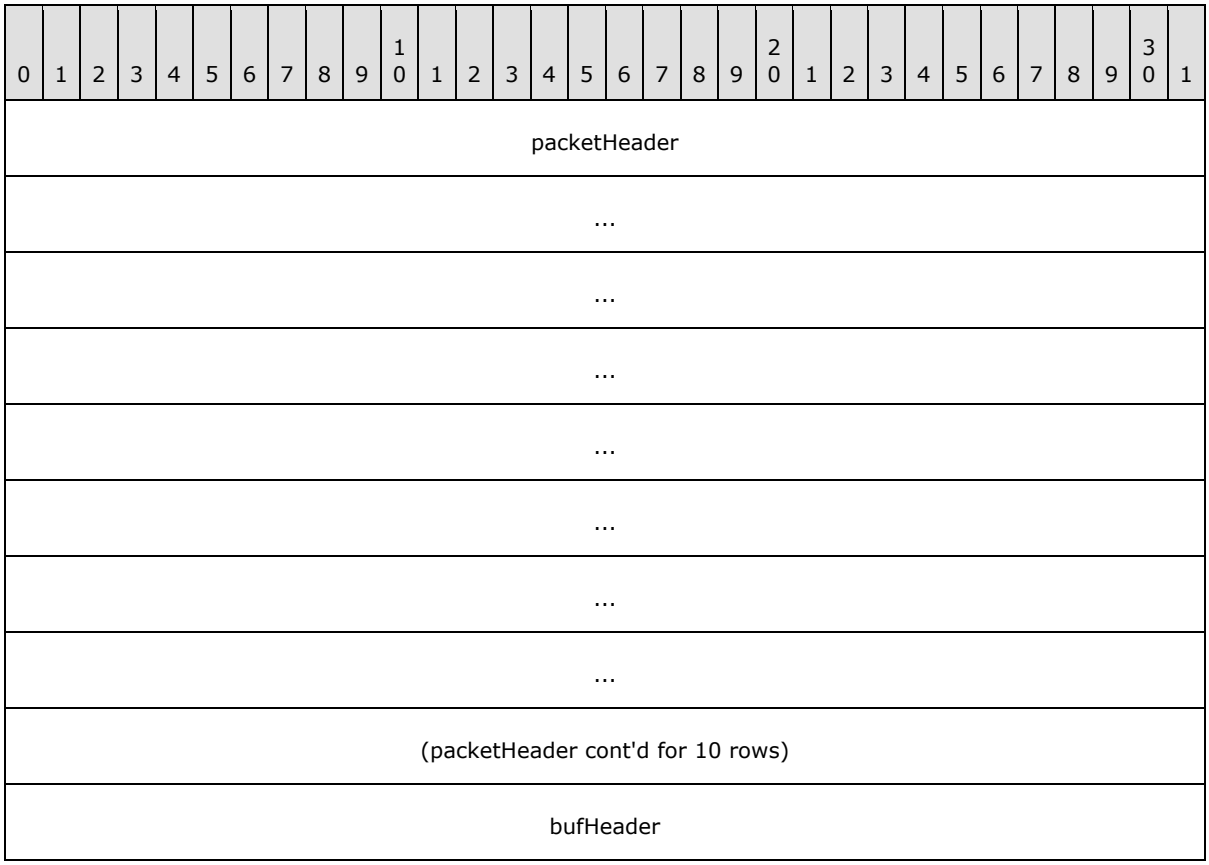
bufHeader (4 bytes): The REMOTEDESKTOP_CTL_BUFHEADER part of the packet. The packet type SHOULD be set to REMOTEDESKTOP_CTL_AUTHENTICATE.

raConnectionString (variable): A variable-length string containing the remote assistance connection string, as specified in [MS-RAI].

expertBlob (variable): A variable-length, semicolon-delimited, Unicode-based set of PropertyName, PropertyValue pairs. Each pair is also prefixed with the length of the characters in the pair, including the equal (=) sign. For example, if PropertyName is "NAME", and PropertyValue is "John", the value of **expertBlob** is "9;NAME=John". This is a mechanism to provide more information about the expert that is connecting to the novice.<4>

2.2.1.5 REMOTEDESKTOP_CTL_DISCONNECT_PACKET

The REMOTEDESKTOP_CTL_DISCONNECT_PACKET indicates that the sender has disconnected.



packetHeader (72 bytes): The REMOTEDESKTOP_CTL_PACKETHEADER part of the packet. The virtual channel name SHOULD be set to "RC_CTL".

bufHeader (4 bytes): The REMOTEDESKTOP_CTL_BUFHEADER part of the packet. The packet type SHOULD be set to REMOTEDESKTOP_CTL_DISCONNECT.

It is possible that a disconnected client or server will not send this packet. The REMOTEDESKTOP_CTL_ISCONNECTED_PACKET packet should be used to track connection state. There is no other additional data.

2.2.1.6 REMOTEDESKTOP_CTL_ISCONNECTED_PACKET

The REMOTEDESKTOP_CTL_ISCONNECTED_PACKET indicates that the sender is present.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
packetHeader																															
...																															
...																															
...																															
...																															
...																															
...																															
...																															
(packetHeader cont'd for 10 rows)																															
bufHeader																															

packetHeader (72 bytes): The REMOTEDESKTOP_CTL_PACKETHEADER part of the packet. The virtual channel name SHOULD be set to "RC_CTL".

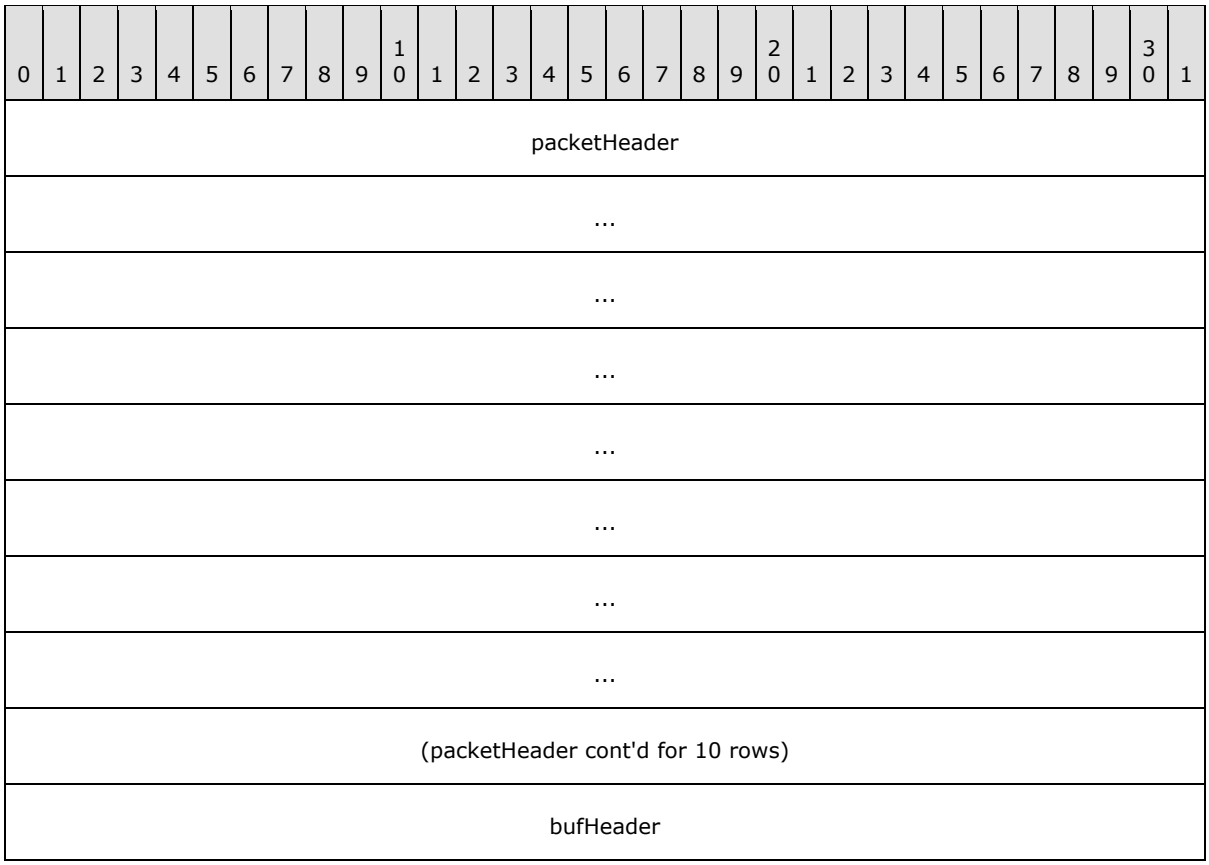
bufHeader (4 bytes): The REMOTEDESKTOP_CTL_BUFHEADER part of the packet. The packet type SHOULD be set to REMOTEDESKTOP_CTL_ISCONNECTED.

There is no additional data beyond this.

Both novice and expert send this packet to each other every 30 seconds over an idle connection. If they do not receive a REMOTEDESKTOP_CTL_ISCONNECTED_PACKET packet in a 30-second idle interval, they disconnect.<5>

2.2.1.7 REMOTEDESKTOP_CTL_SERVERANNOUNCE_PACKET

The REMOTEDESKTOP_CTL_SERVERANNOUNCE_PACKET is sent from the server to the client to begin the remote assistance connection sequence.



packetHeader (72 bytes): The REMOTEDESKTOP_CTL_PACKETHEADER part of the packet. The virtual channel name SHOULD be set to "RC_CTL".

bufHeader (4 bytes): The REMOTEDESKTOP_CTL_BUFHEADER part of the packet. The packet type SHOULD be set to REMOTEDESKTOP_CTL_SERVER_ANNOUNCE.

2.2.1.8 REMOTEDESKTOP_CTL_VERSIONINFO_PACKET

The REMOTEDESKTOP_CTL_VERSIONINFO_PACKET indicates the version of the **Remote Desktop Protocol** that is supported by the novice. It includes a major and a minor version. This packet is sent either from the novice to the expert or vice versa.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
packetHeader																															
...																															
...																															
...																															
...																															
...																															
...																															
...																															
(packetHeader cont'd for 10 rows)																															
bufHeader																															
versionMajor																															
versionMinor																															

packetHeader (72 bytes): The REMOTEDESKTOP_CTL_PACKETHEADER part of the packet. The virtual channel name SHOULD be set to "RC_CTL".

bufHeader (4 bytes): The REMOTEDESKTOP_CTL_BUFHEADER part of the packet. The packet type SHOULD be set to REMOTEDESKTOP_CTL_VERSIONINFO.

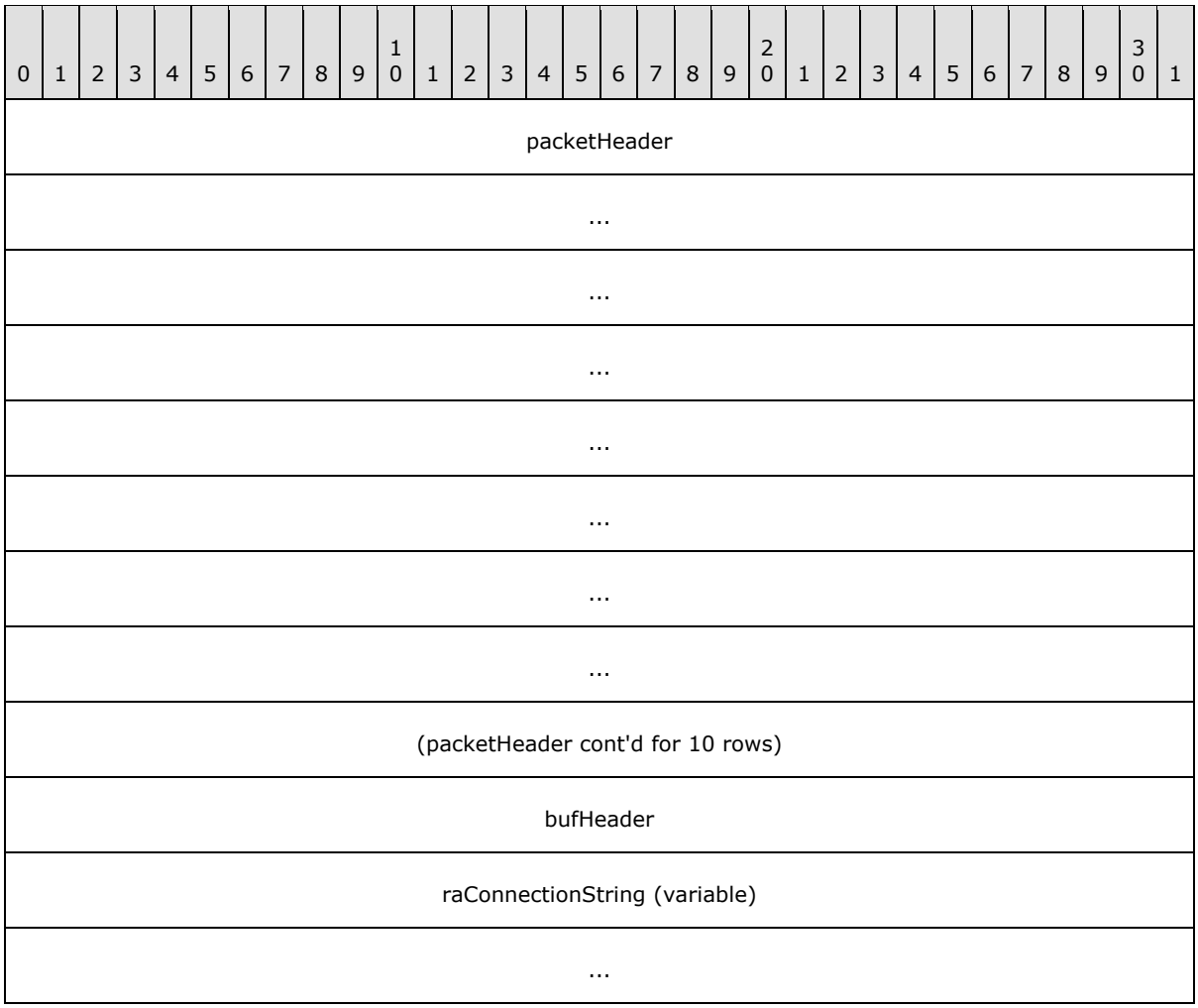
versionMajor (4 bytes): Major version number of the Remote Desktop Protocol implemented by the sender as a DWORD.

versionMinor (4 bytes): Minor version number of the Remote Desktop Protocol implemented by the sender as a DWORD.

The versionMajor and versionMinor fields MUST be set to 1 and 2, respectively. If this is not the case, the novice and the expert both disconnect.

2.2.1.9 REMOTEDESKTOP_RCCTL_REQUEST_PACKET

The REMOTEDESKTOP_RCCTL_REQUEST_PACKET is sent by the expert to the novice to request a view of the novice screen.<6>



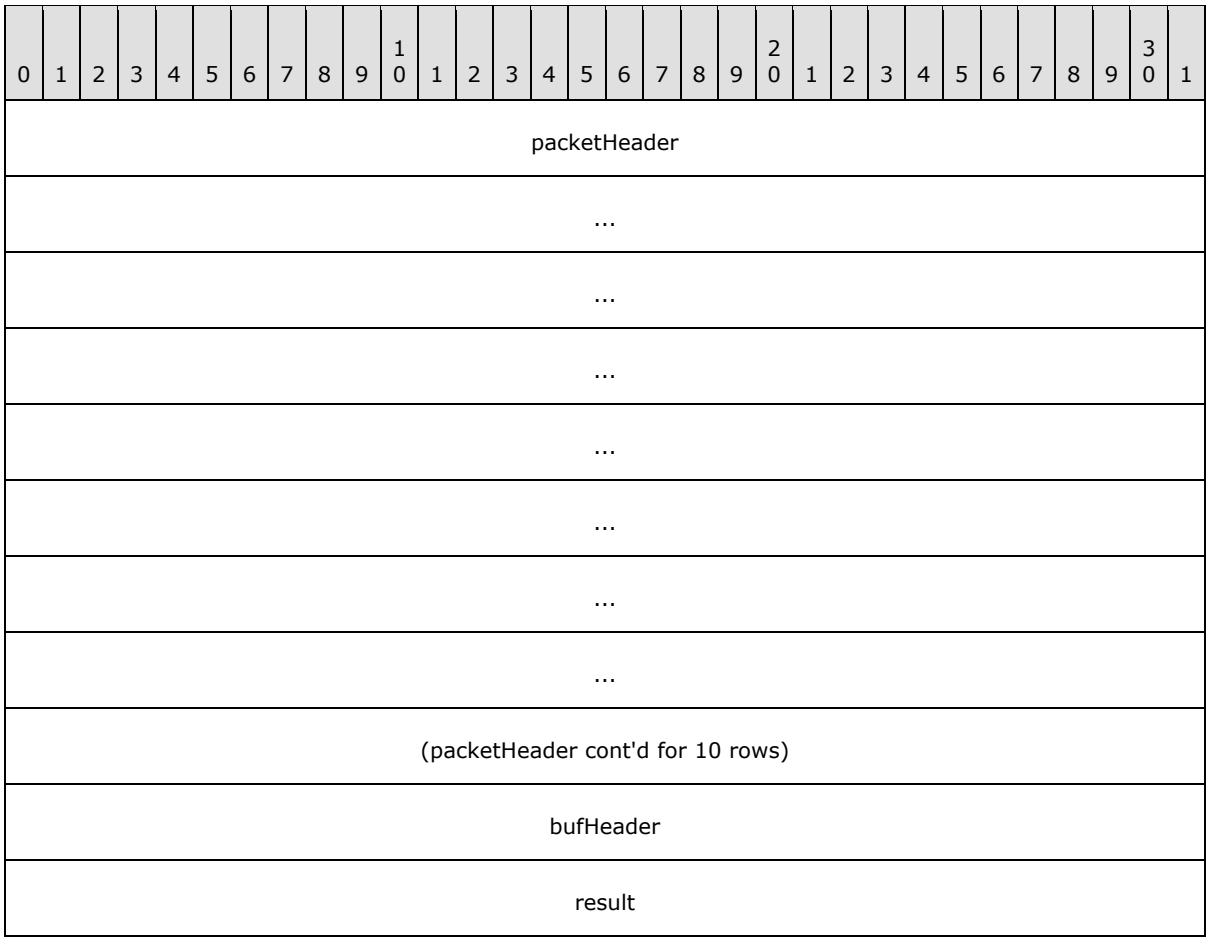
packetHeader (72 bytes): The REMOTEDESKTOP_CTL_PACKETHEADER part of the packet. The virtual channel name SHOULD be set to "RC_CTL".

bufHeader (4 bytes): The REMOTEDESKTOP_CTL_BUFHEADER part of the packet. The packet type SHOULD be set to REMOTEDESKTOP_CTL_REMOTE_CONTROL_DESKTOP.

raConnectionString (variable): A variable-length string containing a remote assistance connection string, as defined in [MS-RAI].

2.2.1.10 REMOTEDESKTOP_CTL_RESULT_PACKET

The REMOTEDESKTOP_CTL_RESULT_PACKET indicates the result of a client request.



packetHeader (72 bytes): The REMOTEDESKTOP_CTL_PACKETHEADER part of the packet. The virtual channel name SHOULD be set to "RC_CTL".

bufHeader (4 bytes): The REMOTEDESKTOP_CTL_BUFHEADER part of the packet. The packet type SHOULD be set to REMOTEDESKTOP_CTL_RESULT.

result (4 bytes): One of the values from the remote assistance error codes, as an integer.

2.2.1.11 REMOTEDESKTOP_CTL_VERIFY_PASSWORD_PACKET

The REMOTEDESKTOP_CTL_VERIFY_PASSWORD_PACKET contains the encrypted password. This packet is sent by the expert to the novice.<7>

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
packetHeader																															
...																															
...																															
...																															
...																															
...																															
...																															
...																															
(packetHeader cont'd for 10 rows)																															
bufHeader																															
Encrypted Password (variable)																															
...																															

packetHeader (72 bytes): The REMOTEDESKTOP_CTL_PACKETHEADER part of the packet. The virtual channel name SHOULD be set to "RC_CTL".

bufHeader (4 bytes): The REMOTEDESKTOP_CTL_BUFHEADER part of the packet. The packet type SHOULD be set to REMOTEDESKTOP_CTL_VERIFY_PASSWORD.

Encrypted Password (variable): Encrypted Password string included in the message as a BSTR for verification by the novice. In response, the novice sends the REMOTEDESKTOP_CTL_RESULT packet to the expert.

2.2.1.12 REMOTEDESKTOP_CTL_VISTA_EXPERT_PACKET

The REMOTEDESKTOP_CTL_VISTA_EXPERT_PACKET is an announcement that expert is running Windows Vista.

This packet is sent from expert to novice.<8>

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
packetHeader																															
...																															
...																															
...																															
...																															
...																															
...																															
...																															
(packetHeader cont'd for 10 rows)																															
bufHeader																															
Encrypted Password (variable)																															
...																															

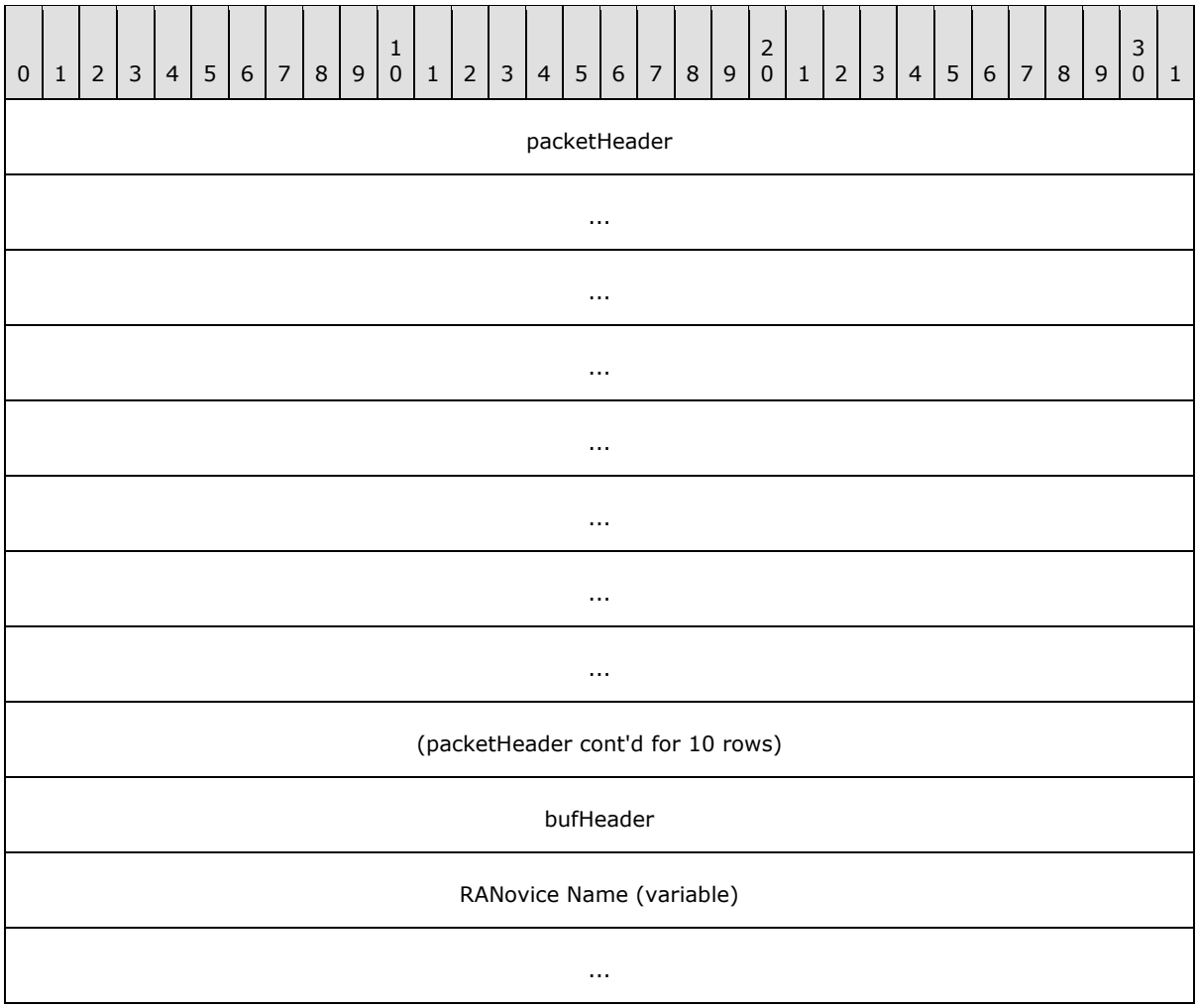
packetHeader (72 bytes): The REMOTEDESKTOP_CTL_PACKETHEADER part of the packet. The virtual channel name SHOULD be set to "RC_CTL".

bufHeader (4 bytes): The REMOTEDESKTOP_CTL_BUFHEADER part of the packet. The packet type SHOULD be set to REMOTEDESKTOP_CTL_VISTA_EXPERT.

Encrypted Password (variable): Encrypted Password string included in the message as a BSTR.

2.2.1.13 REMOTEDESKTOP_CTL_RANOVICE_NAME_PACKET

The REMOTEDESKTOP_CTL_RANOVICE_NAME_PACKET contains the novice name. The message MAY be sent either from novice to expert or vice versa, but only the novice SHOULD send this message.<9>



packetHeader (72 bytes): The REMOTEDESKTOP_CTL_PACKETHEADER part of the packet. The virtual channel name SHOULD be set to "RC_CTL".

bufHeader (4 bytes): The REMOTEDESKTOP_CTL_BUFHEADER part of the packet. The packet type SHOULD be set to REMOTEDESKTOP_CTL_RANOVICE_NAME.

RANovice Name (variable): Novice name string that is sent either from the expert to the novice or vice versa, as a **BSTR**.

2.2.1.14 REMOTEDESKTOP_CTL_RAEXPERT_NAME_PACKET

The REMOTEDESKTOP_CTL_RAEXPERT_NAME_PACKET specifies the name of the expert for display purposes. The message MAY be sent either from expert to novice or vice versa, but only the expert SHOULD send this message.<10>

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
packetHeader																															
...																															
...																															
...																															
...																															
...																															
...																															
...																															
(packetHeader cont'd for 10 rows)																															
bufHeader																															
RAEXPERT NAME (variable)																															
...																															

packetHeader (72 bytes): The REMOTEDESKTOP_CTL_PACKETHEADER part of the packet. The virtual channel name SHOULD be set to "RC_CTL".

bufHeader (4 bytes): The REMOTEDESKTOP_CTL_BUFHEADER part of the data. The packet type SHOULD be set to REMOTEDESKTOP_CTL_RAEXPERT_NAME.

RAEXPERT NAME (variable): The expert name string is sent either from the expert to the novice or vice versa, as a BSTR.

2.2.2 Session Control (RCCOMMAND) Messages

Session Control (RCCOMMAND) messages are XML formatted messages that are used for share control, file transfer initialization, and VoIP control. They MUST be sent and received over virtual channel 71.

The format of the RCCOMMAND message is as follows:

```
<RCCOMMAND
   NAME = "CDATA"
```

```
    FILENAME = "CDATA"
    FILESIZE = "CDATA"
    CHANNELID = "CDATA"
    WIDTH = "CDATA"
    HEIGHT = "CDATA"
    COLORDEPTH = "CDATA"
    SCHEMAVERSION = "CDATA"
    CONTROLCHANNELVERSION = "CDATA"
    VOIPVER = "CDATA"
    VOIPGOKEY = "CDATA"
    VOIPIPLIST = "CDATA"
/>
```

Based on the function, the relevant attributes of RCCOMMAND MAY be present in the message. The RCCOMMAND MUST be sent as a null-terminated Unicode string.

2.2.2.1 NAME

The NAME attribute of RCCOMMAND specifies the type of activity requested.

Attribute	Type	Required?	Description
NAME	CDATA	Yes	Leading attribute whose content determines the remainder of the message.

The following NAME value specifies file transfer commands.

Value	Meaning
FILEXFER	Indicates that the sender wants to begin a file transfer. This value is sent by either the client or the server.

The following NAME values specify session control messages.

Value	Meaning
ABORTRC	Sent by the client when an error occurs, terminating remote control.
ACCEPTRC	Sent by the server to accept a REMOTECONTROLSTART request from the client.
DENIEDRC	Sent by the server if it refuses a client remote control request due to group policy settings.
ERRORC	Sent by the server if an error occurs when attempting to grant the client remote control.
ESCRC	Sent by the server when it takes control back from the client by pressing the ESC key.
REJECTRC	Sent by the server to reject a REMOTECONTROLSTART request from the client.
REMOTECTRLSTART	Sent by the client to request remote control permissions from the server. <11>
REMOTECTRLEND	Sent by the client when it no longer wants to control the server.

Value	Meaning
TAKECONTROL	Sent by the server when it takes control back from the client.

The following NAME values specify synchronization parameters.

Value	Meaning
HELPERVERSION	Version number of the sender. This packet type is used in Session Initialization messages.
SCREENINFO	Describes the screen of the server; sent by the server.
TYPINGSTART	Indicates that the sender has begun typing a chat message. This packet type is sent by either the client or the server.
EXPERTIP	Notifies the novice of the IP address of the expert. It is only sent by the expert to the novice.<12>

The following NAME values specify voice-enabling commands.

Value	Meaning
BANDWTOLOW	Indicates that the sender would prefer to receive low bandwidth voice communications. This packet type is sent by either the client or the server.
BANDWTOHIGH	Indicates that the sender would prefer to receive high bandwidth voice communications. This packet type is sent by either the client or the server.
DISABLEVOICE	Indicates that the sender has disabled voice functionality. This packet type is sent by either the client or the server.
PRESTART	Indicates that the sender wishes to begin a voice session. This packet type is sent by either the client or the server.
PRESTARTNO	Indicates that the sender already has a PRESTART packet outstanding. This packet type is sent by the client or the server in response to a PRESTART packet.
PRESTARTYES	Indicates acceptance of a voice session request. This packet type is sent by the client or the server in response to a PRESTART packet.
VOIPGO	Indicates that the client should create an RTC connection to the server. This packet type is sent by the server in response to a VOIPPREGO or VOIPPREGO2 packet.
VOIPGONO	Indicates that an error has occurred and a voice session cannot be established. This packet type is sent by the server in response to a VOIPPREGO packet.
VOIPPREGO	Indicates that the server should or will begin listening for an RTC connection. This packet type is sent by the client or the server in response to a PRESTARTYES packet.
VOIPPREGO2	Indicates that the client will create an RTC connection to the server. This packet type is sent by the client in response to a VOIPPREGO packet.
VOIPQNO	Indicates that the sender has rejected a request for a voice session. This packet type is sent by the client or the server in response to a VOIPPREGO packet.
WIZARDBAD	Indicates that the sender has unsuccessfully used the Audio Tuning Wizard, and that the receiver should disable voice transmission. This packet type is sent by either the

Value	Meaning
	client or the server.
WIZARDGOOD	Indicates that the sender has successfully used the Audio Tuning Wizard, and that the receiver should enable voice transmission. This packet type is sent by either the client or the server.

2.2.2.2 FILENAME

The FILENAME attribute specifies the name of the file to be transferred.

Attribute	Type	Required?	Description
FILENAME	CDATA	No	Specifies filename for transfer. This attribute MUST be used only for FILEXFER packets.

2.2.2.3 FILESIZE

The FILESIZE attribute specifies the size of the file to be transferred.

Attribute	Type	Required?	Description
FILESIZE	CDATA	No	Specifies the file size for transfer. This attribute MUST be used only for FILEXFER packets.

2.2.2.4 CHANNELID

The CHANNELID attribute specifies the virtual channel on which the file data and FILEXFERACK, FILEXFERREJECT, and FILEXFEREND packets will be transferred.

Attribute	Type	Required?	Description
CHANNELID	CDATA	No	Specifies the virtual channel on which the file data and FILEXFERACK, FILEXFERREJECT, and FILEXFEREND packets will be transferred. This attribute MUST be used only for FILEXFER packets. The format of the virtual channel name is version dependent.<13>

2.2.2.5 WIDTH

The WIDTH attribute specifies the width of the server screen in pixels.

Attribute	Type	Required?	Description
WIDTH	CDATA	No	Width of the server screen in pixels. This attribute is only used in session initialization messages.

2.2.2.6 HEIGHT

The HEIGHT attribute specifies the height of the server screen in pixels.

Attribute	Type	Required?	Description
HEIGHT	CDATA	No	Height of the server screen in pixels. This attribute is only used in session initialization messages.

2.2.2.7 COLORDEPTH

The COLORDEPTH attribute specifies the color depth of the server screen in bits.

Attribute	Type	Required?	Description
COLORDEPTH	CDATA	No	Color depth of the server screen in bits. This attribute is only used in session initialization messages.

2.2.2.8 SCHEMAVERSION

The SCHEMAVERSION attribute specifies the version of the Remote Assistance Protocol for the sender.

Attribute	Type	Required?	Description
SCHEMAVERSION	CDATA	No	Version of the Remote Assistance Protocol for the sender. This attribute MUST only be used for SCREENINFO and HELPERVERSION packets.

2.2.2.9 CONTROLCHANNELVERSION

The CONTROLCHANNELVERSION attribute specifies the version of the remote assistance channel for the sender.

Attribute	Type	Required?	Description
CONTROLCHANNELVERSION	CDATA	No	Version of the remote assistance channel for the sender. This attribute MUST only be used for SCREENINFO and HELPERVERSION packets.

2.2.2.10 VOIPVER

The VOIPVER attribute MUST only be used for VOIPGO packets.

Attribute	Type	Required?	Description
VOIPVER	CDATA	No	This attribute MUST be used for VOIPGO packets.

2.2.2.11 VOIPGOKEY

The VOIPGOKEY attribute specifies that a key should be generated by the server and used by the client to create a real-time communication (RTC) connection.

Attribute	Type	Required?	Description
VOIPGOKEY	CDATA	No	Specifies that a key should be generated by the server and used by the client to create an RTC connection. This attribute should

Attribute	Type	Required?	Description
			only be used for VOIPGO packets.

2.2.2.12 VOIPIPLIST

The VOIPIPLIST specifies a comma-delimited list of IP addresses on which the server will accept a real-time communication (RTC) connection.

Attribute	Type	Required?	Description
VOIPIPLIST	CDATA	No	This attribute specifies a comma-delimited list of IP addresses on which the server will accept an RTC connection. This attribute MUST only be used for VOIPGO packets.

2.2.2.13 EXPERTIPDATA

The EXPERTIPDATA value specifies the IP address of the connecting expert.

Attribute	Type	Required?	Description
EXPERTIPDATA	CDATA	No	This attribute specifies the IP address of the connecting expert. It MUST only be used for EXPERTIP packets.

2.2.3 File Transfer Commands

The following values specify file transfer commands.

Value	Meaning
FILEXFERACK	Indicates successful receipt of a packet. This value MUST be sent by the receiver of the file after a packet is received.
FILEXFEREND	Indicates that all packets in the file have been sent. This value MUST be sent by the receiver of the file after all packets have been sent.
FILEXFEREJECT	Indicates that either the file transfer was canceled or an error occurred and the transmission was terminated. This value MAY be sent by the sender or receiver of the file.

2.2.4 Remote Assistance Error Codes

The following remote assistance error codes MUST be returned as part of REMOTEDESKTOP_CTL_RESULT.

Error Code	Description
SAFERROR_NOERROR 0	No errors occurred.
SAFERROR_NOINFO 1	An error occurred in a dependent component, and no detailed information is available.
SAFERROR_LOCALNOTERROR	Connection disconnected by local user.

Error Code	Description
3	
SAFERROR_REMOTEBYUSER 4	Connection disconnected by remote user.
SAFERROR_BYSERVER 5	Connection dropped by remote machine.
SAFERROR_DNSLOOKUPFAILED 6	DNS resolution failed.
SAFERROR_OUTOFMEMORY 7	A memory allocation error occurred.
SAFERROR_CONNECTIONTIMEDOUT 8	A connection could not be established within the timeout period.
SAFERROR_SOCKETCONNECTFAILED 9	Connection to the remote machine failed.
SAFERROR_HOSTNOTFOUND 11	The remote machine is unreachable.
SAFERROR_WINSOCKSENDFAILED 12	A socket write failed.
SAFERROR_INVALIDIPADDR 14	The IP address given is invalid.
SAFERROR_SOCKETRECVFAILED 15	A socket read failed.
SAFERROR_INVALIDENCRYPTION 18	An encryption error occurred.
SAFERROR_GETHOSTBYNAMEFAILED 20	Winsock name resolution failed.
SAFERROR_LICENSINGFAILED 21	Remote Desktop Protocol: Basic Connectivity and Graphics Remoting Specification [MS-RDPBCGR] licensing for the connection failed.
SAFERROR_ENCRYPTIONERROR 22	Remote Desktop Protocol: Basic Connectivity and Graphics Remoting Specification [MS-RDPBCGR] encryption error occurred.
SAFERROR_DECRYPTIONERROR 23	A Remote Desktop Protocol decryption error occurred.
SAFERROR_INVALIDPARAMETERSTRING 24	An invalid remote assistance connection string, as specified in [MS-RAI], was used.
SAFERROR_HELPSESSIONNOTFOUND 25	A remote assistance connection string [MS-RAI] was not found.

Error Code	Description
SAFERROR_INVALIDPASSWORD 26	An invalid password was used.
SAFERROR_HELPSESSIONEXPIRED 27	The remote assistance connection string, as specified in [MS-RAI], has expired.
SAFERROR_CANTOPENRESOLVER 28	The remote machine could not resolve the session.
SAFERROR_UNKNOWNSESSMGRERROR 29	An unknown error occurred in the remote session manager.
SAFERROR_CANTFORMLINKTOUSERSESSION 30	The remote machine could not establish a connection to the specified user session.
SAFERROR_RCPROTOCOLERROR 32	A remote control protocol error occurred.
SAFERROR_RCUNKNOWNERROR 33	An unknown remote control error occurred.
SAFERROR_INTERNALERROR 34	An internal error occurred.
SAFERROR_HELPEERESPONSEPENDING 35	This code is not used.
SAFERROR_HELPEESAIDYES 36	This code is not used.
SAFERROR_HELPEEALREADYBEINGHELPED 37	The user is already under remote control.
SAFERROR_HELPEECONSIDERINGHELP 38	This code is not used.
SAFERROR_HELPEENOTFOUND 39	The specified remote user was not found.
SAFERROR_HELPEENEVERRESPONDED 40	The remote user did not accept the remote control request during the timeout period.
SAFERROR_HELPEESAIDNO 41	The remote user denied the remote control request.
SAFERROR_HELPSESSIONACCESSDENIED 42	The remote user does not have access to the specified connection string, as specified in [MS-RAI].
SAFERROR_USERNOTFOUND 43	The specified remote user was not found.
SAFERROR_SESSMGRERRORNOTINIT	A remote error occurred with the session manager.

Error Code	Description
44	
SAFERROR_SELFHELPNOTSUPPORTED 45	Attempting to control your own session remotely is not supported.
SAFERROR_INCOMPATIBLEVERSION 47	An incompatible version was given.
SAFERROR_SESSIONNOTCONNECTED 48	This code is not used.
SAFERROR_SYSTEMSHUTDOWN 50	The remote system is shutting down.
SAFERROR_STOPLISTENBYUSER 51	The remote system has stopped listening for an incoming connection.
SAFERROR_WINSOCK_FAILED 52	A Winsock call has failed.
SAFERROR_MISMATCHPARMS 53	A parameter mismatch has occurred.
SAFERROR_SHADOWEND_BASE 300	Remote control of the user session has been terminated.
SAFERROR_SHADOWEND_CONFIGCHANGE 301	Remote control of the user session terminated due to a color depth or resolution change.
SAFERROR_SHADOWEND_UNKNOWN 302	Remote control of the user session has ended.

3 Protocol Details

The following sections specify details of the Remote Assistance Protocol, including session initialization, file transfer, chat, share control remote assistance (RA), and voice abstract data models and message processing rules.

3.1 Session Initialization Expert (Client) Details

After a remote assistance connection string is obtained by the expert (as specified in the remote assistance initiation protocol in [MS-RAI]) a basic remote assistance connection is established from the expert to the novice using the Remote Desktop Protocol: Basic Connectivity and Graphics Remoting Specification, as specified in [MS-RDPBCGR]. This basic connection does not allow the expert to view the novice screen. Before the expert can view the novice screen, control messages MUST be exchanged between the novice and the expert. When this exchange is completed successfully, the expert can view the novice screen, and the remote assistance session initialization is completed.

Sections 3.1 and 3.2 specify message exchange between the novice and the expert to establish a remote assistance session.

The following diagram shows the connection sequence between novice and expert.<14>

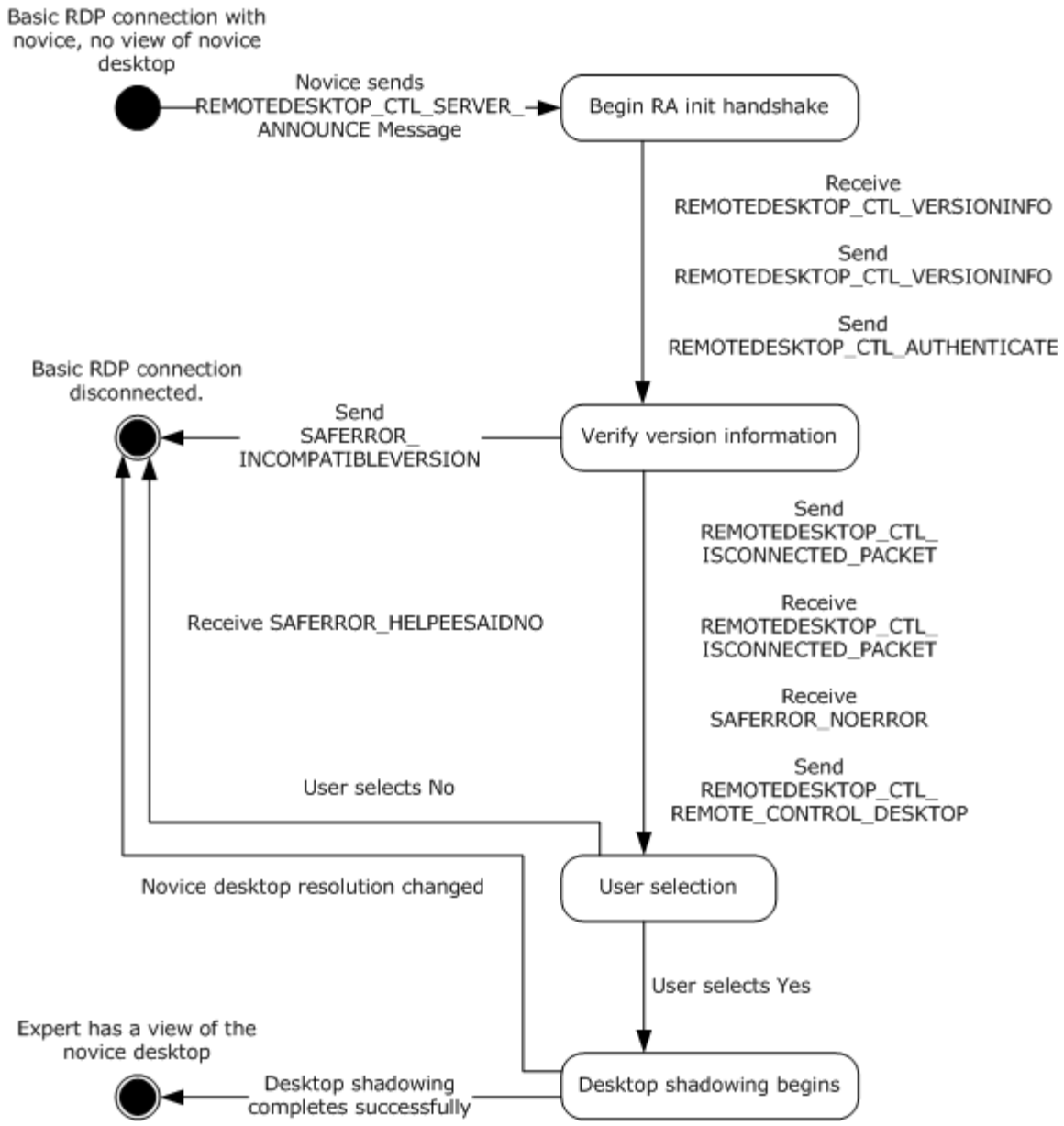


Figure 1: Remote assistance session connection sequence between novice (server) and expert (client)

3.1.1 Abstract Data Model

The message signaling that takes place in the session initialization protocol is to complete the remote assistance connection; that is, to change the state from the basic remote assistance connection state in which the expert does not have the view of the novice screen to the state in which the expert has the view of the novice screen.

When the control message arrives to the expert indicating that a remote assistance connection has completed successfully or that there was an error during connection, the expert SHOULD keep track of this state change.

3.1.2 Timers

There are no timers associated with session initialization on the expert.

3.1.3 Initialization

The Remote Assistance Protocol sends control message packets on the RC_CTL virtual channel. A virtual channel named "RC_CTL" MUST be opened before any control messages can be sent or received.

3.1.4 Higher-Layer Triggered Events

The messages and events described here have no other dependent events or messages from a higher layer.

3.1.5 Message Processing Events and Sequencing Rules

This section describes the sequence of the control packets that the expert receives as well as the control message packets with which the expert responds.

Figure 2: Remote access session initialization sequence diagram

The expert receives the REMOTEDESKTOP_CTL_SERVERANNOUNCE control message packet. In response, the expert sends a REMOTEDESKTOP_CTL_VERSIONINFO packet with the following values:

```
REMOTEDESKTOP MAJOR VERSION = 1
REMOTEDESKTOP MINOR VERSION = 2
```

The expert also sends the REMOTEDESKTOP_CTL_AUTHENTICATE packet. This packet includes the remote assistance connection string and the expert Blob, as specified in [MS-RAI]. The expert name MAY be included in the blob so the novice can be informed.

The expert receives the REMOTEDESKTOP_CTL_VERSIONINFO packet. The expert extracts the major and minor version numbers from the packet. The major version number MUST be equal to 1, and the minor version number MUST be equal to 2; otherwise, a SAFERROR_INCOMPATIBLEVERSION error is returned in the REMOTEDESKTOP_CTL_RESULT packet to the novice.

When SAFERROR_NOERROR is returned by the novice indicating that the remote assistance connection string is good, the expert sends the REMOTEDESKTOP_CTL_REMOTE_CONTROL_DESKTOP message element from the REMOTEDESKTOP_CTL_BUFHEADER packet to the novice. This message is followed by the remote assistance connection string. After receiving the remote control request message, the novice completes the desktop shadowing so the expert can view the novice screen.<15>

The REMOTEDESKTOP_CTL_RESULT packet can be received with the following error codes.

Value	Meaning
SAFERROR_HELPEENOTFOUND	Sent from novice to expert when novice may have logged off.
SAFERROR_HELPEESAIDNO	Sent from novice to expert when novice rejects the remote assistance connection. This error is returned when the novice rejects remote assistance by clicking No in the Remote Assistance Acceptance UI dialog box; otherwise, when the novice clicks Yes in this UI dialog box, the novice desktop shadowing completes, and the expert can view the novice screen.
SAFERROR_INVALIDPASSWORD	Returned by the novice when the password entered by the expert is incorrect.<16>
PASSWORDS_DONT_MATCH	Returned by the novice when the password entered by the expert is incorrect.<17>

After receiving REMOTEDESKTOP_CTL_RESULT the novice or expert MAY send one of the following packets:

Value	Meaning
REMOTEDESKTOP_CTL_DISCONNECT	Sent from the novice to the expert when the novice has disconnected the remote assistance session.<18>
REMOTEDESKTOP_CTL_VERIFY_PASSWORD	Authentication request received by the novice from the expert.<19>

The expert also sends the REMOTEDESKTOP_CTL_ISCONNECTED packet every 30 seconds over an idle connection. The expert SHOULD receive this packet every 30 seconds. If the expert does not receive this packet in a 30-second idle interval, it disconnects.<20>

3.1.6 Timer Events

The REMOTEDESKTOP_CTL_ISCONNECTED packet MUST be used to track the state of a remote assistance connection. Both the expert and the novice MUST send this packet once every 30 seconds to indicate a connected state.<21>

3.1.7 Other Local Events

The Remote Assistance Protocol does not have external event dependencies.

3.2 Session Initialization Novice (Server) Details

After a remote assistance connection string is obtained by the expert (as specified in [MS-RAI]), a basic remote assistance connection is established from the expert to the novice using the Remote Desktop Protocol: Basic Connectivity and Graphics Remoting Specification, as specified in [MS-

RDPBCGR]. This basic connection does not have the Expert View capability; that is, the expert cannot view the novice screen. Before the expert can view the novice screen, there is control message exchange between the novice and the expert. When this exchange is completed successfully, the expert is granted a view of the novice screen, and the remote assistance session is considered established.

The remote assistance session initialization protocol sends control message packets on the RC_CTL virtual channel. The RC_CTL virtual channel persists throughout the duration of the remote assistance connection.

If any errors occur during signaling, remote assistance error codes are returned in the REMOTEDESKTOP_CTL_RESULT over the RC_CTL channel.

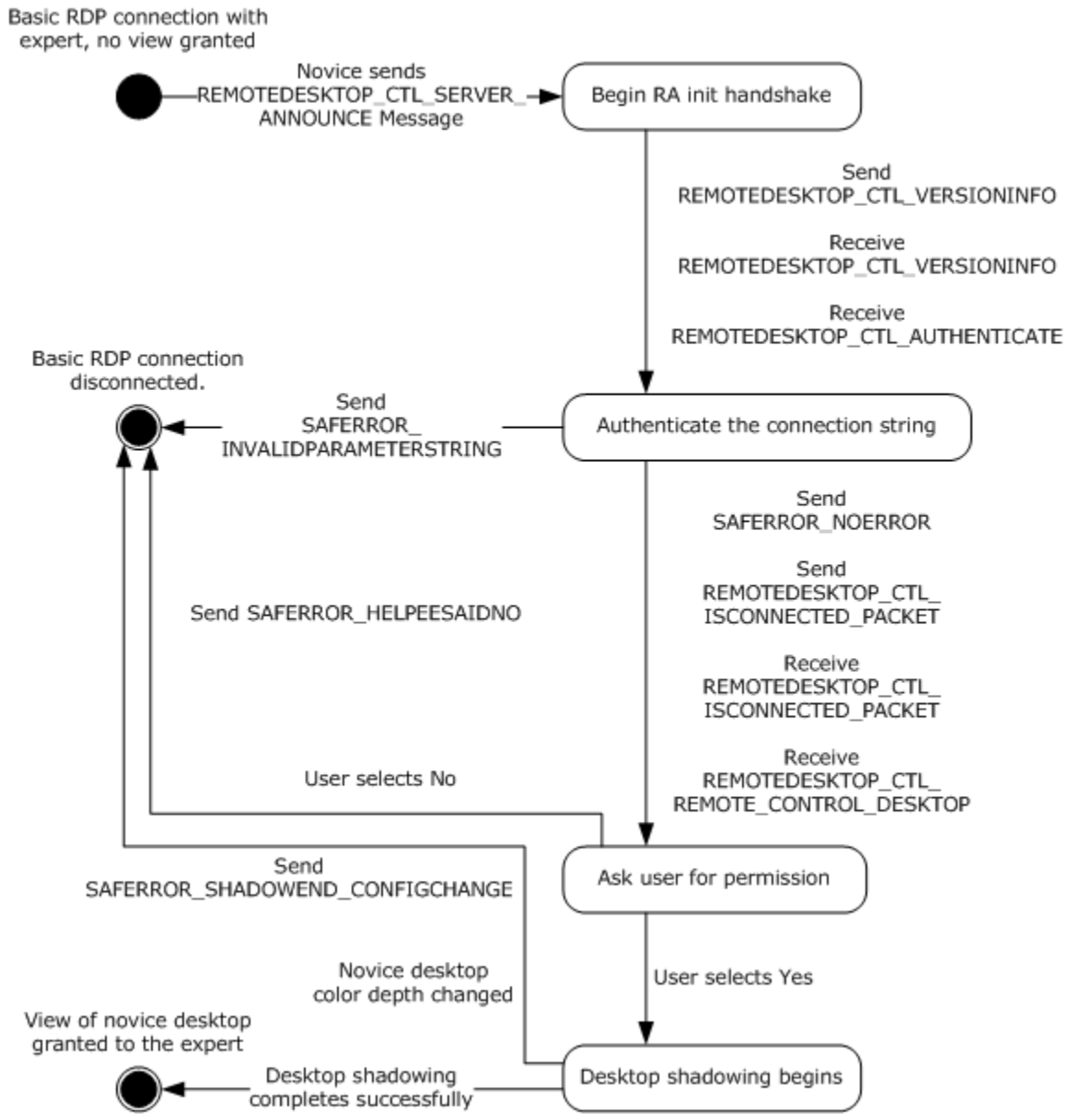


Figure 3: Remote assistance initialization state diagram (novice/server perspective)

3.2.1 Abstract Data Model

The message signaling that takes place in the session initialization protocol is to complete the remote assistance connection; that is, to change the state from the basic remote assistance connection state in which the expert does not have the view of the novice screen to the state in which the expert has the view of the novice screen.

When the remote assistance connection has completed successfully, or if there was an error during connection, the novice SHOULD keep track of this state change.

3.2.2 Timers

There are no timers associated with session initialization on the novice.

3.2.3 Initialization

The Remote Assistance Initiation Protocol sends control message packets on the RC_CTL virtual channel. Therefore, a virtual channel with the name RC_CTL MUST be created before any control messages can be sent or received.

3.2.4 Higher-Layer Triggered Events

The messages and events described in this specification have no other dependent events or messages from a higher layer.

3.2.5 Message Processing Events and Sequencing Rules

This section describes the control message packets that the novice receives and the control message packets that the novice responds with.

When the novice sends the REMOTEDESKTOP_CTL_SERVER_ANNOUNCE packet, it expects the following two packets to be sent by the expert:

- REMOTEDESKTOP_CTL_VERSIONINFO
- REMOTEDESKTOP_CTL_AUTHENTICATE

The novice also sends the REMOTEDESKTOP_CTL_VERSIONINFO packet with the following values:

- REMOTEDESKTOP MAJOR VERSION = 1
- REMOTEDESKTOP MINOR VERSION = 2

The novice receives the REMOTEDESKTOP_CTL_VERSIONINFO packet and extracts the major and minor version numbers from the packet. The major version number MUST be equal to 1, and the minor version number MUST be equal to 2; otherwise, the SAFERROR_INCOMPATIBLEVERSION error is returned in the REMOTEDESKTOP_CTL_RESULT packet to the expert.

The novice receives the REMOTEDESKTOP_CTL_AUTHENTICATE packet and extracts the remote assistance connection string. The novice authenticates whether or not the expert is connecting with the correct remote assistance connection string. If the remote assistance connection string is invalid, the SAFERROR_INVALIDPARAMETERSTRING error is returned to the expert. If the remote assistance connection string is valid, the expert blob is extracted. Also, the success code SAFERROR_NOERROR is returned in the REMOTEDESKTOP_CTL_RESULT packet.

The REMOTEDESKTOP_CTL_RESULT packet can be received with the following error codes:

Value	Meaning
SAFERROR_HELPEENOTFOUND	Sent from novice to expert when novice may have logged off.
SAFERROR_HELPEESAIDNO	Sent from novice to expert when novice rejects the remote assistance connection. This error is returned when the novice rejects remote assistance by clicking No in the Remote Assistance Acceptance UI dialog box; otherwise, when the novice clicks Yes in this UI dialog box, the novice desktop shadowing completes, and the expert can view the novice screen.
SAFERROR_INVALIDPASSWORD	Returned by the novice when the password entered by the expert is incorrect.<22>
PASSWORDS_DONT_MATCH	Returned by the novice when the password entered by the expert is incorrect.<23>

After receiving REMOTEDESKTOP_CTL_RESULT the novice or expert MAY send one of the following packets:

Value	Meaning
REMOTEDESKTOP_CTL_DISCONNECT	Sent from the novice to the expert when the novice has disconnected the remote assistance session.<24>
REMOTEDESKTOP_CTL_VERIFY_PASSWORD	Authentication request received by the novice from the expert.<25>

The novice also sends the REMOTEDESKTOP_CTL_ISCONNECTED packet every 30 seconds over an idle connection. The novice SHOULD receive this packet every 30 seconds. If the novice does not receive this packet in a 30-second idle interval, it disconnects.<26>

3.2.6 Timer Events

There are no timers or timeout periods specified by this protocol.

3.2.7 Other Local Events

This protocol does not have external event dependencies.

3.3 File Transfer Sender Details

File transfer in a remote assistance session is initiated by the sender of the file; there is no mechanism for the receiver of a file to request the transfer to begin. This section will only focus on the file transfer messages and the sequence expected from the file sender's side. A high-level state machine depicting message exchanges from the sender's point of view is shown here. File transfer supports only one file being transferred at a time.

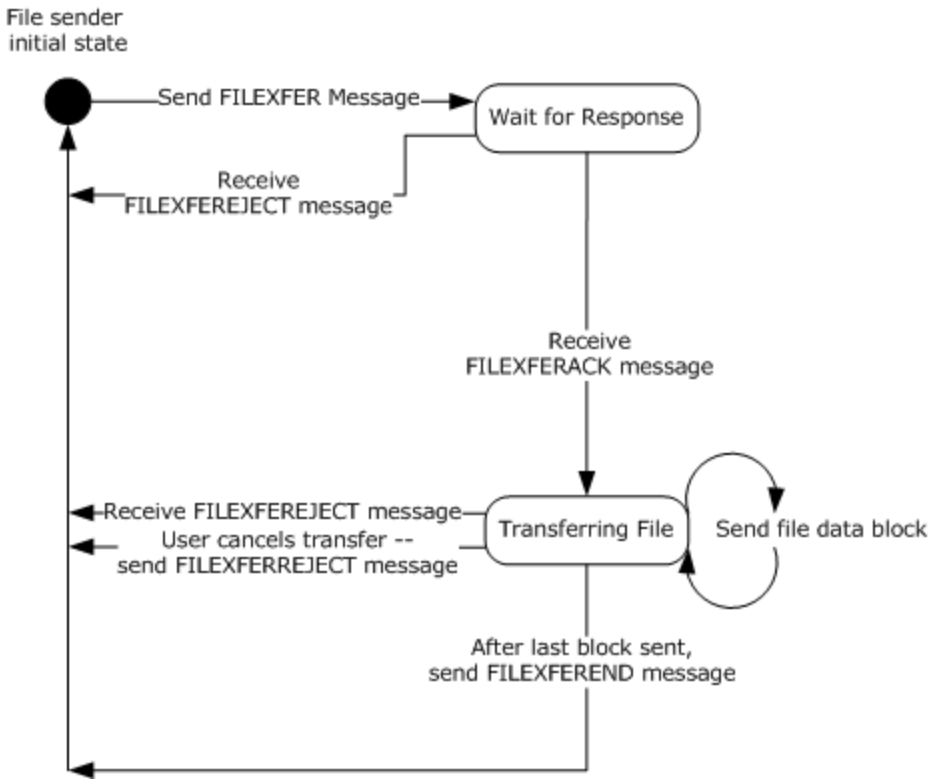


Figure 4: Session-state diagram (file sender perspective)

3.3.1 Abstract Data Model

There is no data model needed to maintain internal state.

3.3.2 Timers

There are no timers or timeout periods specified by the Remote Assistance Protocol.

3.3.3 Initialization

The virtual channel used to signal the intent of file transfer (RCCOMMAND NAME="FILEXFER") is initialized when the remote assistance connection is first established. The virtual channel that is used to transfer the file data MUST be opened before sending the FILEXFER message because opening the channel after the message is dispatched can lead to missed response messages from the receiving side. The name of this virtual channel is specified by the sender as an attribute in the FILEXFER message.

3.3.4 Higher-Layer Triggered Events

The messages and events described in this specification have no other dependent events or messages from a higher layer.

3.3.5 Message Processing Events and Sequencing Rules

There are two virtual channels involved during file transfer. The virtual channel 71 is used to initiate file transfer through a RCCOMMAND message, and a second dynamically created virtual channel is used to transfer the file data.

The first thing the file sender must do is to signal the need to copy a file from its computer to the receiver's computer. This is accomplished by sending a RCCOMMAND message on the virtual channel 71 with the NAME attribute set to FILEXFER. The message must also include the attributes FILENAME, FILESIZE, and CHANNELID. The FILENAME attribute should be set to the original name of the file, as seen by the sender. The FILESIZE attribute should be set to the size in bytes of the file about to be sent. The CHANNELID MUST be set to the name of the virtual channel that the file data will be sent on. Also, the CHANNELID will be the channel through which the sender expects to get any response from the receiver.

After sending the RCCOMMAND message, the sender waits for a response to the file transfer request on the file transfer channel specified in the message just sent. If the sender receives the FILEXFEREJECT message, it should not expect any more messages on the channel, and should not send file data on the channel to the receiver. On receipt of the FILEXFERACK message, the sender should proceed with sending the actual file data to the receiver on the file transfer channel.

Sending a file requires the sender to break the file into blocks and send them serially to the receiver. The protocol itself has no limit for the size of the blocks.<27> The last block MAY be of a shorter length, if the file data is not exactly divisible by the block size chosen.

In all cases, the data sent must be sent serially because there is no header information to allow for odd order reassembly on the receiving side. Also, there is no acknowledgment of the receipt of the block from the receiving side provided for in this protocol. The remote assistance application uses call backs from the Terminal Services layer signaling the successful sending of a block to the receiving side, which causes the next block to be sent.

If, while the file is being sent, the sending user wants to cancel the transfer, this user should send the FILEXFEREJECT message to the receiver on the file transfer channel. If the receiver wants to cancel the file transfer, it sends the FILEXFEREJECT message to the sender on the file transfer channel. In either case, the sender should stop sending data blocks. No other messages should be expected or sent on the file transfer channel after the sending or receiving of the FILEXFEREJECT message.

For the file sender, there are several messages that can arrive during the entire process (see section 2.2.3). When a message arrives, a string comparison to detect the type of message arriving is all that is needed. The state machine shown in section 3.3 illustrates the expected sequence of messages; any message that arrives out of sequence SHOULD cause the receiver to generate a FILEXFEREJECT message to signal the error in processing messages. If errors in the sequence are ignored, it is possible that file corruption can occur on the file receiving side.

A sample follows of the messages exchanged over time between the file sender and receiver.

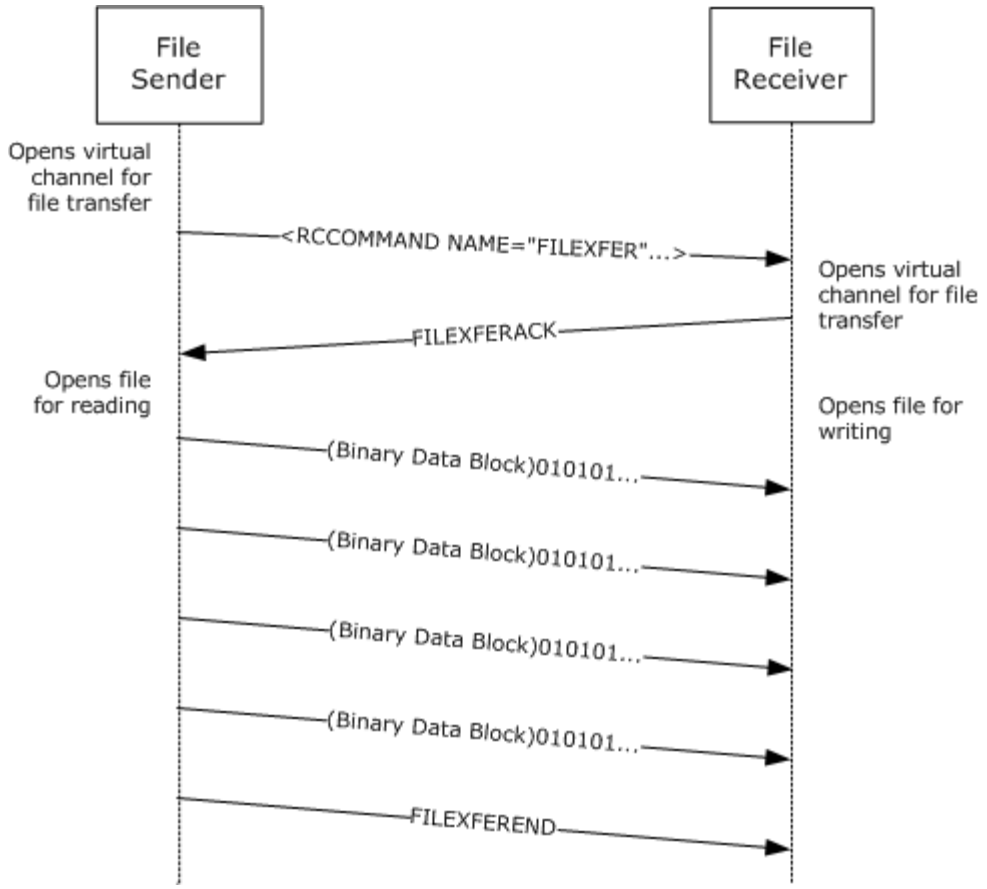


Figure 5: File transfer packet sequencing

3.3.6 Timer Events

There are no timer events associated with the Remote Assistance Protocol.

3.3.7 Other Local Events

The Remote Assistance Protocol does not have external event dependencies.

3.4 File Transfer Receiver Details

File transfer in a remote assistance connection is initiated by the sender of the file; there is no mechanism for the receiver of a file to request the transfer to begin. The method employed to transfer the file from one machine to the other is very basic. When considering the file transfer exchange that happens, other messages for things such as session control and VoIP are not considered, although they can be sent and received at any point during the sequence described below. This section only focuses on the file transfer messages and the sequence expected from the file receiver's side. A high-level state machine depicting message exchanges from the receiver's point of view follows.

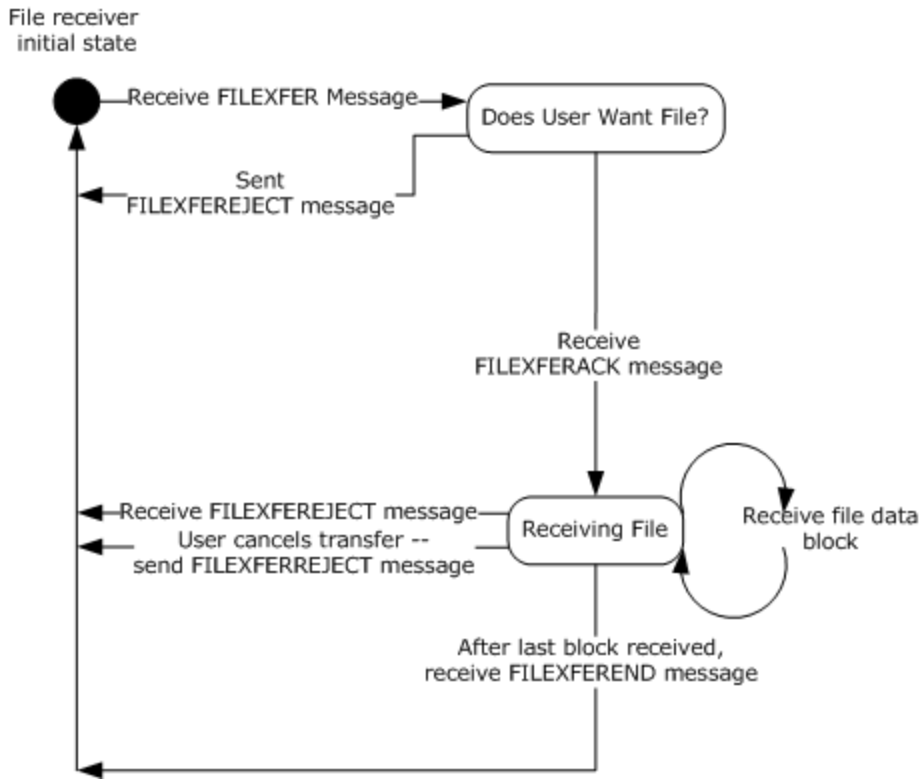


Figure 6: Session-state diagram (file receiver perspective)

3.4.1 Abstract Data Model

There is no internal state that needs to be maintained that requires abstract data models.

3.4.2 Timers

There are no timers associated with file transfer.

3.4.3 Initialization

The virtual channel used to receive the signal for file transfer (RCCOMAND NAME="FILEXFER") is initialized when the remote assistance connection is first established. The virtual channel that is used to transfer the file data MUST be opened before sending the FILEXFERACK message or the FILEXFEREJECT message because opening this virtual channel is used to send the reply that is expected by the sender. The name of this virtual channel is specified by the sender as an attribute in the FILEXFER message, and MUST be named RA_FX.

3.4.4 Higher-Layer Triggered Events

There are no higher-layer triggered events that are addressed by this portion of the Remote Assistance Protocol.

3.4.5 Message Processing Events and Sequencing Rules

For the file receiver, there are several messages that can arrive during the entire process (see section 2.2.3). When a message arrives, a string comparison MAY be used to determine the type of message that has arrived. The state machine shown in section 3.4 shows the expected sequence of messages; any messages that arrive out of sequence SHOULD cause the receiver of the message to generate a FILEXFEREJECT message to signal the error in processing the messages. If errors in the sequence are ignored, it is possible that file corruption can occur on the file receiving side.

The first message that is received is an RCCOMMAND message on the virtual channel 71 with the NAME attribute set to FILEXFER. The message MUST also include the attributes FILENAME, FILESIZE, and CHANNELID. The FILENAME attribute SHOULD be set to the original name of the file, as seen by the sender of the file. The FILESIZE attribute SHOULD be set to the size in bytes of the file about to be sent. The CHANNELID MUST be set to the name of the virtual channel that the file data will be sent on. Also, the CHANNELID will be the channel through which the file sender expects to get a response from the file receiver.

After receiving this message, the file receiver MUST send a response to the file transfer request on the channel specified in the message just received. If the file transfer is wanted, the file receiver MUST send the FILEXFERACK message to the file sender. After sending the FILEXFERACK message, the file receiver SHOULD prepare for file data to arrive on the same channel. If the file transfer is not wanted, the file receiver MUST send the FILEXFEREJECT message. If the FILEXFEREJECT message is sent, the file receiver SHOULD not expect any more messages or file data, and the channel SHOULD not send any more messages on the virtual channel.

When receiving file data, the file will be received in discrete blocks. The protocol itself has no limit for the size of the blocks.<28> The last block MAY be of a shorter length, if the file data is not exactly divisible by the block size chosen.

In all cases, the data must be sent serially because there is no header information to allow for odd order reassembly on the receiving side. Also, there is no acknowledgment of the receipt of the block from the receiving side provided for in this protocol.

If, while the file is being sent, the receiving user wants to cancel the transfer, this user SHOULD send the FILEXFEREJECT message to the file sender on the file transfer channel. If the sender wants to cancel the file transfer, it sends the FILEXFEREJECT message to the sender on the file transfer channel. In either case, the sender SHOULD stop sending data blocks. No other messages should be expected or sent on the file transfer channel after the sending or receiving of the FILEXFEREJECT message.

A sample follows of the messages exchanged over time between the file sender and receiver.

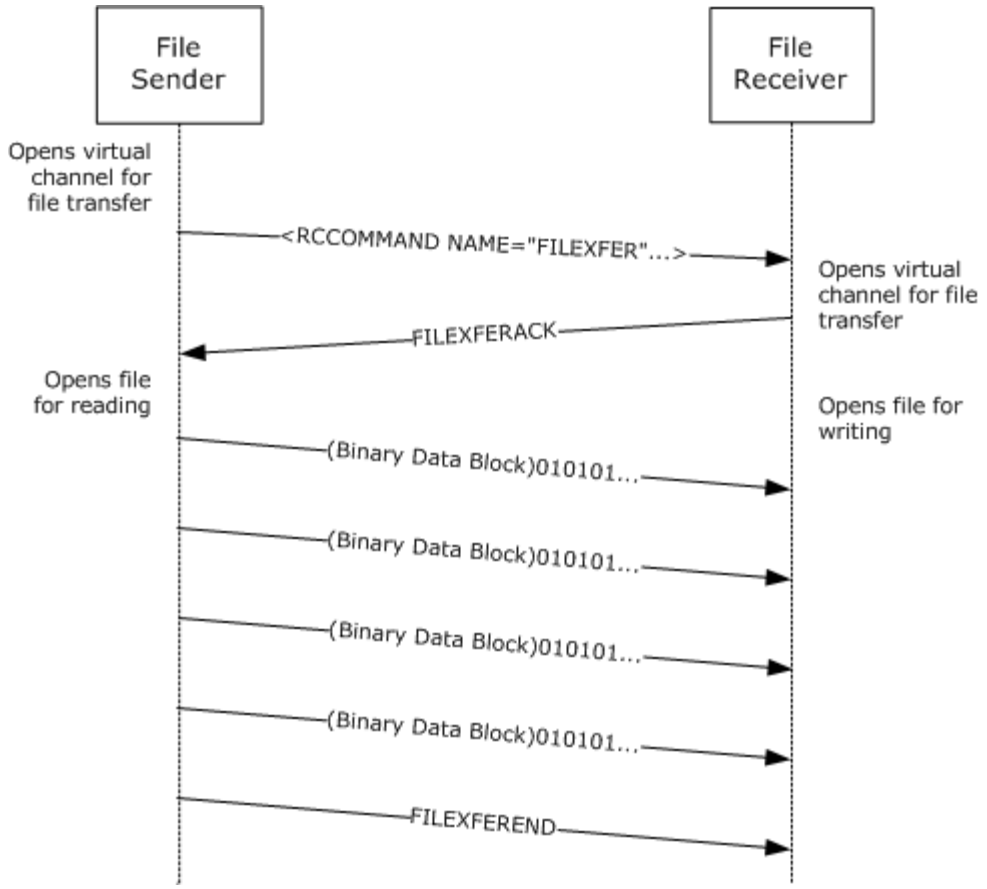


Figure 7: File transfer process during remote assistance session

3.4.6 Timer Events

There are no timer events associated with this portion of the Remote Assistance Protocol.

3.4.7 Other Local Events

There are no local events that have an impact on this portion of the Remote Assistance Protocol.

3.5 Chat (Text) Sender Details

Once the remote assistance connection is established, the application SHOULD open the virtual channel 70 to send and receive chat messages. Because there are only two computers involved in the connection, it is assumed that any data received on the chat virtual channel 70 is a Unicode string to be shown to the user as a message from the other person they are connected to. There is no header information, and each block of data received on the channel is conceptually thought of as a distinct message from the other user. To send a chat message, the message formatted as a null-terminated Unicode string should be sent on virtual channel 70.

3.5.1 Abstract Data Model

There is no data model used in this portion of the Remote Assistance Protocol.

3.5.2 Timers

There are no timers associated with this portion of the Remote Assistance Protocol.

3.5.3 Initialization

The virtual channel that allows chat messages to be exchanged is initialized immediately after the remote assistance connection is established. The virtual channel name is 70, and it is used solely to transfer Unicode strings as chat messages between the two connected computers.

3.5.4 Higher-Layer Triggered Events

There are no higher-layer triggered events that affect this portion of the Remote Assistance Protocol.

3.5.5 Message Processing Events and Sequencing Rules

To send a chat message, the chat message must be sent on the virtual channel 70 as a null-terminated Unicode string. The string must not exceed 1,024 bytes in size (including null termination). There is no expected response from the receiving side.

3.5.6 Timer Events

There are no timer events associated with sending chat messages.

3.5.7 Other Local Events

No local events are handled through this section of the Remote Assistance Protocol.

3.6 Chat (Text) Receiver Details

Once the remote assistance connection is established, the application SHOULD open the virtual channel 70 to send and receive chat messages. Because there are only two computers involved in the connection, it is assumed that any data received on the chat virtual channel 70 is a Unicode string to be shown to the user as a message from the other person they are connected to. There is no header information, and each block of data received on the channel is conceptually thought of as a distinct message from the other user. To send a chat message, the message formatted as a null-terminated Unicode string should be sent on virtual channel 70.

3.6.1 Abstract Data Model

There is no data model associated with this portion of the Remote Assistance Protocol.

3.6.2 Timers

There are no timers required for the chat portion of the Remote Assistance Protocol.

3.6.3 Initialization

The virtual channel that allows chat messages to be exchanged is initialized immediately after the remote assistance connection is established. The virtual channel name is 70 and is used solely to transfer Unicode strings as chat messages between the two connected computers.

3.6.4 Higher-Layer Triggered Events

There are no events that are used in this section of the Remote Assistance Protocol.

3.6.5 Message Processing Events and Sequencing Rules

When a message arrives on the virtual channel reserved for chat, it is always assumed to be a null-terminated Unicode string. Because there can be only one possible sender, the message has no header and no packet information. Chat messages of a length greater than 1,024 bytes MUST NOT be sent; therefore, each packet MUST be considered a discrete message that SHOULD be displayed in its entirety to the receiving user. There are no error codes or responses expected or sent in response to receiving a chat message.

3.6.6 Timer Events

There are no timers associated with this portion of the Remote Assistance Protocol.

3.6.7 Other Local Events

There are no events that are associated with this portion of the Remote Assistance Protocol.

3.7 Share Control Remote Assistance Expert (Client) Details

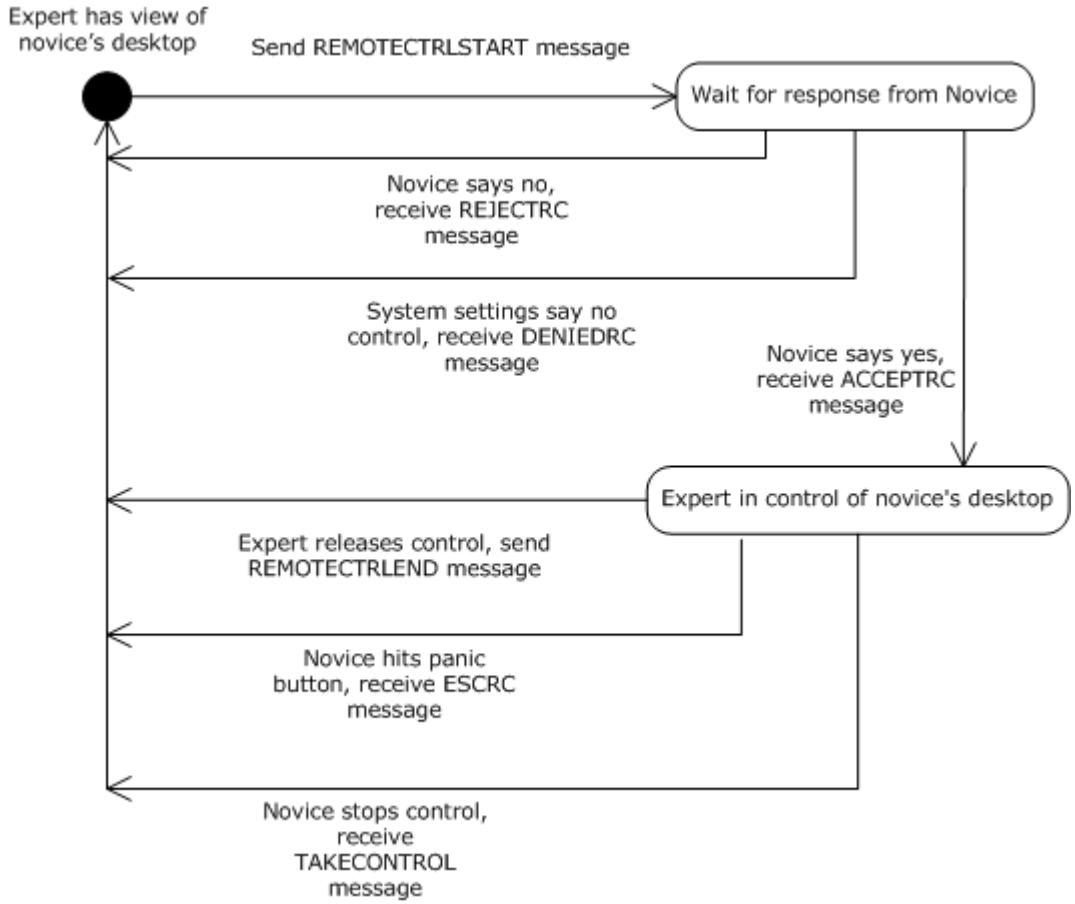


Figure 8: Desktop sharing session lifecycle (expert/client perspective)

Normally during the remote assistance connection, the expert can only observe what the novice is seeing on their screen. If the expert wants to control the mouse and keyboard on the novice machine, the expert can request control from the novice. This portion of the Remote Assistance Protocol concerns the messages that are exchanged as permission for the expert is sought, granted, denied, and/or finally revoked or given up. The state machine shown above describes the messages involved in the exchange between expert and novice. Details provided in this section describe what is expected from the point-of-view of the expert.

3.7.1 Abstract Data Model

The message exchanging that occurs in this section of the Remote Assistance Protocol is used to synchronize the state of the desktop sharing. In response to a share control request from the expert, the novice SHOULD first enable the sharing of the screen, and then send a response to the expert that share control has been granted. When share control is stopped by the novice, the novice MUST send a message indicating that desktop sharing has been stopped. When share control is released by the expert, the expert MUST send a message indicating this action. When the novice receives this message, the novice MUST disable share control of the screen.

3.7.2 Timers

There are no timers associated with this portion of the Remote Assistance Protocol.

3.7.3 Initialization

The virtual channel used to send messages described in this section of the protocol (RCCOMMAND) is initialized when the remote assistance connection is first established. All RCCOMMAND messages are sent on the virtual channel named 71.

3.7.4 Higher-Layer Triggered Events

This section of the Remote Assistance Protocol does not depend on higher-layer triggered events.

3.7.5 Message Processing Events and Sequencing Rules

For the expert, there are several messages that are sent or that can arrive during the entire process of requesting control (see section 2.2.2). When a message arrives, a string comparison MAY be used to determine the type of message that has arrived. The state machine shown in section 3.7 illustrates the expected sequence of messages; any messages that arrive out of sequence SHOULD be ignored by the receiving side. Processing of messages that arrive out of expected sequence MAY lead to incorrect event notification. All messages sent and received in this portion of the Remote Assistance Protocol are sent on the virtual channel named 71.

To assume control of the novice's mouse and keyboard, the expert MUST send the message <RCCOMMAND NAME="REMOTECTRLSTART"/>. When the novice receives this message, the Remote Assistance Protocol provides for three different responses:

1. If the novice wants to allow the expert to have control of the screen, the response <RCCOMMAND NAME="ACCEPTRC"/> MUST be sent.

2. If the novice does not want to allow the expert to control the screen, the novice MUST send the response <RCCOMMAND NAME="REJECTRC"/>.

3. Optionally, if system settings on the novice do not permit share control, the novice MAY send the response <RCCOMMAND NAME="DENIEDRC "/>.

After share control has been established, the novice can stop share control at any time. If share control is stopped by the novice, the novice MUST send <RCCOMMAND NAME="TAKECONTROL"/>.

If the expert wants to end the control, the expert can send the message <RCCOMMAND NAME="REMOTECTRLEND"/> to the novice to signal that it no longer wants to control the novice screen. The novice MUST disable share control in response to the <RCCOMMAND NAME="REMOTECTRLEND"/> message.

The expert may receive the message <RCCOMMAND NAME="ESCRC"/> indicating that the novice has terminated share control.

Below is an example of the expert requesting to share control, and the novice allowing it. After some indefinite time, the novice stops allowing control and signals this to the expert.

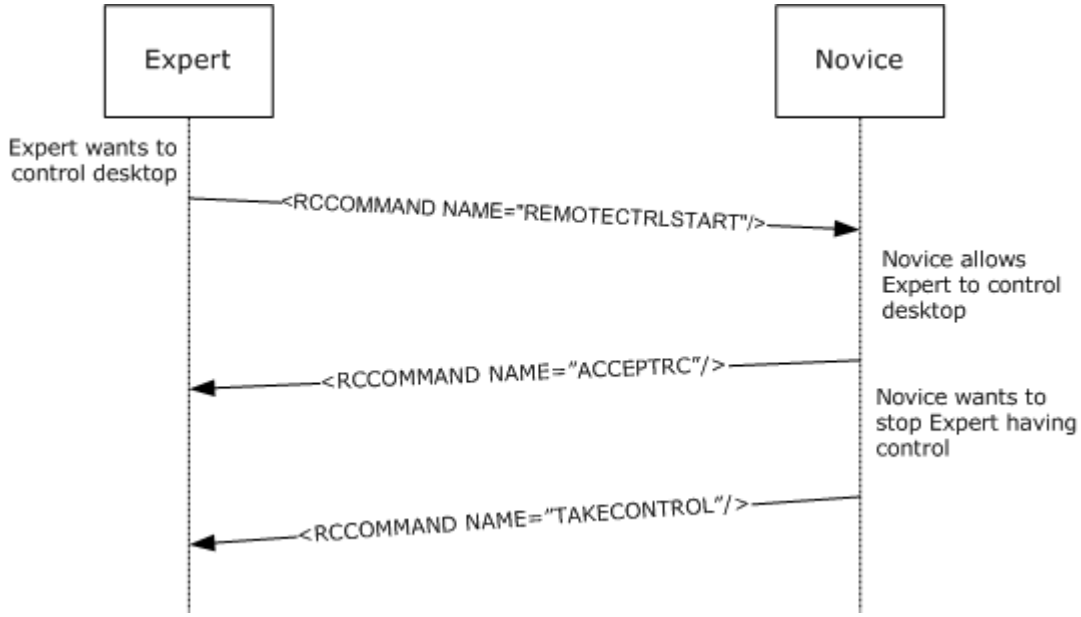


Figure 9: Remote control packet sequencing

3.7.6 Timer Events

There are no timer events associated with the Remote Assistance Protocol.

3.7.7 Other Local Events

There are no local events associated with the Remote Assistance Protocol.

3.8 Share Control Remote Assistance Novice (Server) Details

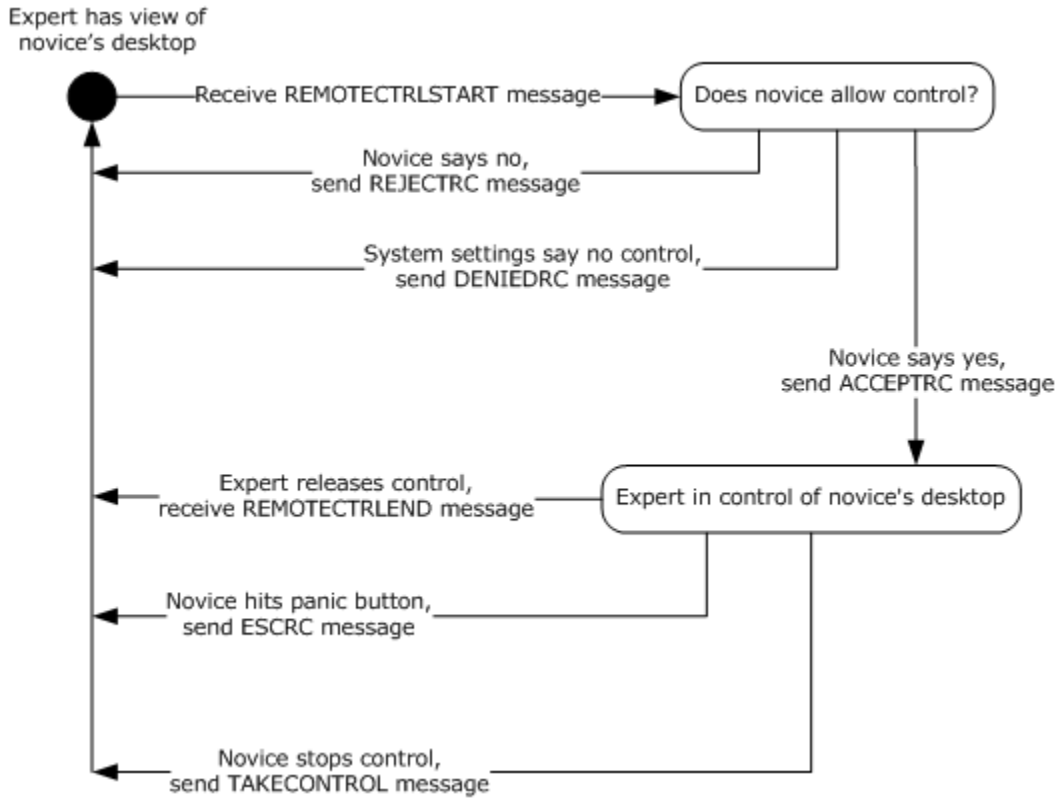


Figure 10: Desktop sharing session (novice/server perspective)

Normally, during the remote assistance connection, the expert can only observe what the novice is seeing on his or her desktop. If the expert wants to control the mouse and keyboard on the novice machine, the expert can request control from the novice. This portion of the Remote Assistance Protocol concerns the messages that are exchanged as permission for the expert is sought, granted, denied, and/or finally revoked or given up. The state machine shown above describes the messages involved in the exchange between expert and novice. Details provided in this section describe what is expected from the novice's point of view.

3.8.1 Abstract Data Model

The application that implements this portion of the Remote Assistance Protocol SHOULD track the current permissions granted to the expert to be able to process the received messages. Messages that fall outside of the state diagram shown above should be ignored. Processing messages out of the expected sequence MAY lead to incorrect event reporting or unanswered requests from the expert.

3.8.2 Timers

There are no timers associated with this portion of the Remote Assistance Protocol.

3.8.3 Initialization

The virtual channel used to send messages described in this section of the Remote Assistance Protocol (RCCOMMAND) is initialized when the remote assistance connection is first established. All RCCOMMAND messages are sent on the virtual channel named 71.

3.8.4 Higher-Layer Triggered Events

This portion of the Remote Assistance Protocol is not associated with any higher-layer triggered events.

3.8.5 Message Processing Events and Sequencing Rules

For the novice, there are several messages that are sent or can arrive during the entire process of requesting control (see section 2.2.2). When a message arrives, a string comparison MAY be used to determine the type of message that has arrived. The state machine shown in section 3.8.5 shows the expected sequence of messages; any messages that arrive out of sequence SHOULD be ignored by the receiving side. Processing of messages that arrive out of expected sequence MAY lead to incorrect event notification. All messages sent and received in this portion of the Remote Assistance Protocol are sent on the virtual channel named 71.

If the expert wants to assume control of the novice's mouse and keyboard, it MUST send the message <RCCOMMAND NAME="REMOTECTRLSTART"/>. When the novice receives this message, this protocol provides for three different responses. If there is a system setting or group policy that states that experts MUST not control the novice screen, the novice MUST send the response <RCCOMMAND NAME="DENIEDRC "/>. If the novice does not want to allow the expert to control the screen, the novice MUST send the response <RCCOMMAND NAME="REJECTRC"/>. The messages are exclusive with the DENIEDRC message superseding the REJECTRC message. If the novice does not want to allow the expert to control the screen, and the system does not allow for control to be taken, the DENIEDRC message MUST be sent, and the REJECTRC message MUST NOT be sent. If the novice wants to allow the expert to have control of the screen, and the system settings do not deny the expert's request, the response <RCCOMMAND NAME="ACCEPTRC"/> MUST be sent. The novice is considered to be allowing the expert control of the screen at this point.

After receiving the ACCEPTRC message from the novice, the expert can expect two different messages from the novice, both of which signal that control has been ended by the novice. If the novice ended control by pressing the ESC key (remote assistance has the concept of a Panic Key, which is a key listened to system-wide that, when pressed, immediately revokes control. This key is implemented as the ESC key although any key can be chosen by the implementing application), the message <RCCOMMAND NAME="ESCRC"/> is received by the expert. If the novice wants to signal the end of control through any other means, the message <RCCOMMAND NAME="TAKECONTROL"/> is received by the expert. In either case, the expert is now considered to be only viewing the novice screen.

If the expert wants to end the control before receiving either of these messages, it can send the message <RCCOMMAND NAME="REMOTECTRLEND"/> to the novice to signal that the expert no longer wants to control the screen. After sending this message, the expert should now be considered to be only viewing the novice screen.

An example follows of the expert requesting to share control and the novice allowing it. After some indefinite time, the expert wants to stop controlling the novice screen and signals this to the novice.

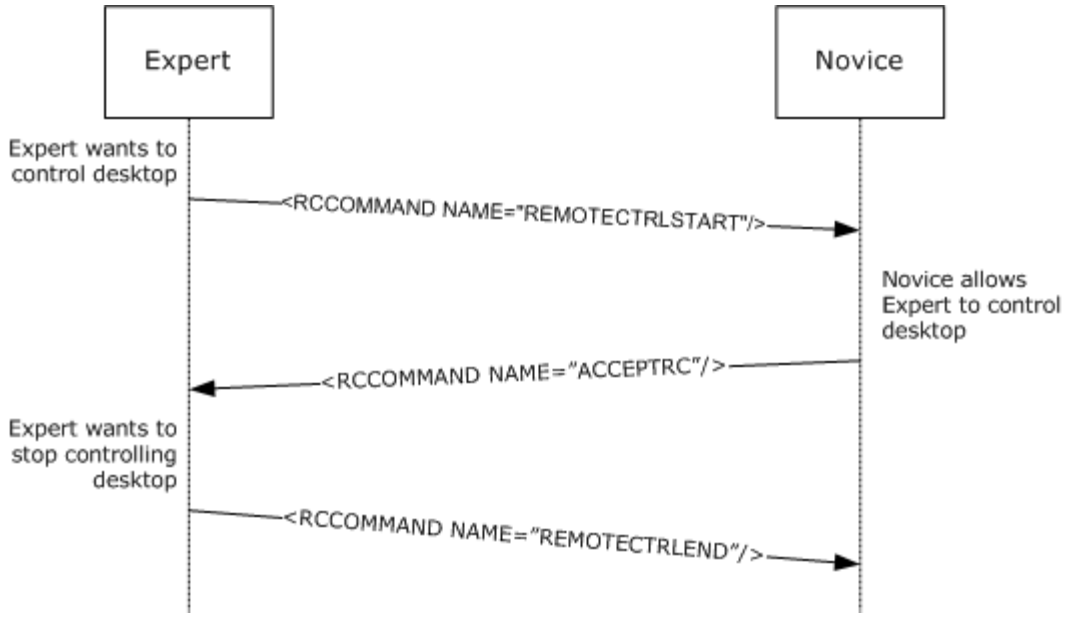


Figure 11: Expert-requested desktop control (in remote assistance session)

3.8.6 Timer Events

There are no timer events associated with this section of the Remote Assistance Protocol.

3.8.7 Other Local Events

There are no local events associated with this portion of the Remote Assistance Protocol.

3.9 Voice Expert (Client) Details

Voice communication while in a remote assistance connection is implemented using real-time communications (RTC) (for more information, see [MSDN-RTC]) to transmit and receive audio signal from the remote computer. The Remote Assistance Protocol has messages provided to initialize VoIP communication, to signal that VoIP is no longer wanted, and to coordinate voice quality or voice capability of the remote computer. The novice MUST act as the RTC server, and the message exchange is different depending on which side initially requested the VoIP communication because of this. A diagram follows detailing the message sequencing for initialization and tear down of the VoIP communication (showing both possibilities).<29>

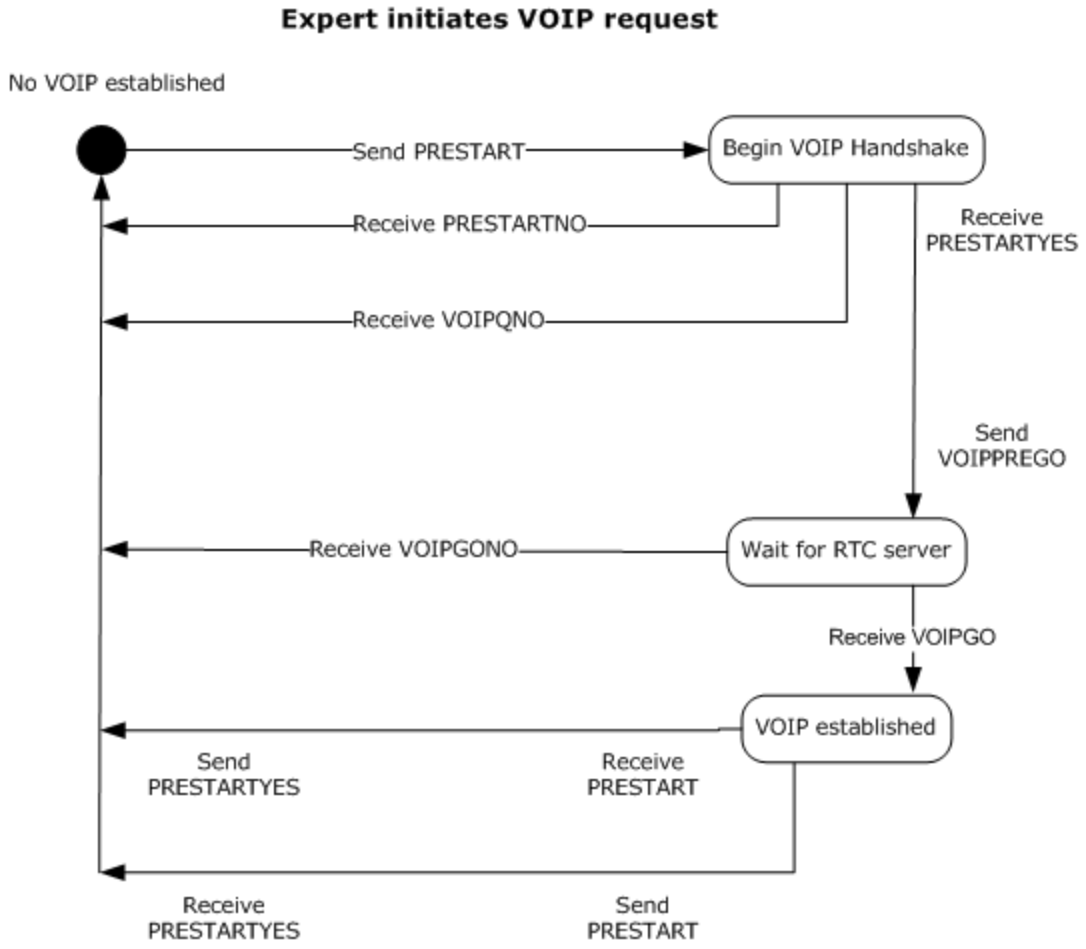


Figure 12: Remote assistance request (expert)

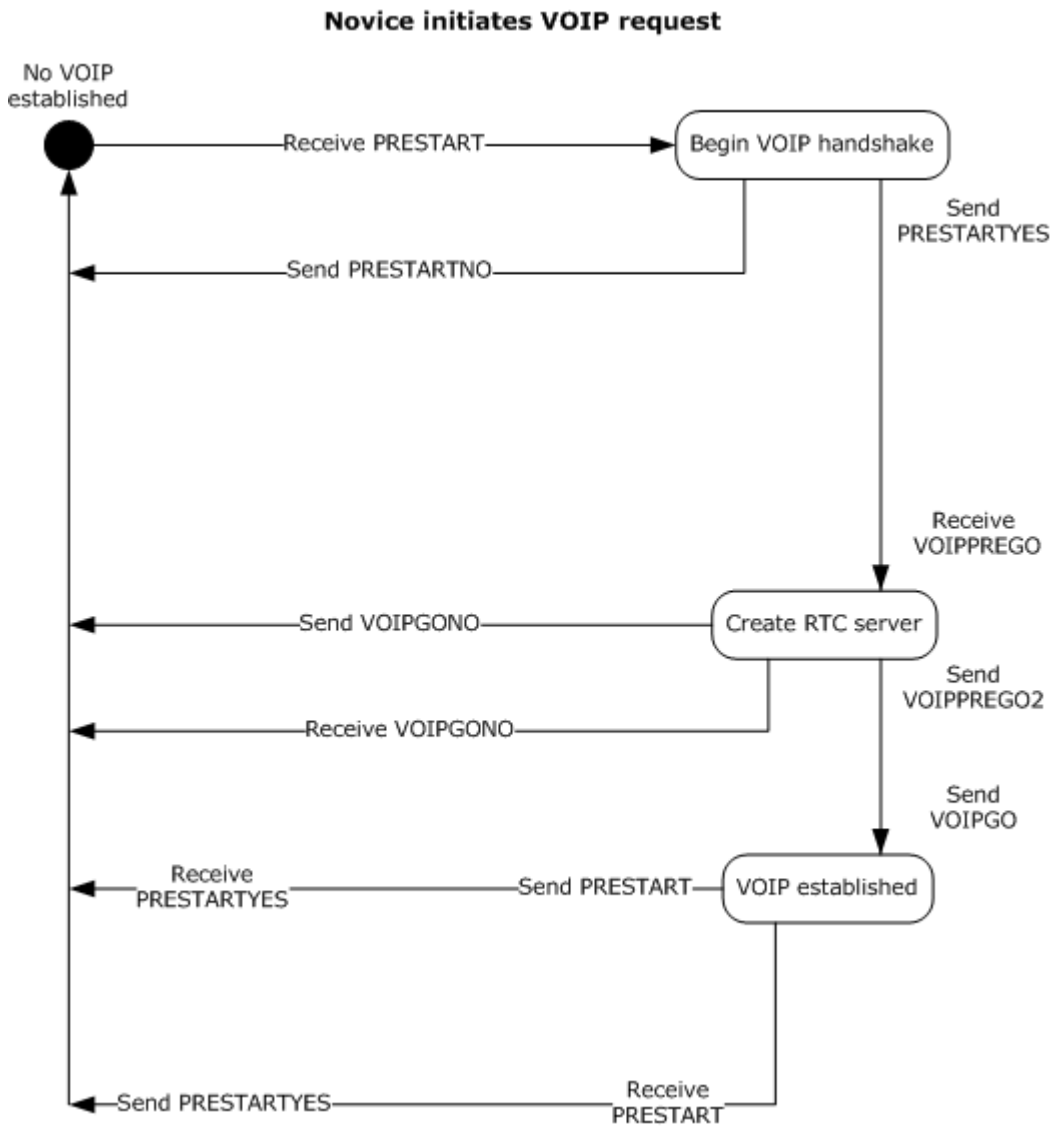


Figure 13: Remote assistance request (novice)

3.9.1 Abstract Data Model

An implementation of this portion of the Remote Assistance Protocol SHOULD maintain the VoIP connection status as it transitions from inactive to active, and then back to inactive again. The states MAY be represented by an enumeration and follow the states shown in the diagram in section 3.9.

3.9.2 Timers

There are no timers used in this portion of the Remote Assistance Protocol.

3.9.3 Initialization

Initialization of the virtual channel for VoIP messages is initialized when the remote assistance connection begins. All messages described in this section are sent on the virtual channel named 71 and follow the format shown in the section concerning RCCOMMAND.

3.9.4 Higher-Layer Triggered Events

The protocol does not make use of any higher-layer triggered events.

3.9.5 Message Processing Events and Sequencing Rules

The first category of messages deals with the quality of the voice transmission or the capability of the remote machine that has the hardware configured to make and receive audio signals. Real-time communications (RTC) allows for the bandwidth usage to be of a set sampling rate by calling the method put_MaxBitRate on the IRTCClient interface. RTC also has a method that can be called to check if the local computer has the capability to do VoIP communications, InvokeTuningWizard also on the IRTCClient interface.

The Remote Assistance Protocol allows for an application to signal a request to lower or raise the bandwidth used with the messages BANDWTOLOW and BANDWTOHIGH, respectively. When the message is received, the implementing application SHOULD set the MaxBitRate to 6,400 (when BANDWTOLOW is received) and SHOULD set the MaxBitRate to 64,000 (when BANDWTOHIGH is received). These messages MAY be sent if a lower or higher bandwidth usage is needed.

The Remote Assistance Protocol allows for an application to signal that the RTC wizard failed or succeeded when it checked for the hardware and drivers needed to do VoIP communications on the local machine. If the WIZARDBAD message is received, the receiving side SHOULD NOT attempt to initiate VoIP communication with the remote computer. If the WIZARDGOOD message is received, the receiver MAY attempt to initiate VoIP communications.

The second category of messages deals with the initialization of VoIP using real-time communications (RTC) (for more information, see [MSDN-RTC]). The novice MUST act as the RTC server. The messages exchanged validate that the request for voice communication is wanted by the other user, can be utilized by the remote system, and can provide the encryption key and IP address of the RTC server to the client. This message exchange is detailed in the diagrams shown in section 3.9.

The first message sent (if the expert initiated the request for VoIP) or received (if the novice initiated the request) is the PRESTART message. This message signals the expectation for voice communications. If VoIP is not wanted, the response to this message is VOIPQNO, and the exchange is considered complete. If VoIP is wanted by the receiver, the message PRESTARTYES is sent.

After receiving the PRESTARTYES message, the initiator of the VoIP request signals the capability and readiness to start VoIP communications by sending the VOIPPREGO message. If sent by the expert, it signals that the application is ready to use RTC to start VoIP communications. The expert SHOULD have access to the IRTCClient interface and have checked if the hardware has been tuned for VoIP use with a call to the IsTuned method on the IRTCClient interface before sending this message. If sent by the novice, it is a query to determine if the expert can use RTC for VoIP. If the expert receives the message VOIPPREGO, it SHOULD obtain a pointer to the IRTCClient interface and determine if the hardware has been tuned for VoIP use with a call to the IsTuned method. If the expert fails to do these things, the expert MUST send the message VOIPGO NO to the novice. If the expert succeeds, it MUST send the message VOIPPREGO2.

At this stage, the expert is waiting for the creation of the RTC server on the novice side and is waiting for a message from the novice. If the expert receives VOIPGO NO, it signals that the

creation of the RTC server failed. If the expert receives VOIPGO, it signals that the RTC server has been successfully created, and the expert should now connect. The VOIPGO message has two attributes used to make the connection, the key used to encrypt the data being sent between the two machines, and the IP list that the novice is listening on (see sections 2.2.2.11 and 2.2.2.12, respectively). Using the PC to PC call model provided by RTC, the expert connects to the novice through RTC and can now send and receive audio data.

When either side wants to end the VoIP communications, the message PRESTART MUST be sent. When this message is received, and VoIP is already established, the receiver SHOULD clean up the RTC objects it has reference to and SHOULD send the PRESTARTYES message when finished.

A diagram follows showing the messages exchanged while setting up and cleaning up after a VoIP session.

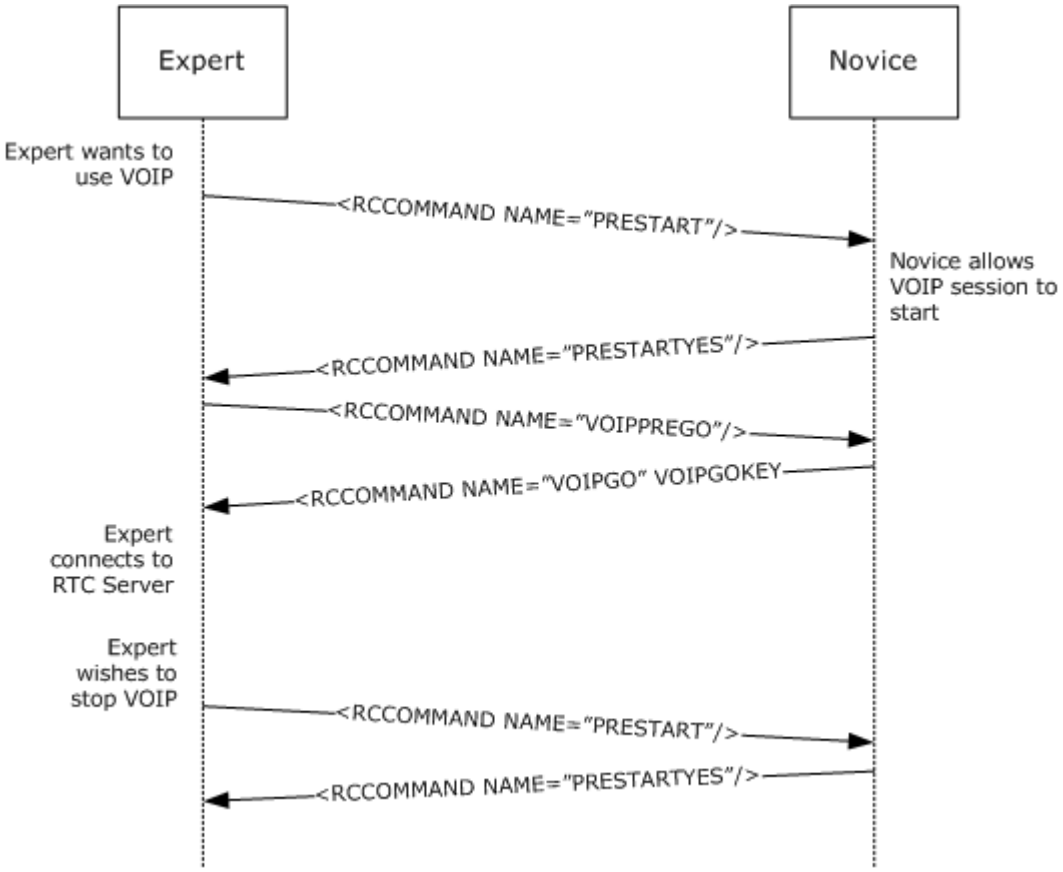


Figure 14: Remote assistance VoIP session message exchange

3.9.6 Timer Events

This section of the Remote Assistance Protocol has no timer events.

3.9.7 Other Local Events

The Remote Assistance Protocol has no interaction with other local events.

3.10 Voice Novice (Server) Details

Voice communication while in a remote assistance connection is implemented using RTC to transmit and receive audio signals from the remote computer. The Remote Assistance Protocol has messages provided to initialize VoIP communication, to signal that VoIP is no longer wanted, and to coordinate voice quality or voice capability of the remote computer. The novice MUST act as the RTC server, and the message exchange is different depending on which side initially requested the VoIP communication because of this. A diagram follows detailing the message sequencing for initialization and tear down of the VoIP communication showing both possibilities considered from the novice's point of view.

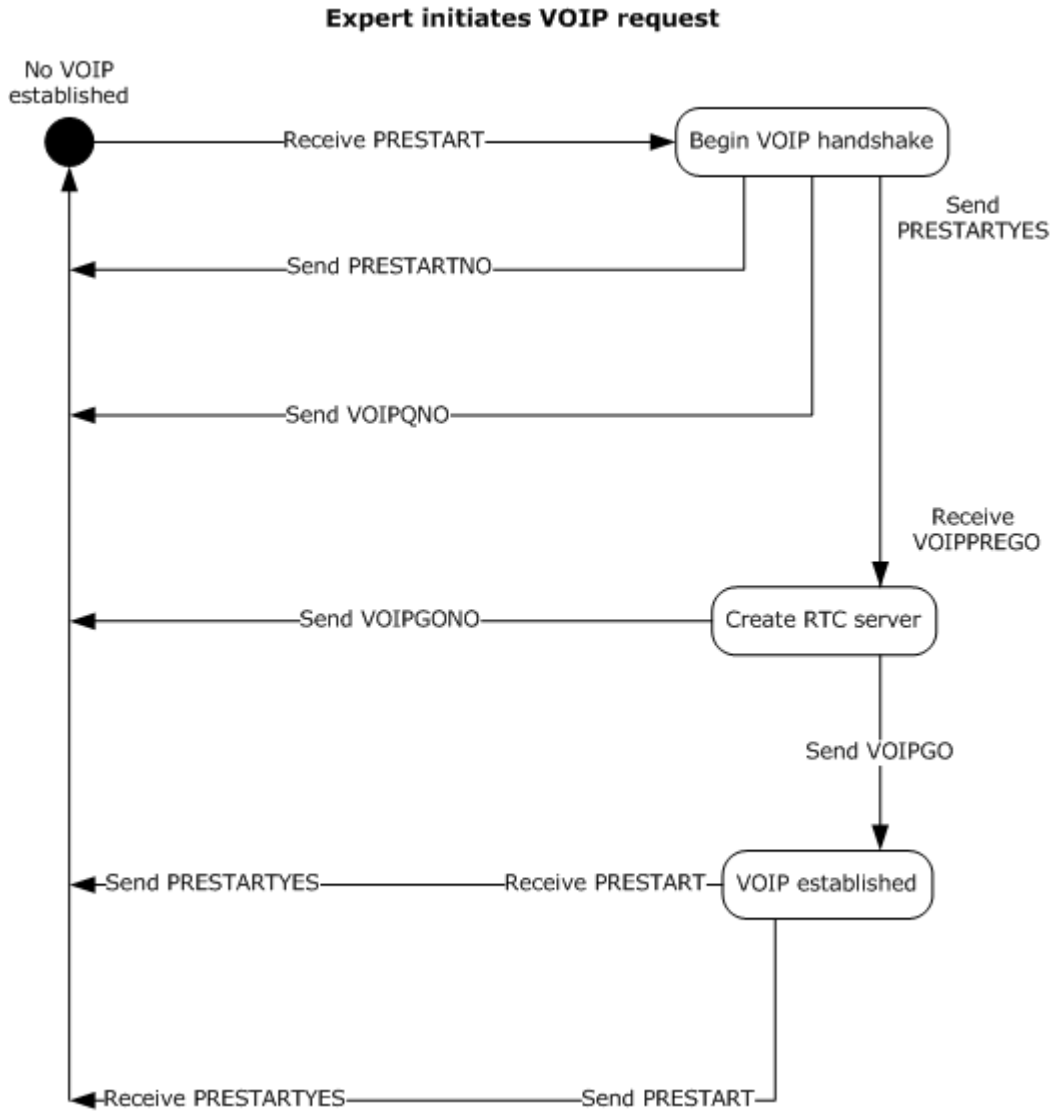


Figure 15: Remote assistance session diagram (initiated by expert/client)

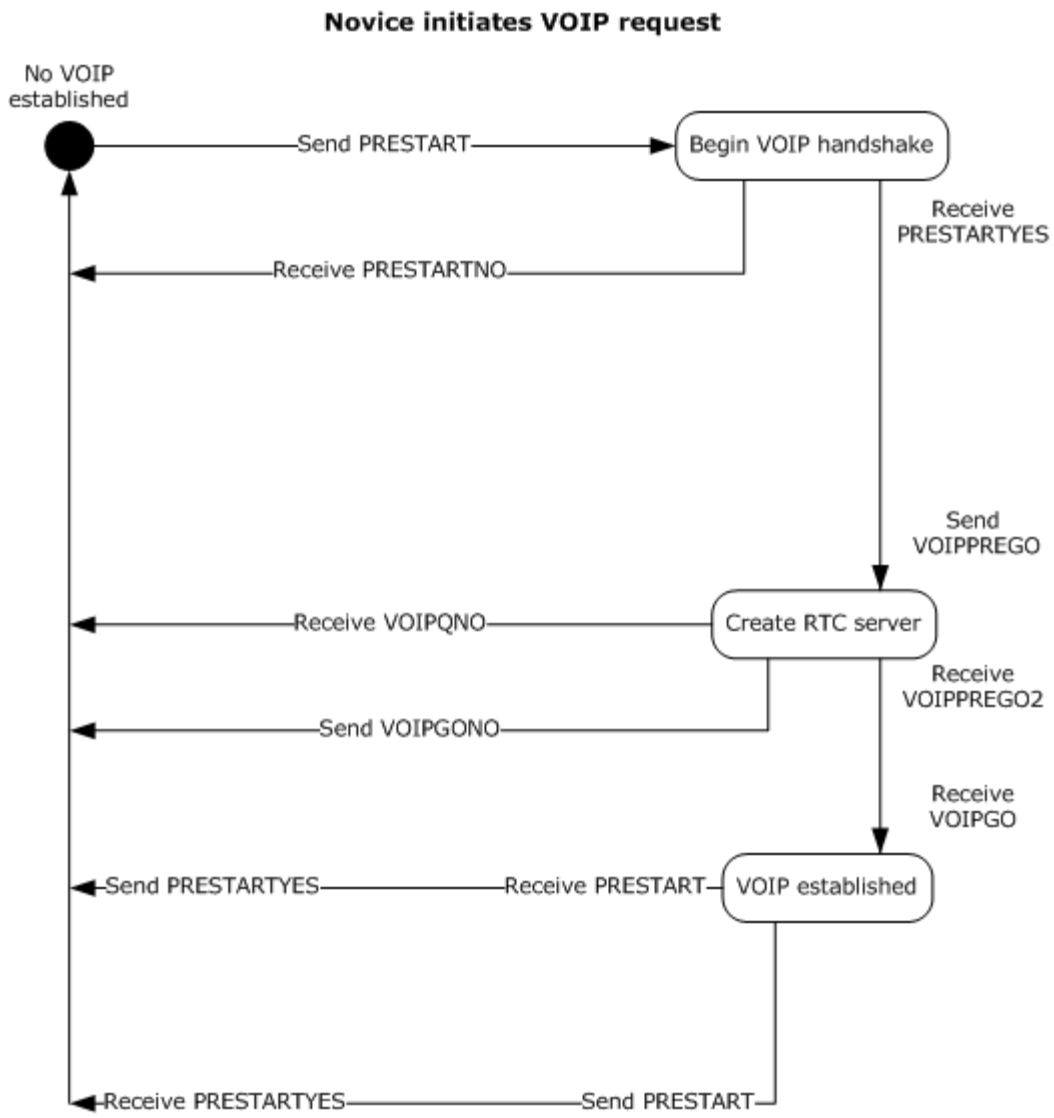


Figure 16: Remote assistance session diagram (initiated by novice/server)

3.10.1 Abstract Data Model

An implementation of this portion of the Remote Assistance Protocol SHOULD maintain the VoIP connection status as it transitions from inactive to active, and then back to inactive again. The states MAY be represented by an enumeration and follow the states shown in the diagram in section 3.10.

3.10.2 Timers

There are no timers used in this portion of the Remote Assistance Protocol.

3.10.3 Initialization

Initialization of the virtual channel for VoIP messages is initialized when the remote assistance connection begins. All messages described in this section are sent on the virtual channel named 71 and follow the format shown in the section concerning RCCOMMAND.

3.10.4 Higher-Layer Triggered Events

This portion of the Remote Assistance Protocol does not utilize any external higher-layer events.

3.10.5 Message Processing Events and Sequencing Rules

The first category of messages deals with the quality of the voice transmission or the capability of the remote machine that has the hardware configured to make and receive audio signals. RTC allows for the bandwidth usage to be of a set sampling rate by calling the method put_MaxBitRate on the IRTCClient interface. RTC also has a method that can be called to check if the local computer has the capability to do VoIP communications, InvokeTuningWizard also on the IRTCClient interface.

The Remote Assistance Protocol allows for an application to signal a request to lower or raise the bandwidth used with the messages BANDWTOLOW and BANDWTOHIGH, respectively. When the message is received, the implementing application SHOULD set the MaxBitRate to 6,400 (when BANDWTOLOW is received) and SHOULD set the MaxBitRate to 64,000 (when BANDWTOHIGH is received). These messages MAY be sent if a lower or higher bandwidth usage is needed.

The Remote Assistance Protocol allows for an application to signal that the RTC wizard failed or succeeded when it checked for the hardware and drivers needed to do VoIP communications on the local machine. If the WIZARDBAD message is received, the receiving side SHOULD NOT attempt to initiate VoIP communication with the remote computer. If the WIZARDGOOD message is received, the receiver MAY attempt to initiate VoIP communications.

The second category of messages deals with the initialization of VoIP using RTC. The novice MUST act as the RTC server. The messages exchanged validate that the request for voice communication is wanted by the other user, can be utilized by the remote system, and can provide the encryption key and IP address of the RTC server to the client. This message exchange is detailed in the diagrams shown in section 3.10.

The first message sent (if the expert initiated the request for VoIP) or received (if the novice initiated the request) is the PRESTART message. This message signals the expectation for voice communications. If VoIP is not wanted, the response to this message is VOIPQNO, and the exchange is considered complete. If VoIP is wanted by the receiver, the message PRESTARTYES is sent.

After receiving the PRESTARTYES message, the initiator of the VoIP request signals the capability and readiness to start VoIP communications by sending the VOIPPREGO message. If sent by the expert, it signals that the application is ready to use RTC to start VoIP communications. The expert SHOULD have access to the IRTCClient interface and have checked if the hardware has been tuned for VoIP use with a call to the IsTuned method on the IRTCClient interface before sending this message. If sent by the novice, it is a query to determine if the expert can use RTC for VoIP. If the expert receives the message VOIPPREGO, it SHOULD obtain a pointer to the IRTCClient interface and determine if the hardware has been tuned for VoIP use with a call to the IsTuned method. If the expert fails to do these things, the expert MUST send the message VOIPGO NO to the novice. If the expert succeeds, it MUST send the message VOIPPREGO2.

At this stage, the expert is waiting for the creation of the RTC server on the novice side and is waiting for a message from the novice. If the expert receives VOIPGO NO, it signals that the creation of the RTC server failed. If the expert receives VOIPGO, it signals that the RTC server has been successfully created, and the expert should now connect. The VOIPGO message has two

attributes used to make the connection, the key used to encrypt the data being sent between the two machines and the IP list that the novice is monitoring on (see sections 2.2.2.11 and 2.2.2.12, respectively). Using the PC to PC call model provided by RTC, the expert connects to the novice through RTC and can now send and receive audio data.

When either side wants to end the VoIP communications, the message PRESTART MUST be sent. When this message is received, and VoIP is already established, the receiver SHOULD clean up the RTC objects it has reference to and SHOULD send the PRESTARTYES message when finished.

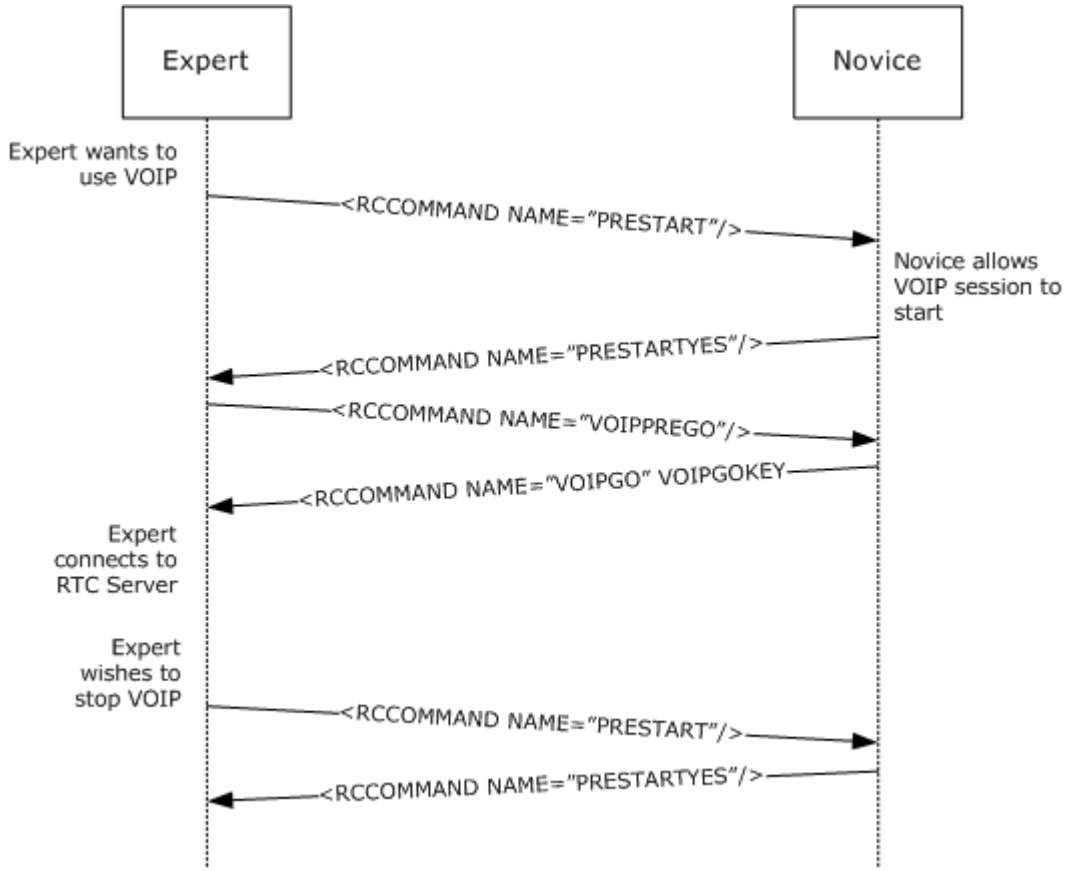


Figure 17: Remote assistance VoIP session initialization sequence

3.10.6 Timer Events

This section of the Remote Assistance Protocol has no timer events.

3.10.7 Other Local Events

The Remote Assistance Protocol has no interaction with other local events.

4 Protocol Examples

The following sections provide examples of how the Remote Assistance Protocol operates in common scenarios.

The RCCOMMAND message is used in VoIP initialization, session control synchronization, and file transfer initialization. For the full extent of messages that can be sent in the RCCOMMAND format, see section 2.2.2. Sample messages are shown with scenarios of where the message would be sent or received.

4.1 Example of a VOIPGO Message

The last message that the novice (acting as the RTC server) sends to the expert (acting as the client) is the IP and encryption key to use when making the RTC connection. The message MUST be a null-terminated Unicode string. The following is an example of a valid VOIPGO message:

[XML]

```
<RCCOMMAND NAME="VOIPGO" VOIPVER="VOIPVER2"
VOIPGOKEY="NzaogjS5hQMun/saZ1YCBMT9GwrdJwOomrldiOmXTrE="
VOIPIPLIST="172.31.242.5:11334"/>
```

The RCCOMMAND message is formed as an XML element and has several attributes. The NAME attribute specifies what kind of message this is. The VOIPVER attribute SHOULD always be set to VOIPVER. The VOIPGOKEY attribute is set by the server (novice) for use as an encryption key. The VOIPIPLIST shows one IP address and port that the client (expert) can try to connect on.

4.2 Example of a FILEXFER Message

The first message the file sender sends is the FILEXFER message. This message contains information on the file to be sent such as original filename (so it can be suggested on the receiving side as the filename to save as) and byte size (so the remaining data to be transferred can be displayed to the user). The message MUST be a null-terminated Unicode string. The following is an example of a valid FILEXFER message:

[XML]

```
<RCCOMMAND NAME="FILEXFER" FILENAME="20070130182140.xml"
FILESIZE="436" CHANNELID="RA_FX"/>
```

This message is formed as an XML element and has several attributes. The NAME attribute specifies what kind of message this is. The FILENAME attribute is set to the recommended filename for the recipient, the original name of the file being copied. The FILESIZE attribute is set to the size in bytes of the file being copied. The CHANNELID attribute is set to the name of the virtual channel that the rest of the exchange for this file transfer takes place on.

5 Security

The following sections specify security considerations for implementers of the Remote Assistance Protocol.

5.1 Security Considerations for Implementers

There are no security considerations for implementers of the Remote Assistance Protocol.

5.2 Index of Security Parameters

There are no security parameters for the Remote Assistance Protocol.

6 Appendix A: Windows Behavior

The information in this specification is applicable to the following versions of Windows:

- Windows Server 2003
- Windows XP
- Windows Vista

Exceptions, if any, are noted below. Unless otherwise specified, any statement of optional behavior in this specification prescribed using the terms SHOULD or SHOULD NOT implies Windows behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that Windows does not follow the prescription.

<1> Section 1.3.5: VoIP is supported in Windows XP. VoIP is not supported in Windows Vista.

<2> Section 1.7: VoIP capability is only supported in Windows XP. Windows Vista is not capable of VoIP, and rejects a VoIP initiation request from Windows XP.

<3> Section 2.2.1.4: The REMOTEDESKTOP_CTL_AUTHENTICATE_PACKET is used only when the novice or expert is Windows XP or Windows Server 2003.

<4> Section 2.2.1.4: "NAME" and "PASS" are two properties used in **expertBlob**. The PASS property is used when the Remote Assistance Invitation File is protected by a password on Windows XP or Windows Server 2003, or when a Windows XP or Windows Server 2003 expert is making a connection with a Remote Assistance Invitation File. The PASS property value is a string that contains the result of encrypting the PassStub in the Remote Assistance Invitation File with the password. For more information, see [MS-RAI] section 6.

An example of an **expertBlob** with the PASS property

```
9;NAME=John69;PASS=
 D4B4277E9E9D06CFC19BB23FD869B5E0C99B0908280407A6E2EEC43F98035F7D
```

<5> Section 2.2.1.6: This is true only in the cases of Windows XP and Windows Server 2003.

<6> Section 2.2.1.9: This packet is sent only in the case of Windows XP and Windows Server 2003.

<7> Section 2.2.1.11: This packet is applicable only for Windows Vista.

<8> Section 2.2.1.12: The REMOTEDESKTOP_CTL_VISTA_EXPERT_PACKET is a Windows Vista-only message.

<9> Section 2.2.1.13: The REMOTEDESKTOP_CTL_RANOVICE_NAME_PACKET is a Windows Vista-only message. When the expert sends the password hash, the expert name is sent before sending over the password. When the novice receives the expert's name, the novice responds with the novice name packet. The name exchange occurs before the password verification, to enable the novice to decide, based on a dialog displaying the expert's name, whether to allow the expert to connect. At any time during the session, if the expert sends the packet announcing the expert name, the expert name is updated in memory on the novice's machine. Similarly, if the novice name is re-sent, the novice name is updated in memory on the expert's machine.

<10> Section 2.2.1.14: The REMOTEDESKTOP_CTL_RAEXPERT_NAME_PACKET is a Windows Vista-only message. When the expert sends the password hash, the expert name is sent before sending over the password. When the novice receives the expert's name, the novice responds with the

novice name packet. The name exchange occurs before the password verification, to enable the novice to decide, based on a dialog displaying the expert's name, whether to allow the expert to connect. At any time during the session, if the expert sends the packet announcing the expert name, the expert name is updated in memory on the novice's machine. Similarly, if the novice name is re-sent, the novice name is updated in memory on the expert's machine.

<11> Section 2.2.2.1: This message is only sent by clients on Windows XP or Windows Server 2003, Windows Vista and Windows Server 2008 use collaborative events to notify the server of a remote control request.

<12> Section 2.2.2.1: This value is only sent by Windows XP and Windows Server 2003.

<13> Section 2.2.2.4: For Windows XP/protocol_rfc_ms-ra3_1.xml protocol_rfc_ms-ra3_2.xml protocol_rfc_ms-ra3_3_3.xmlWindows Server 2003, in the case where the novice started the file transfer, the name will be the IP address of sender followed by the character '.', followed by the number of seconds since January 1, 1970. In the case where the expert started the file transfer, the name will be the characters '1000.' followed by the number of seconds since January 1, 1970.

For Windows Vista, regardless of which role started the transfer, the name will be 'RA_FX'.

<14> Section 3.1: The REMOTEDESKTOP_CTL_ISCONNECTED_PACKET is only sent and received between Windows XP and Windows Server 2003. If a Windows Vista machine is the expert, the REMOTEDESKTOP_CTL_ISCONNECTED_PACKET is not sent.

<15> Section 3.1.5: This is the case in which the novice is on Windows XP or Windows Server 2003.

<16> Section 3.1.5: This is the case in which either the novice or the expert is on Windows XP or Windows Server 2003.

<17> Section 3.1.5: This is the case in which both novice and expert are on Windows Vista.

<18> Section 3.1.5: This is the case in which either the novice or the expert is on Windows XP or Windows Server 2003.

<19> Section 3.1.5: This is the case in which both the novice and the expert are on Windows Vista.

<20> Section 3.1.5: This is true only in the cases of Windows XP and Windows Server 2003.

<21> Section 3.1.6: The REMOTEDESKTOP_CTL_ISCONNECTED packet is only sent by Windows XP and Windows Server 2003. Windows Vista does not send this packet.

<22> Section 3.2.5: This is the case in which either the novice or the expert is on Windows XP or Windows Server 2003.

<23> Section 3.2.5: This is the case in which both novice and expert are on Windows Vista.

<24> Section 3.2.5: This is the case in which either the novice or the expert is on Windows XP or Windows Server 2003.

<25> Section 3.2.5: This is the case in which both the novice and the expert are on Windows Vista.

<26> Section 3.2.5: This is true only in the cases of Windows XP and Windows Server 2003.

<27> Section 3.3.5: A file SHOULD be broken into blocks of 1024 bytes when transferring to a Windows Vista machine, although longer blocks MAY be received without error. When transferring to a Windows XP or Windows Server 2003 machine, a file MUST be broken into blocks of 409600 bytes (for all blocks except the last) to avoid errors in the transfer process.

<28> Section 3.4.5: A file SHOULD be broken into blocks of 1024 bytes when transferring to a Windows Vista machine, although longer blocks MAY be received without error. When transferring to a Windows XP or Windows Server 2003 machine, a file MUST be broken into blocks of 409600 bytes (for all blocks except the last) to avoid errors in the transfer process.

<29> Section 3.9: Remote assistance in Windows XP and Windows Server 2003 (x86 only) offers the feature of using a speaker and microphone to do voice communication while in a remote assistance connection. 64-bit implementations of Windows XP and Windows Server 2003 as well as all versions of Windows Vista do not offer this feature. 64 bit implementations of Windows XP and Windows Server 2003 always respond to the initial message for voice communications as if the hardware configuration does not allow for voice communications (RCCOMMAND NAME="DISABLEVOICE"). Windows Vista does not respond to the voice request at all.

7 Index

L

M

N

O

P

R

S

T

V

W