ETSI TS 103 544-2 V1.3.1 (2019-10)

Publicly Available Specification (PAS); Intelligent Transport Systems (ITS); MirrorLink®; Part 2: Virtual Network Computing (VNC) based Display and Control

CAUTION

The present document has been submitted to ETSI as a PAS produced by CCC and approved by the ETSI Technical Committee Intelligent Transport Systems (ITS).

Reference

RTS/ITS-98-2

Keywords

interface, ITS, PAS, smartphone, USB

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

The present document can be downloaded from:
http://www.etsi.org/standards-search

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format at www.etsi.org/deliver.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at
https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx

If you find errors in the present document, please send your comment to one of the following services:
https://portal.etsi.org/People/CommiteeSupportStaff.aspx

Copyright Notification

Contents

Intellectual Property Rights 5
Foreword 5
Modal verbs terminology 5
1 Scope 6
2 References 6
2.1 Normative references 6
2.2 Informative references 7
3 Definition of terms, symbols and abbreviations 7
3.1 Terms 7
3.2 Symbols 7
3.3 Abbreviations 7
4 Introduction 8
5 Managing a VNC session 9
5.1 Identifying Remote Applications and the VNC Server 9
5.2 Launching the VNC Session 9
5.3 Intentionally Terminating the VNC Session 9
5.4 Unintentionally Terminating the VNC Session 10
5.5 Testing Considerations 10
6 Traditional VNC Protocol Phases 10
6.1 General 10
6.2 Handshaking Phase 11
6.3 Initialization Phase 12
6.4 Framebuffer Update and Event Phase 13
7 VNC MirrorLink Extension Messages 16
7.1 General 16
7.2 ByeBye Message 17
7.3 Display Configuration Messages 18
7.3.1 General 18
7.3.2 Framebuffer Scaling (VNC Server) 22
7.3.3 Framebuffer Scaling (VNC Client) 23
7.3.4 Handling of Different Framebuffer Aspect Ratios 23
7.3.5 Handling of Server Pixel Aspect Ratios 24
7.3.6 Handling of Application, Framebuffer and Display Orientation 24
7.4 Event Configuration Messages 25
7.5 Event Mapping Messages 29
7.6 Device Status Messages 30
7.7 Content Attestation Messages 34
7.8 Framebuffer Blocking Notification 37
7.9 Audio Blocking Notification 43
7.10 Touch Event 46
8 Additional Encodings and Pseudo Encodings 48
8.1 General 48
8.2 MirrorLink Pseudo Encoding 49
8.3 Context Information Pseudo Encoding 49
8.4 Desktop Size Pseudo Encoding 50
8.5 Scan Line based Run-Length Encoding 51
8.6 VA H.264 Encoding 52
8.6.1 Overview 52
8.6.2 Theory of operation 53
Annex A (normative): Knob Configuration 59

Annex B (normative): Key Event Mapping 60

Annex C (normative): Language Sets 63

C.1 Basic Set Latin-1 63

Annex D (informative): Authors and Contributors 64

History 65

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee Intelligent Transport Systems (ITS).

The present document is part 2 of a multi-part deliverable. Full details of the entire series can be found in part 1 [i.1].

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in Clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

1 Scope

The present document is part of the MirrorLink® specification which specifies an interface for enabling remote user interaction of a mobile device via another device. The present document is written having a vehicle head-unit to interact with the mobile device in mind, but it will similarly apply for other devices, which provide a colour display, audio input/output and user input mechanisms.

The contents of the MirrorLink Server device's screen are transferred to the MirrorLink Client device. The control inputs are transferred from the MirrorLink Client to the MirrorLink Server. Screen copy methods can be used to copy the content of the MirrorLink Server's framebuffer to the MirrorLink Client's display. The copy operation can include rotation or colour conversion. The frame buffer is used as an abstraction layer, allowing any changes to the applications and services running on the mobile device to be avoided. For this purpose, the Virtual Networking Computing (VNC) protocol is used.

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at https://docbox.etsi.org/Reference.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long-term validity.

The following referenced documents are necessary for the application of the present document.

[1] IETF RFC 6143: "The Remote Framebuffer Protocol", March 2011.

NOTE: Available at https://tools.ietf.org/html/rfc6143.

[2] Bluetooth® Specification: "Hands-free Profile", Audio, Telephony, and Automotive Working Group, Revision 1.7.2, January 21, 2019

NOTE: Available at https://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=457090.

[3] ETSI TS 103 544-3 (V1.3.1): "Publicly Available Specification (PAS); Intelligent Transport Systems (ITS); MirrorLink®; Part 3: Audio".

[4] ETSI TS 103 544-9 (V1.3.1): "Publicly Available Specification (PAS); Intelligent Transport Systems (ITS); MirrorLink®; Part 9: UPnP Application Server Service".

[5] ETSI TS 103 544-10 (V1.3.1): "Publicly Available Specification (PAS); Intelligent Transport Systems (ITS); MirrorLink®; Part 10: UPnP Client Profile Service".

[6] ETSI TS 103 544-26 (V1.3.1): "Publicly Available Specification (PAS); Intelligent Transport Systems (ITS); MirrorLink® ; Part 26: Consumer Experience Principles and Basic Features".

[7] X Consortium Standard: "X Window System Protocol", X Version 11, Release 6.9/7.0.

NOTE: Available at ftp://ftp.x.org/pub/X11R7.0/doc/PDF/proto.pdf.

[8] Recommendation ITU-T H.264 (04-2017): "Advanced video coding for generic audiovisual services".

NOTE: Available at https://www.itu.int/rec/T-REC-H.264-201704-S/en.

[9] ISO 639-1: "Codes for the representation of names of languages -- Part 1: Alpha-2 code".

[10] ISO 3166-1: "Codes for the representation of names of countries and their subdivisions -- Part 1: Country codes".

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long-term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] ETSI TS 103 544-1 (V1.3.1): "Publicly Available Specification (PAS); Intelligent Transport Systems (ITS); MirrorLink®; Part 1: Connectivity".

[i.2] US Department of Homeland Security: "Emergency Alerts".

NOTE: Available at https://www.ready.gov/alerts.

[i.3] IANA, Protocol Registries, Remote Framebuffer (RFB) assignments.

NOTE: Available at https://www.iana.org/assignments/rfb/rfb.xhtml.

[i.4] ISO/IEC 10646:2014: "Information technology -- Universal Coded Character Set (UCS)".

[i.5] ETSI TS 103 544-22 (V1.3.1): "Publicly Available Specification (PAS); Intelligent Transport Systems (ITS); MirrorLink®; Part 22: Android Specific Specifications enabling AIDL-based MirrorLink® Applications".

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the following terms apply:

pointer event: touch screen action in which the user touches the screen with one (virtual) finger only at a single location

touch event: touch screen action in which the user touches the screen with two or more separate fingers at different locations

NOTE: touch events are used to describe more complex touch action, like pinch-open or pinch-close

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

A2DP	Bluetooth Advanced Audio Distribution Profile
API	Application Programming Interface
ARGB	Alpha-Red-Green-Blue
BT	Bluetooth
BVRA	Bluetooth Voice Recognition Activation

DAP	Device Attestation Protocol
FAR	Framebuffer Aspect Ratio
FBU	FrameBuffer Unit
FEMA	Federal Emergency Management Agency
HSML	High-Spead Media Link
IANA	Internet Assigned Numbers Authority
ISO	International Standards Organization
NAL	Network Abstraction Layer
PAR	Pixel Aspect Ratio
PIN	Personal Identification Number
PPS	Picture Parameter Set
RFB	Remote Framebuffer
RGB	Red-Green-Blue
RLE	Run-Length Encoding
RTP	Real-time Transport Protocol
SPS	Sequence Parameter Set
TCP	Transmission Control Protocol
TV	TeleVision
UDP	User Datagram Protocol
UI	User Interface
UPnP	Universal Plug and Play
URL	Universal Resource Locator
US	United States
USB	Universal Serial Bus
UTF	Unicode Transformation Format
VA	Video Audio
VCL	Video Coding Layer
VNC	Virtual Network Computing

4 Introduction

The Virtual Networking Computing (VNC) uses the Remote Framebuffer Protocol (RFB) as a simple protocol for remote access to any sort of framebuffer-based user interface. The remote endpoint is called the VNC Client, whereas the endpoint driving the framebuffer is called the VNC Server. In the MirrorLink context, the VNC Client resides in the vehicle head-unit (MirrorLink Client) and the VNC Server is in the mobile device (MirrorLink Server). The VNC Client will show the remote display either on the entire local display or on a subset of it, as shown in Figure 1.

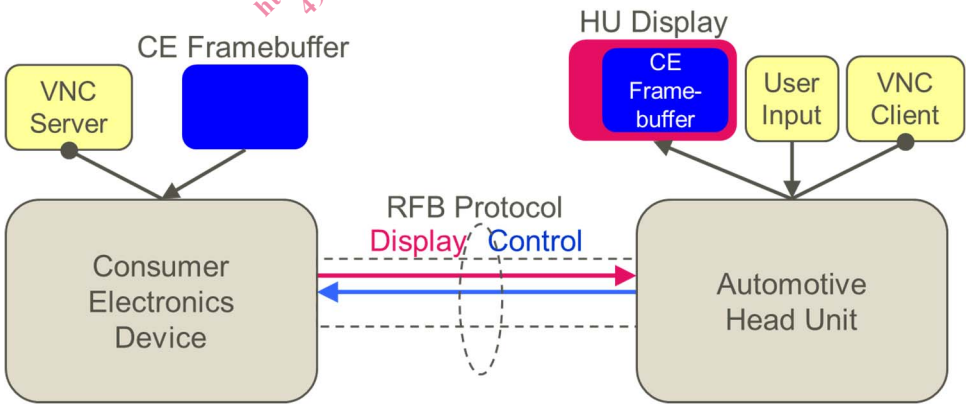

Figure 1: MirrorLink VNC Setup

The command and control input is handled as part of the VNC protocol by key and pointer events. A key or pointer event on the MirrorLink Client will be signalled to the MirrorLink Server via a specific key symbol value, which uniquely identifies the event. The mobile device and/or its application will not necessarily support all possible keys defined. Some applications may even have a dynamic behaviour on the selection of key inputs they expect.

The RFB protocol originates from the desktop computing world and has been designed as a thin client protocol, i.e. it assumes a VNC Client with only a few requirements, and a VNC Server having access to more processing capabilities.

The protocol allows the VNC Client to be as simple as possible. In the MirrorLink context this assumption needs to be reconsidered, as mobile devices are experiencing performance limitations as well.

The MirrorLink Client shall implement the VNC Client functionality.

The MirrorLink Server shall implement the VNC Server functionality.

5 Managing a VNC session

5.1 Identifying Remote Applications and the VNC Server

The identification of remote VNC based applications and of the VNC Server is described in [4].

5.2 Launching the VNC Session

The VNC Server start-up is automatically facilitated via UPnP. The MirrorLink Server's VNC Server shall be running, when launching any application in response to a UPnP TmApplicationServer:1 service *LaunchApplication* action, as defined in [4]. The *LaunchApplication* action shall return with a URL to the VNC Server, hosting the VNC session.

If the returned URL is already used from any established VNC session, this session will continue without any change.

Otherwise a new VNC session shall be established, given the following steps:

1) VNC Server shall listen for the VNC Client to make TCP connection at the provided URL.

2) VNC Client shall make a TCP connection to the provided URL.

3) VNC Server and Client shall initialize the VNC session according to the VNC protocol.

5.3 Intentionally Terminating the VNC Session

A VNC session can be terminated any time from both the VNC Server and Client. This mechanism is not meant to handle error situations, which require immediate action from both the VNC Server and Client (use the unintentional termination mechanisms instead). In particular, intentional termination shall be used, if the VNC session is terminated based on an intended user action resulting in a termination of the VNC session.

The VNC Client shall terminate a VNC session using the VNC *ByeBye* message, as given below:

1) VNC Client shall send a VNC *ByeBye* message. The VNC Client shall not send any further VNC message, after sending the VNC *ByeBye* message. The VNC Client should ignore all incoming VNC messages, after sending a VNC *ByeBye* message.

2) VNC Server shall respond with a VNC *ByeBye* message.

3) VNC Client shall disconnect the TCP connection. The VNC Client should disconnect the TCP connection, if it does not receive a VNC *ByeBye* message back within 5 s.

4) VNC Server should disconnect the TCP connection on detection of the VNC Client TCP disconnect or 5 s after sending the VNC *ByeBye* message, whatever comes first.

The VNC Client shall wait with the VNC *ByeBye* message, if it has an outstanding UPnP *LaunchApplication* action until it has received the corresponding UPnP response. This avoids any potential race condition problems.

The VNC Server shall terminate a VNC session, using the VNC *ByeBye* message, as given below:

1) VNC Server shall send a VNC *ByeBye* message. The VNC Server shall not send any further VNC message, after sending the VNC *ByeBye* message. The VNC Server should ignore all incoming VNC messages, after sending a VNC *ByeBye* message.

2) VNC Client shall disconnect the TCP connection.

3) VNC Server should disconnect the TCP connection on detection of the VNC Client TCP disconnect or 5 s after sending the VNC *ByeBye* message, whatever comes first.

It is up to the MirrorLink Client, whether a new VNC session is launched immediately, after the old one has been terminated, from the VNC Client or Server.

The MirrorLink Client should send a UPnP TmApplicationServer:1 service *TerminateApplication* action for the stand-alone VNC Server, if the VNC Client has previously launched it for the terminated session.

Terminating a VNC session shall not impact the application status of any application on the MirrorLink Server. If the MirrorLink Client decides to re-establish the VNC session, it shall follow the steps given in Clause 5.2.

If MirrorLink Server has intentionally terminated the VNC session, the MirrorLink Client should provide a mechanism to start a VNC session again. If the MirrorLink Client decides to re-establish the VNC session, it shall follow the steps given in Clause 5.2.

In case the MirrorLink Client has intentionally terminated the VNC session, the MirrorLink Client shall wait until it received the VNC *ByeBye* message from the MirrorLink Server (or it ran into the disconnect timeout), prior sending any new UPnP Launch Application message.

In case the MirrorLink Server has intentionally terminated the VNC session, it shall wait until it detects the TCP disconnect (or it ran into the disconnect timeout), prior responding to any new UPnP Application launch action or it shall use a different URL than the previous VNC session.

5.4 Unintentionally Terminating the VNC Session

Unintentional termination of the VNC session may happen any time due to error conditions. In case of unintentional termination of the VNC session, the respective VNC Server or Client will disconnect the TCP connection. The respective counterpart should disconnect as well.

If the MirrorLink Client decides to re-establish the VNC session, it shall follow the steps given in Clause 5.2.

To avoid the VNC Server or Client persisting in a TCP TIME-WAIT time-out loop, as a result of an unintentional active disconnect, the TCP socket should be established using the SO_REUSEADDR option (or similar platform specific variants), allowing the operating system to reuse a port address, even it is currently in the TIME-WAIT state or the VNC Server should use a different, unaffected port number.

5.5 Testing Considerations

If the MirrorLink Client is in a dedicated testing state (as part of the MirrorLink Certification), it shall launch a new VNC session (either initiated automatically or manually from the user), whenever the VNC Server has intentionally terminated the VNC session.

If the MirrorLink Client is in a dedicated testing state (as part of the MirrorLink Certification), it shall launch a new VNC session (either initiated automatically or manually from the user), whenever the VNC Server has unintentionally terminated the VNC session.

6 Traditional VNC Protocol Phases

6.1 General

After the connection between the VNC Server and Client has been established, the VNC protocol processing will start according to the VNC specification. The VNC protocol consists of three main steps:

1) Exchange of handshaking messages. After the handshaking phase, the VNC connection parameters are negotiated and the connection is established.

2) Exchange of initialization messages. After this phase, both ends have agreed on all needed parameter for the following operational phase.

3) VNC Client to Server and VNC Server to Client messages are used to reflect changes of the framebuffer content on the local endpoint and user interaction on the remote endpoint.

These three VNC protocol phases are specified in more detail in the following clause.

6.2 Handshaking Phase

The handshaking phase defines a couple of messages, which are exchanged between the VNC Client and the VNC Server, as shown in the Figure 2. In general, the VNC Server presents its capabilities and the VNC Client selects the best option with regard to its own capabilities.

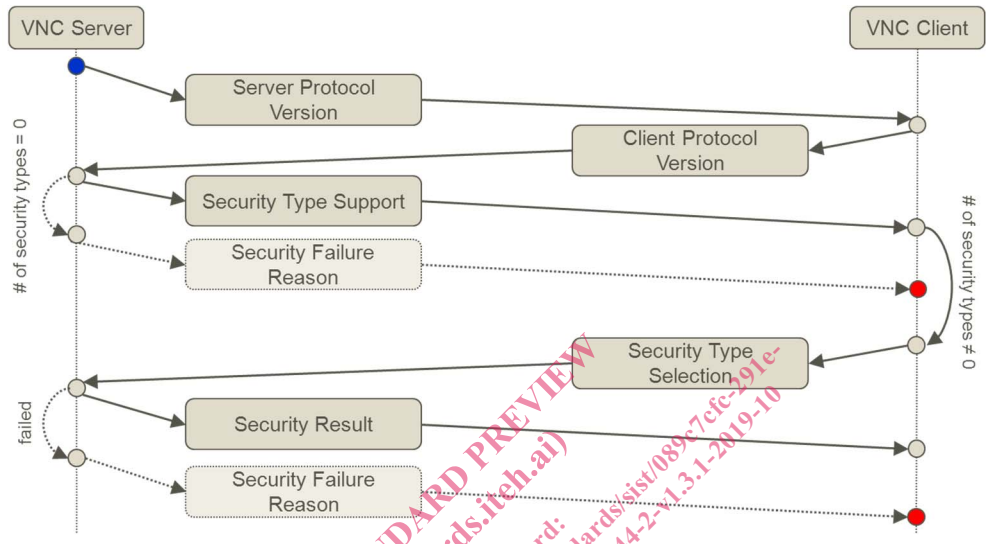

Figure 2: VNC Handshaking Phase

The Table 1 parameters shall be supported from the VNC Client and the Server.

Table 1: Requirements for Handshaking Phase Messages

Message	Origin	Parameter	Mandatory Values
Protocol Version	Server	Max. protocol version	At least 3.8
Protocol Version	Client	Max. protocol version	3.8
Security Type Support	Server	# of security types	(as specified in RFB)
		Security types	1 (None)
Security Type Selection	Client	Security type	(as specified in RFB)
Security Failure Reason	Server	Reason length	(as specified in RFB)
		Reason string	
Security Result	Server	Security status	(as specified in RFB)

Authentication and security are handled outside the VNC protocol on the link-layer and transport-layer. The VNC Client cannot expect the VNC Server to offer additional security or authentication features.

The VNC Client shall disconnect the TCP connection, after receiving a Security Failure Reason from the VNC Server. The VNC Server should disconnect the TCP connection on detection of the VNC Client TCP disconnect or 5 s after sending the VNC *SecurityFailureReason* message, whatever comes first.

The VNC Server and VNC Client shall support a MirrorLink 1.0 compliant counterpart supporting only RFB version 3.7.

6.3 Initialization Phase

The initialization phase defines a couple of messages, which are exchanged between the VNC Client and the VNC Server, as shown in the Figure 3. In general, the VNC Server presents its capabilities and the VNC Client selects the best option with regard to its own capabilities.

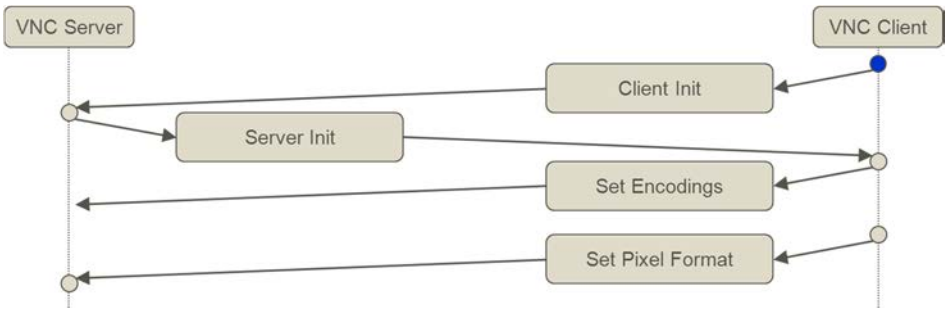

Figure 3: Initialization Phase

The Table 2 parameters shall be supported from the VNC Client and the VNC Server. For the RFB message structure, please refer to the dedicated RFB specification, as given in [1].

Table 2: Requirements for Initialization Phase Messages

Message	Origin	Parameter	Mandatory Values
Client Init	Client	Shared-flag	(as specified in RFB)
Server Init (using native framebuffer configuration)	Server	Framebuffer-width	(as specified in RFB)
		Framebuffer-height	
		Name-length	
		Name-string	
		Bits-per-pixel	
		Depth	
		Big-endian-flag	
		True-colour-flag	
		Red-/Green-/Blue max	
		Red-/Green-/Blue shift	
Set Encodings	Client	Number of encodings	(as specified in RFB)
		Encoding-type	List of required Encodings is given in Clause 8.
Set Pixel Format	Client	Bits-per-pixel	ARGB 888, RGB 565
		Depth	
		Big-endian-flag	
		True-colour-flag	
		Red-/Green-/Blue max	
		Red-/Green-/Blue shift	

The MirrorLink Server shall start in Landscape orientation. The MirrorLink Server may start in Portrait Mode within the VNC *ServerInit* message, but it shall switch to Landscape Mode using the *DesktopSizePseudoEncoding* message, with the first *FramebufferUpdate* message, following the MirrorLink Client's VNC *SetEncodings* message announcing MirrorLink support.

The MirrorLink Server shall support a framebuffer resolution of at least 800 x 480. The actual transmitted framebuffer resolution may be less to preserve the MirrorLink Server framebuffer's aspect ratio on the MirrorLink Client's display, i.e. one framebuffer dimension shall be equal to original framebuffer resolution.

The MirrorLink limits the RFB protocol, as shown in Table 2 with regard to supported colour formats, to allow for efficient implementations. Some more specific recommendations and requirements are given below.

The VNC Client shall not select a pixel format, for which the Server has not indicated support, using the Server Display Configuration VNC extension message. The VNC Server shall support ARGB 888 and RGB 565 pixel formats. The VNC Client shall at least support either ARGB 888 or RGB 565.

The VNC Client shall initially send all supported encodings within a single *SetEncodings* message. The encoding order may be used from the VNC Server as an indication on the VNC Client's priority order (first entry has highest priority). Subsequent *SetEncodings* messages shall not invalidate the use of any previous encoding or pseudo encoding, even if encodings are not repeated. They may change the priority order though. The initial *SetEncodings* message shall be sent prior the first *FramebufferUpdateRequest* message.

NOTE: The VNC Client can include support for other framebuffer encodings, referenced within [i.3], and VNC Servers can use them if supported from both devices.

NOTE: MirrorLink 1.1 or 1.2 devices can support VA H.264 framebuffer encoding, in line with previous note. MirrorLink 1.3 devices cannot expect these implementations to be fully compliant with all requirements defined in Clause 8.6.2 though. Hence, a MirrorLink 1.3 Client may exclude VA H.264 framebuffer encoding from MirrorLink 1.1 and 1.2 Servers.

The VNC Client shall not send a *SetPixelFormat* message, after the first *FramebufferUpdateRequest* message.

6.4 Framebuffer Update and Event Phase

The update and event phase defines a couple of messages, which are exchanged between the VNC Client and the VNC Server. The VNC Server only responds to framebuffer update requests, as shown in Figure 4. No response message is sent to any of the other messages.

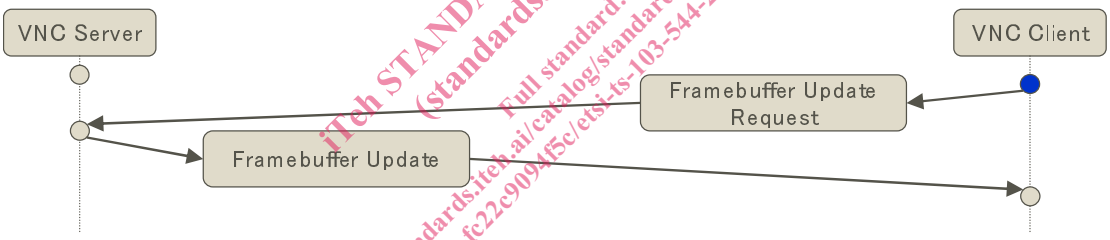

Figure 4: Framebuffer Update Phase

The Table 3 parameters shall be supported from the VNC Client and the VNC Server.

Table 3: Requirements for Framebuffer Update and Event Phase Messages

Message	Origin	Parameter	Mandatory Values
Framebuffer Update Request	Client	Incremental	(as specified in RFB)
		x-position	
		y-position	
		Width	
		Height	
Framebuffer Update	Server	Number-of-rectangles	(as specified in RFB)
		x-position	
		y-position	
		Width	
		Height	
		encoding-type	0 (Raw)