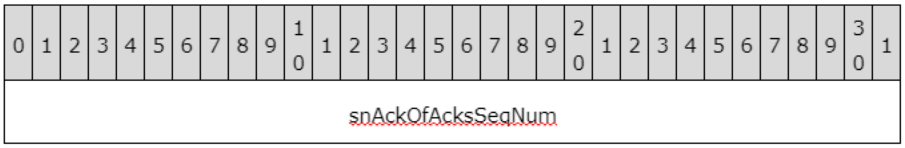
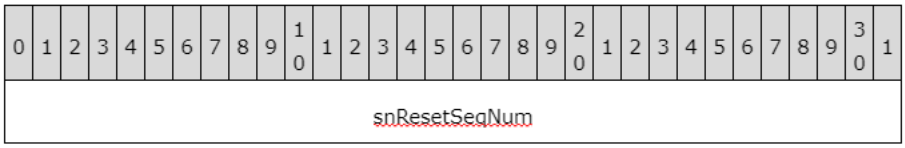
[MS-RDPEUDP]: Remote Desktop Protocol: UDP Transport Extension

This topic lists the Errata found in [MS-RDPEUDP] since it was last published. Since this topic is updated frequently, we recommend that you subscribe to these RSS or Atom feeds to receive update notifications.

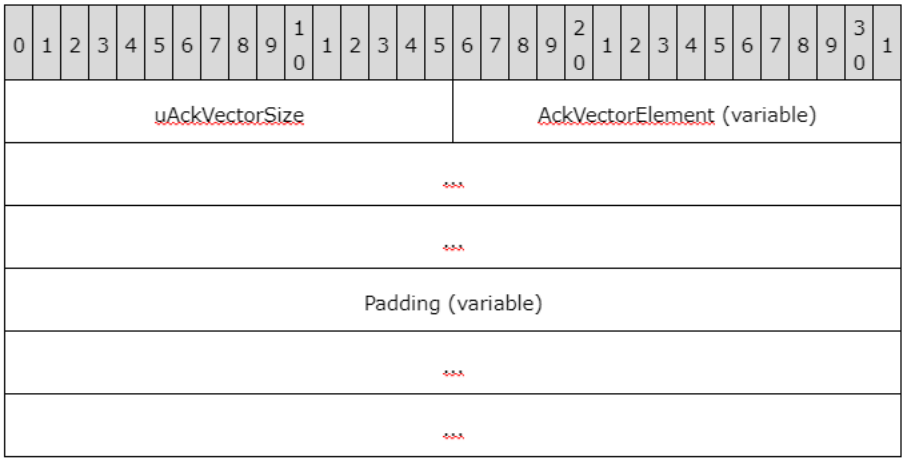
Errata are subject to the same terms as the Open Specifications documentation referenced.

RSS
Atom

Errata below are for Protocol Document Version V14.0 – 2020/03/04.

Errata Published*	Description
2020/07/06	In Section 2.2.2.6 RDPUDP_ACK_OF_ACKVECTOR_HEADER Structure, revised the field snAckofAcksSeqNum to snResetSeqNum, and updated its description. Changed from: The RDPUDP_ACK_OF_ACKVECTOR_HEADER structure resets the start position of an ACK vector (section 2.2.3.1).  The sender sets the AckOfAck sequence number with the greatest cumulative ACK it has received and processed. The sender SHOULD send AckOfAck every 20 packets. Changed to: The RDPUDP_ACK_OF_ACKVECTOR_HEADER structure resets the start position of an ACK vector (section 2.2.3.1). This structure SHOULD be sent after every 20 packets.  The sender populates snResetSeqNum with the greatest cumulative ACK it has received and processed.
2020/07/06	In Section 2.2.2.7 RDPUDP_ACK_VECTOR_HEADER Structure, revised structure description, field AckVectorElement to AckVector and added normative information to field definitions. Changed from: The RDPUDP_ACK_VECTOR_HEADER structure contains the ACK vector (section 2.2.3.1) that specifies the states of the datagram in the receiver's queue. This vector is a variable-size array. The states are encoded by using run-length encoding (RLE) and are stored in this array.

Errata Published*	Description



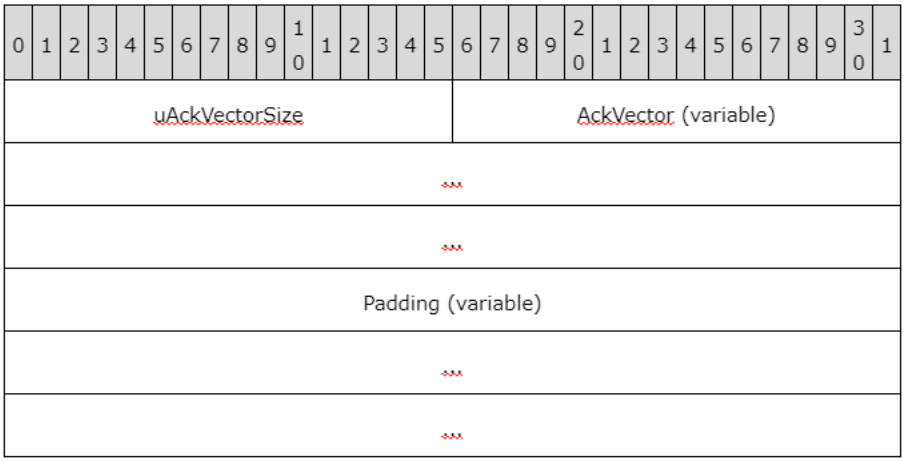
AckVectorElement (variable): An array of ACK Vector elements. Each element is composed of a state, and the number of contiguous datagrams that share the same state.

Padding (variable): A variable-sized array, of length zero or more, such that this structure ends on a DWORD ([MS-DTYP] section 2.2.9) boundary.

Changed to:

The RDPUDP_ACK_VECTOR_HEADER structure contains a variable size array of ACK Vector Elements (section 2.2.2.7.1), referred to as the ACK vector.

The ACK vector captures the state of the queue of Source Packets at the receiver endpoint. Each position in the queue can have two values that indicate whether a Source Packet is present in the queue, or not. Run-length encoding (RLE) compression is used to encode the state of Source Packets in the array.

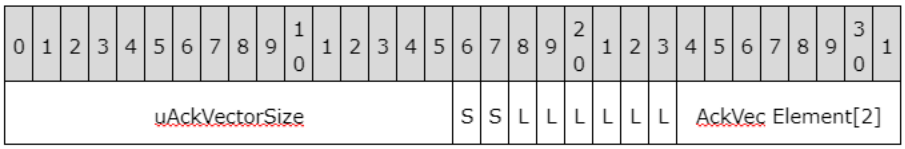


AckVector (variable): A variable size array of ACK Vector Elements (section 2.2.2.7.1). The size of the AckVector field is specified by the uAckVectorSize field.

Padding (variable): A variable-sized array, of length zero or more, such that this structure ends on a DWORD ([MS-DTYP] section 2.2.9) boundary.

Added section 2.2.2.7.1 ACK Vector Element. …

An ACK Vector Element is an 8-bit structure. The two most significant bits of each element encode the VECTOR_ELEMENT_STATE enumeration (section 2.2.1.1), while the six least

Errata Published*	Description
	significant bits specify the length of a continuous sequence of datagrams that share the same state. Removed sections 2.2.3 Vectors & 2.2.3.1 ACK Vector. 2.2.3 Vectors 2.2.3.1 ACK Vector The ACK vector captures the state of the queue of Source Packets at the receiver endpoint. Each position in the queue can have two values that indicate whether a Source Packet is present in the queue, or not. The run-length encoding (RLE) compression is used for encoding the states of Source Packets in the array. An ACK Vector comprises a number of elements, as specified by the uAckVectorSize field in the RDPUDP_ACK_VECTOR_HEADER structure (section 2.2.2.7). Each element is 8 bits long.  The ACK vectors form a binary large object (BLOB), and are padded so that they are aligned to WORD ([MS-DTYP] section 2.2.61) boundaries. This is similar to the description of ACK vectors in the Datagram Congestion Control Protocol (DCCP), as described in [RFC4341]. Revised 'section 2.2.3.1' to '2.2.2.7.1' in the following sections: . 2.2.1.1 VECTOR_ELEMENT_STATE Enumeration . 3.1.1.4.1 Lost Datagrams . 3.1.5.1.2 ACK Datagrams .5.1.2 RDP-UDP Datagram Validation
2020/07/06	In Section 3.1.1.2 Sequence Number, revised transposed characters. Changed from: Initial Sequence Number = snInitialSequenceNumber in the RPDUDP_SYNDATA_PAYLOAD Structure (section 2.2.2.5). Changed to: Initial Sequence Number = snInitialSequenceNumber in the RDPUDP_SYNDATA_PAYLOAD Structure (section 2.2.2.5). In Section 3.1.5.1.1 SYN Datagrams, corrected typos. Changed from: The RDPUDP_SYNEX_PAYLOAD structure (section 2.2.2.9) MUST be appended to the UDP datagram if the RDPUDP_FLAG_SYNEX flag is set in uFlags. Not appending this structure implies that RDPUDP_PROTOCOL_VERSION_1 is the highest protocol version supported. This structure SHOULD NOT be appended if this datagram is in response to a SYN from the other endpoint where the RDP_FLAG_SYNEX flag was not specified. The uSynExFlags field MUST be set as follows: Changed to: The RDPUDP_SYNDATAEX_PAYLOAD structure (section 2.2.2.9) MUST be appended to the UDP datagram if the RDPUDP_FLAG_SYNEX flag is set in uFlags. Not appending this structure implies that RDPUDP_PROTOCOL_VERSION_1 is the highest protocol version supported. This structure SHOULD NOT be appended if this datagram is in response to a SYN from the other

Errata Published*	Description
	endpoint where the RDPUDP_FLAG_SYNEX flag was not specified. The uSynExFlags field MUST be set as follows: In 3.1.5.1.3 SYN+ACK Datagrams, revised title. Changed from: 3.1.5.1.3 SYN and ACK Datagrams Changed to: 3.1.5.1.3 SYN+ACK Datagrams In 3.1.5.1.3 SYN+ACK Datagrams, made edits to the text. Changed from: A SYN datagram is generated, as specified in section 3.1.5.1.1, with the following fields set as follows: & ●The RDPUDP_SYNEX_PAYLOAD structure (section 2.2.2.9) SHOULD only be present if it is also present in the received SYN packet. The uUdpVer field MUST be set to the highest RDP-UDP protocol version supported by both endpoints. The highest version supported by both endpoints, which is RDPUDP_PROTOCOL_VERSION_1 if either this packet or the SYN packet does not specify a version, is the version that MUST be used by both endpoints. Changed to: A SYN+ACK datagram consists of a SYN packet, generated as specified in section 3.1.5.1.1, with these additional fields set as follows: & ●The RDPUDP_SYNDATAEX_PAYLOAD structure (section 2.2.2.9) SHOULD only be present if it is also present in the received SYN packet. The uUdpVer field MUST be set to the highest RDP-UDP protocol version supported by both endpoints. The highest version supported by both endpoints, which is RDPUDP_PROTOCOL_VERSION_1 if either this packet or the SYN packet does not specify a version, is the version that MUST be used by both endpoints. In Section 3.1.5.1.4 ACK and Source Packets Data, renamed structure. Changed from: An RDPUDP_SOURCE_PAYLOAD structure (section 2.2.2.4) header MUST be appended. Changed to: An RDPUDP_SOURCE_PAYLOAD_HEADER structure (section 2.2.2.4) header MUST be appended.

*Date format: YYYY/MM/DD