



NVIDIA Mellanox Rivermax Frequently Asked Questions (FAQ)

Rev 2.0

Document Revision History

Doc_2.0

Version	Date	Description of Change
2.0	Jan 1, 2021	Updated the document based on Rivermax version 1.7 release
1.6	May 20, 2019	Updated list of currently supported adapters.
1.5	April 01, 2019	Updated various sections throughout the document.
1.4	Sept 06, 2018	Official release of this document.
1.3	April 12, 2018	Final Remarks/Edits.
1.2	March 26, 2018	Added additional FAQ.
1.1	March 15, 2018	Initial version of the document.
1.0	March 14, 2018	Updated list of currently supported adapters.

Table of Contents

- INTRODUCTION..... 4
- FAQ: NVIDIA MELLANOX RIVERMAX..... 5
 - 1. Supported Operating Systems/Network Interface Cards (NICs)..... 5
 - 2. Virtual Machine (VM) and Containers..... 6
 - 3. SDP, RTP and Network Headers 7
 - 4. Supported Video Formats 9
 - 5. Miscellaneous..... 10

INTRODUCTION

NVIDIA® Mellanox® Rivermax® offers a unique IP-based solution for any media and data streaming use case. Rivermax together with NVIDIA GPU accelerated computing technologies unlocks innovation for a wide range of applications in Media and Entertainment (M&E), Broadcast, Healthcare, Smart Cities and more.

Rivermax leverages NVIDIA Mellanox ConnectX® hardware streaming acceleration technology that enables direct data transfers to and from the GPU, delivering best-in-class throughput and latency with minimal CPU utilization for streaming workloads.

Rivermax implements a dedicated solution for IP-based video streaming of HD to Ultra HD flows. Moreover, Rivermax is the only fully virtualized streaming solution which complies with the stringent timing and traffic flow requirement of the SMPTE ST2110-21 specification - Enabling the future cloud-based broadcast solutions.

The Rivermax library is a cross-platform library, with support for both Linux and Windows OSs over Bare Metal and Virtualized solutions. For more details, go to the [Rivermax Product Page](#)

The sections below address FAQ related to NVIDIA Mellanox Rivermax.

FAQ: NVIDIA MELLANOX RIVERMAX

1. Supported Operating Systems/Network Interface Cards (NICs)

Question	What Operating Systems (OS) does Rivermax support? Linux/Windows?
Answer	Rivermax supports both Linux and Windows <ul style="list-style-type: none">• Windows Server 2016• Windows 10 Pro• Linux distros: Ubuntu 16.04, 18.04 & 20.04, Centos 7.7 and RH7.5-7 For a complete update list, go to Rivermax Product Page

Question	Which NVIDIA Mellanox Network Adapters (NICs) does Rivermax support?
Answer	Rivermax supports both Linux and Windows Rivermax supports the following list of adapter cards: <ul style="list-style-type: none">• MCX512A-ACAT - ConnectX-5 10/25GbE• MCX516A-CDAT - ConnectX-5 Ex 100GbE• MCX623106AN-CDAT - ConnectX-6 Dx 100GbE• MCX621102AN-ADAT - Connect6 Dx 25GbE For a complete update list, go to Rivermax Product Page BlueField-2 DPU will be supported in future releases

Question	How can I ensure that Rivermax is using the supported NIC type? (i.e., both ConnectX-4 and ConnectX-5 NICs are installed on my machine)
Answer	The device IP is used when creating Rivermax resources. Rivermax uses the IP address to detect if the associated Mellanox NIC is supported by the Rivermax library. If the NIC is not supported, the API returns an error code.

Question	Does the same Rivermax SDK work on all supported NIC types?
Answer	Yes. Advanced features are hardware-dependent; this means they might be supported on specific NICs and not on other types.

Question	Does Rivermax support Arm architecture?
Answer	Rivermax can run on x86 or Arm architectures.

Question	Does Rivermax support AMD ?
Answer	Rivermax supports AMD and Intel. Refer to the tuning guide for tips on achieving maximum performance.

Question	Which applications can be supported by Rivermax SDK?
Answer	The Rivermax SDK can be used with any data streaming application. Rivermax provides very high bandwidth, low latency, GPU-Direct and zero memory copy. Applications that need to support IP-Enabled TV Studio Media (SMPTE ST 2110) is a good example of a use case for the Rivermax SDK. Some other popular use cases include video broadcasting, video wall-display, high BW camera ingest, retail camera input analytics, healthcare DICOM-RTV, and H.264 and H.265 compressed data.

2. Virtual Machine (VM) and Containers

Question	Does Rivermax support a VM/SR-IOV?
Answer	Rivermax supports virtualization on top of VMware ESXi and OpenStack (KVM)

Question	Can Rivermax run from Docker Containers
Answer	Yes. Rivermax can run inside a container, without any impact on performance or functionality; a code example is provided as part of Rivermax release 1.7.

3. SDP, RTP and Network Headers

Question	Does Rivermax generate RTP packets?
Answer	No. Rivermax receives from the application RTP packets and based on that it generates network packets (adds MAC/IP/UDP).

Question	Does Rivermax create the Ethernet & IP layer?
Answer	Yes. The Rivermax layer create all the Ethernet/IP/UDP headers (based on the information extracted from the SDP).

Question	Does Rivermax support receiving or transmitting RTP payload/header with variable length?
Answer	Yes. this is supported by Rivermax. For send requests, the sizes must be indicated upon the output stream creation via <code>rmax_mem_block.app_hdr_size_arr/data_size_arr</code> . For receive requests, the min and max sizes must be indicated upon the input stream creation via <code>rmax_in_buffer_attr.min_data/hdr_size</code> and <code>rmax_in_buffer_attr.max_data/hdr_size</code> .

Note: Rivermax input stream can receive traffic that consists packets with variant sizes that can dynamically change during runtime but must be in the range:

[`min_data/hdr_size` - `max_data/hdr_size`]

Note: It is strongly recommended to work with constant length in order to fully utilize hardware offloads and achieve the optimal performance.

Question	On the sender application (application TX), can the data be separated from the RTP headers?
Answer	Yes. The RTP headers can be separated from the data or placed with the data. Rivermax supports both options.

Question	Is the usage of SDP mandatory?
Answer	<p>No. Rivermax supports any data stream that is based on UDP/IP.</p> <p>SDP provides important information used by Rivermax to configure the correct parameters for the transmitted stream. It is used by applications that support SMPTE ST 2110.</p> <p>A generic API can be used for other types of application that would like to benefit from the Rivermax performance but don't support SMPTE ST 2110.</p> <p>Receiving any data stream by Rivermax doesn't require an SDP file.</p>

Question	Media attributes are provided to Rivermax via string, why?
Answer	<p>Rivermax uses a standard SDP format which is textual.</p> <p>The SDP string stores a lot of information. Some of this information is being used by Rivermax today, while some of it will be used in future versions.</p> <p>The Generic API doesn't require media attributes.</p>

Question	An application holds the data in a binary format. Is it complicated to convert it to an SDP format?
Answer	<p>The application can easily convert the binary format to an SDP textual format, e.g. via <code>sprintf()</code>.</p>

Question	How can I know which SDP attributes are required by Rivermax?
Answer	<p>Documentation of <code>rmax_create_output_stream()</code> routine explains what the minimal set of required SDP attributes is.</p> <p>Please refer to the <code>rivermax_api.h</code> file for an example of an SDP.</p>

Question	Does Rivermax support offload of localhost addresses?
Answer	<p>No. Using addresses in the 127.0.0.0/8 subnet is not supported for loopback offload. To offload packets that must be looped back, use one of the following:</p> <ul style="list-style-type: none"> • NIC local IP address • Multicast address (which the host has joined) - please review documentation for further details

Question	Does Rivermax support PTP? (timing synchronization)
Answer	<p>Yes. Using Rivermax, each frame must be committed with the time parameter set to the system time (in nanoseconds) at which it must be sent.</p> <p>The system time is synchronized using a 3rd party software to lock the system time to a PTP GM.</p> <p>The timing synchronization is supported for both Linux and Windows.</p> <p>The PTP client is provided by a 3rd party.</p>

Question	Does Rivermax support unicast and multicast IP?
Answer	Yes - Rivermax supports both unicast and multicast IP formats

4. Supported Video Formats

Question	Which SMPTE streams are supported by Rivermax?
Answer	Rivermax supports 2110-20, 2110-30, 2110-40 2022-7, 2022-6, 2110-31 and 2110-22

Question	Does Rivermax support the 2022-7 streams? If Yes, how?
Answer	<p>Yes.</p> <p>Transmit:</p> <p>Rivermax reads from the SDP string if a redundant transmission is requested using the a=group: attribute with the DUP semantic, and what are the properties of the duplicated streams.</p> <p>When redundancy is requested, Rivermax internally replicates chunks to multiple streams. This mode is currently not validated.</p> <p>Receive:</p> <p>ConnectX-5 provides software support: Rivermax supports 2022-7 on the receive side just by allowing the application to receive 2 media streams. The redundancy is made in the application layer.</p> <p>ConnectX-6 Dx provides HW offload: Redundant stream seamless reconstruction uses ConnectX-6 Dx hardware</p>

Question	Does Rivermax support 8K UHD streams?
Answer	<p>ConnectX-5 Rivermax Linux version supports receiving an 8K UHD video format of 20-80Gb/s. Accurate pacing for an 8K stream according to the SMTPE ST 2110 is supported by ConnectX-6 Dx.</p> <p>An example code will be provided in future releases.</p>

Question	Does Rivermax support both interlaced & progressive (i.e. 1080i & 1080p) modes?
Answer	Yes. Rivermax supports both interlaced and progressive modes.

5. Miscellaneous

Question	Does Rivermax support NMOS IS-04 /IS-05?
Answer	<p>NMOS is used for discovering, registering and managing Media flows—it is part of the application layer.</p> <p>The NMOS specifications describe a control plane that makes a ST 2110-based infrastructure manageable and simpler to operate.</p> <p>Rivermax versions 1.7 and up provide an example code on how to integrate NMOS with Rivermax.</p> <p>There is a NMOS implementation provided by Richard Hastie- You can run this on our switch or on a host.</p> <p>You can see Richard's presentation from IBC: http://www.ipshowcase.org/wp-content/uploads/2019/10/1500-Simplifying-JT-NM-TR-1001-1-Deployments-through-Microservices.pdf</p> <p>Richard wrote this as well: https://github.com/rhastie/docker-nmos-cpp</p>

Question	Which SMPTE ST 2110 sender types are supported by Rivermax?
Answer	Rivermax supports Narrow Sender (2110-20TPN) and Narrow Linear sender for 2110-22TPNL

Note: Wide sender has not yet been fully tested.

Question	Does Rivermax support the extended UDP Size?
Answer	Yes, make sure you increase interface MTU and set Rivermax strides sizes correctly.

Question	Does Rivermax support GPUDirect (direct access to GPU memory)?
Answer	<p>Rivermax supports GPUDirect—memory can be allocated directly on the GPU.</p> <p>Example code is provided as part of Rivermax release 1.7</p>

Note: This feature is supported only for Linux. Windows will be supported in future releases.

Question	Memory allocation: Is it done by Rivermax or user/application?
Answer	Both modes are supported: memory can be allocated by application or by Rivermax. For input stream, it is necessary to precede the memory allocation with a call to <code>rmax_in_query_buffer_size()</code> to determine the amount of memory required to be allocated by the application.

Question	Does Rivermax check the content of the data?
Answer	No

Question	Can an application dynamically, during runtime, change the output stream data/app headers sizes or can these sizes be set only once upon output stream creation?
Answer	The sizes must be set once upon output stream creation. Dynamic modification of data/app header sizes is not supported for media streams. Using the Tx Generic API, the data/header sizes can be modified on the fly.

Question	Does Rivermax support Receive and Transmit?
Answer	Yes. Both are supported and provide very high bandwidth with low CPU consumption

Question	Why use Rivermax and not Mellanox VMA?
Answer	Rivermax is based on the Mellanox VMA Linux technology combined with a more dedicated solution for the M&E industry needs. Mellanox VMA does not provide a solution for Windows, whereas Rivermax works on both Linux and Windows OSs and provides the same API and functionality set. Dedicated features (software and hardware) for the data streaming will be supported only by Rivermax.

Question	Do we still need VMA for Linux?
Answer	<p>Yes. VMA is needed.</p> <p>Rivermax uses VMA services in the control path: e.g. Hardware resources allocation, address resolution, memory registration and more.</p> <p>However, the data path is fully implemented in Rivermax in order to provide the optimal performance.</p>

Question	Does Rivermax provide information such as how many buffers are available, committed, or sent?
Answer	No. This capability will be available in future versions

Question	Does Rivermax expose advanced programmable capabilities of the Mellanox NIC for enhanced filtering?
Answer	<p>No.</p> <p>Mellanox NICs have some advanced parser capabilities that are not fully exposed by Rivermax.</p>

Question	Does Rivermax support audio or ancillary on top of video?
Answer	Yes—these are 2110-30, 2110-31 and 2110-40. An example code for 2110-30 and 2110-40 is available as part of the SDK.

Question	Does the Rivermax SDK provide an example that sends real video content?
Answer	<p>Yes—Rivermax media_receiver can send a pre-prepared video template file in YCbCr format (part of the example code).</p> <p>Rivermax SDK also provides rivermax_player—an example code that integrates FFmpeg with Rivermax. This example allows sending an MP4 video file.</p>

Question	Is Rivermax SMPTE ST 2110 JT-NM tested ?
Answer	<p>Rivermax is “JT-NM Tested” and has been tested for SMPTE ST 2110-20/30/40, 2022-7 compliance. Rivermax also supports ST2022-6 and ST 2110-31. The full report can be downloaded from the JT-NM.org: https://jt-nm.org/jt-nm_tested/index.shtml</p>

Notice

This document is provided for information purposes only and shall not be regarded as a warranty of a certain functionality, condition, or quality of a product. NVIDIA Corporation ("NVIDIA") makes no representations or warranties, expressed or implied, as to the accuracy or completeness of the information contained in this document and assumes no responsibility for any errors contained herein. NVIDIA shall have no liability for the consequences or use of such information or for any infringement of patents or other rights of third parties that may result from its use. This document is not a commitment to develop, release, or deliver any Material (defined below), code, or functionality.

NVIDIA reserves the right to make corrections, modifications, enhancements, improvements, and any other changes to this document, at any time without notice. Customer should obtain the latest relevant information before placing orders and should verify that such information is current and complete.

NVIDIA products are sold subject to the NVIDIA standard terms and conditions of sale supplied at the time of order acknowledgement, unless otherwise agreed in an individual sales agreement signed by authorized representatives of NVIDIA and customer ("Terms of Sale"). NVIDIA hereby expressly objects to applying any customer general terms and conditions with regards to the purchase of the NVIDIA product referenced in this document. No contractual obligations are formed either directly or indirectly by this document.

NVIDIA products are not designed, authorized, or warranted to be suitable for use in medical, military, aircraft, space, or life support equipment, nor in applications where failure or malfunction of the NVIDIA product can reasonably be expected to result in personal injury, death, or property or environmental damage. NVIDIA accepts no liability for inclusion and/or use of NVIDIA products in such equipment or applications and therefore such inclusion and/or use is at customer's own risk.

NVIDIA makes no representation or warranty that products based on this document will be suitable for any specified use. Testing of all parameters of each product is not necessarily performed by NVIDIA. It is customer's sole responsibility to evaluate and determine the applicability of any information contained in this document, ensure the product is suitable and fit for the application planned by customer, and perform the necessary testing for the application in order to avoid a default of the application or the product. Weaknesses in customer's product designs may affect the quality and reliability of the NVIDIA product and may result in additional or different conditions and/or requirements beyond those contained in this document. NVIDIA accepts no liability related to any default, damage, costs, or problem which may be based on or attributable to: (i) the use of the NVIDIA product in any manner that is contrary to this document or (ii) customer product designs.

No license, either expressed or implied, is granted under any NVIDIA patent right, copyright, or other NVIDIA intellectual property right under this document. Information published by NVIDIA regarding third-party products or services does not constitute a license from NVIDIA to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property rights of the third party, or a license from NVIDIA under the patents or other intellectual property rights of NVIDIA.

Reproduction of information in this document is permissible only if approved in advance by NVIDIA in writing, reproduced without alteration and in full compliance with all applicable export laws and regulations, and accompanied by all associated conditions, limitations, and notices.

THIS DOCUMENT AND ALL NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, "MATERIALS") ARE BEING PROVIDED "AS IS." NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL NVIDIA BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF NVIDIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Notwithstanding any damages that customer might incur for any reason whatsoever, NVIDIA's aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms of Sale for the product.

VESA DisplayPort

DisplayPort and DisplayPort Compliance Logo, DisplayPort Compliance Logo for Dual-mode Sources, and DisplayPort Compliance Logo for Active Cables are trademarks owned by the Video Electronics Standards Association in the United States and other countries.

HDMI

HDMI, the HDMI logo, and High-Definition Multimedia Interface are trademarks or registered trademarks of HDMI Licensing LLC.

OpenCL

OpenCL is a trademark of Apple Inc. used under license to the Khronos Group Inc.

Trademarks

NVIDIA, the NVIDIA logo, Connect X, and Rivermax are trademarks and/or registered trademarks of NVIDIA Corporation in the U.S. and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

Copyright

© 2021 NVIDIA Corporation. All rights reserved.