NDI® Best Practices with TriCaster®

Setup and configuration of an NDI® network with local, remote and cloud based TriCaster® systems.

RELEASE 2024

Contents

1 Introduction

1.1 About this guide

This guide will help with the core design of an NDI® network for use with TriCaster® live production systems. It offers recommendations on network configuration, NDI devices and suggestions for the best performance with remote and cloud-based systems.

It is important to understand that networks can vary greatly, and so the settings required to get the best performance may also vary. Use the suggestions from this guide as a starting point, then perform testing to verify the desired performance is achieved. If different configuration options are required, use those settings for that implementation and document for future reference.

While this document provides details on how to use TriCaster live production solutions, you should refer to the NDI Docs & Guides on the NDI.video website for more information about NDI technology.

NDI.video Docs & Guides

NDI Tools is a free suite of software available for Windows and Macintosh systems found on the ndi.video website. This software helps you test and fix NDI workflows efficiently. You should install this suite of tools on all systems that will use NDI.

Another great resource of information is the *NDI and Performance Media Networking* training course available on Viz University.

You can enroll in this free course by creating an account, by visiting Viz University here.

1.2 TCP/IP knowledge

To get the most from this guide, you should have a basic understanding of TCP/IP networking. **These concepts are not taught in this guide.** If they are unfamiliar, there are many online resources to better understand these concepts.

- IP address
- Netmask
- Gateway
- DNS
- Unicast vs Multicast communication
- TCP and UDP packets

1.3 NDI versions

At the writing of this document, NDI version 6.0 is the current release. NDI is backwards compatible, meaning that newer NDI versions continue to be compatible with previous NDI versions. Any aspects of information that is NDI version specific will be noted.

1.4 Document Revision

Release	Changes
July 2024	Initial Release

2. The NDI Protocol

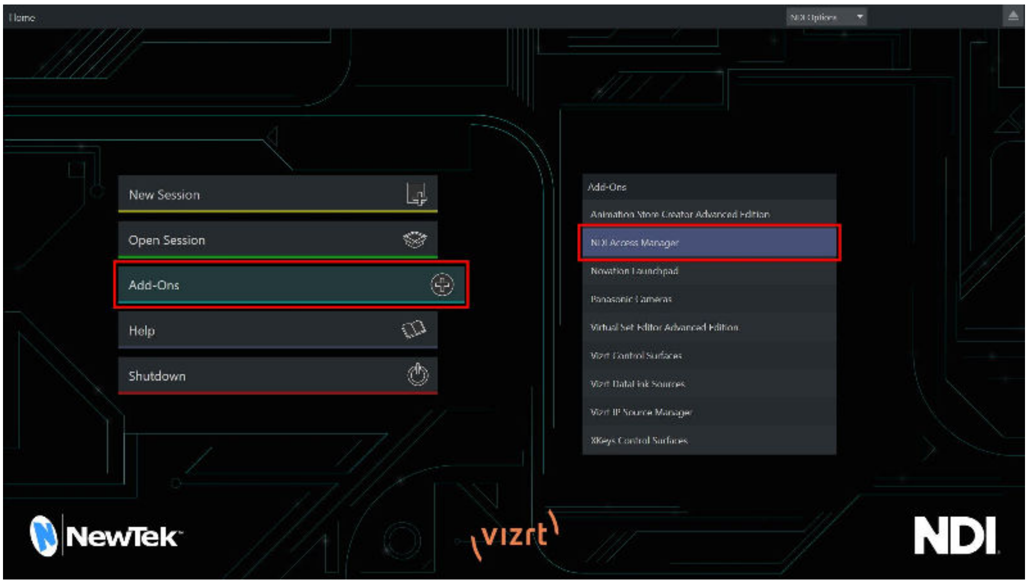
NewTek was acquired by Vizrt in 2019, and NDI was spun out as a separate brand under Vizrt. The NewTek and Vizrt brands merged in 2023, NDI remains a separate brand. Vizrt continues to work as a key partner, and in some cases with shared R&D resources, to improve the NDI technology, initially created by NewTek. NDI is an IP technology for broadcasting that TriCaster, the first ever NDI product, uses extensively.

NDI has rapidly become the leading real-time video broadcast IP technology globally, adopted by major broadcasters and individual content creators alike.

There are many aspects to how the NDI protocol operates on your network and can be configured for operation.

2.1 Configuring NDI

The tool for configuration of NDI on a computer is NDI Access Manager. This application can be found on a TriCaster system by going to the Add-Ons section of the Home Screen.



NDI Access Manager is also included with NDI Tools on Windows and Macintosh systems. Configuration using this tool will affect the default setting for NDI applications run on the system. All NDI applications read a file called **ndi-config.v1.json** that is located on a computer. This file has the settings established by NDI Access Manager, which can be found as referenced in the table below.

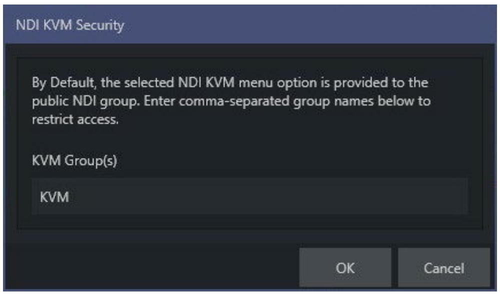
Operating System	Location
Macintosh	$HOME/.ndi/ndi-config.v1.json
Linux	$HOME/.ndi/ndi-config.v1.json
Windows	%programdata%/NDI/ndi-config.v1.json

Note that the NDI configuration file data is only read when an application first launches. Changes to this file will not take effect with NDI applications that are already running. If you update settings using NDI Access Manager, you will need to restart any NDI applications for the changes to take effect.

There is no NDI Access Manager tool on Linux systems, therefore you will need to edit the file using a text editor. You can also use NDI Access Manager on a Windows/Macintosh system to configure the information and then copy the file to a Linux system.

This settings file is used by any application using the NDI SDK. Some applications may use the NDI Advanced SDK, which offers the ability to configure separate, per channel NDI settings. If you find an application is not respecting the changes made by NDI Access Manager, look to see if the application itself offers additional NDI configuration options.

For example, the NDI-KVM feature on TriCaster allows the entering of an NDI group name separate from what is configured using NDI Access Manager.



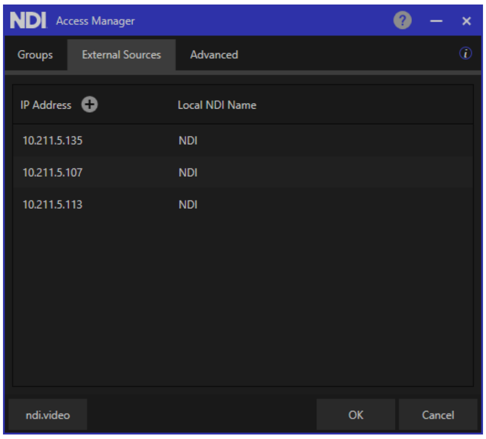
By Default, the selected NDI KVM menu option is provided to the public NDI group. Enter comma-separated group names below to restrict access.

KVM Group(s): KVM

2.2 Discovery and Registration

2.2.1 What is discovery and registration?

NDI employs a built-in discovery mechanism. This allows a user to operate NDI devices without having to know the underlying details of the network. Instead of IP addresses and port numbers, a list of friendly human-readable device names and channels can be displayed and selected. There are different discovery solutions available, which method to use depends on your network size and complexity.



IP Address	Local NDI Name
10.211.5.135	NDI
10.211.5.107	NDI
10.211.5.113	NDI

2.2.2 mDNS

The default and most common type of discovery is mDNS (Multicast Domain Name System). When you first install NDI devices, this is the version of discovery that will be used. mDNS is the only type of discovery supported by all versions of NDI (going back to NDI version 1.0).

This method of discovery is simple and easy to use, as it will bind to all network interfaces on the system. Beyond device and channel naming, no configuration is required. However, on systems connected to multiple networks, this method can open NDI signals to networks where you might not want it to be available.

mDNS discovery takes some time to find all devices on the network. On smaller size networks where there are only a few dozen NDI endpoints, this will not be noticeable. As the number of end-point increases, so will the discovery time. It is possible to take a minute or two to discover all sources if you have many hundreds of NDI devices on the network.

There are some specific network limitations to be aware of with mDNS. Its use of multicast packets will limit discovery of sources on the same subnet (it will not route to other networks). It’s also possible that some networks might have multicast communication disabled, which will make mDNS discovery fail to operate. In these instances, you should utilize either the Discovery Server or External Sources methods, both detailed below.

mDNS is recommended for small to medium size installations (less than 100 NDI devices), and where the NDI network is configured as a single subnet.

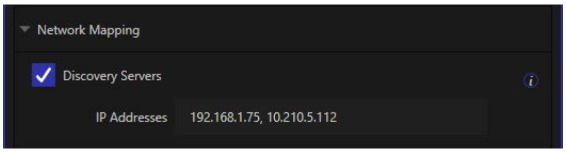
2.2.3 Discovery Server

Discovery Server relies on a central application to provide the discovery mechanism. This application should be run on a system that will be constantly available. It is not recommended to run the Discovery Server on the same system used for production applications, because if the system needs to be restarted, discovery will be unavailable until it becomes active again.

Discovery Server is available for Windows, Macintosh and Linux systems running on Intel x86 or ARM. For Windows systems, this application can be found with NDI Tools. For all other operating systems, you will need to download the free NDI SDK to access the Discovery Server application.

Discovery Server requires that the NDI applications are NDI version 4 or higher. Older NDI applications (using NDI libraries earlier than version 4), will continue to use mDNS even if Discovery Server has been configured.

Discovery Server can be set up on multiple redundant servers. This requires applications that support NDI 5 or higher. You can enter a comma separated list of server IP addresses, or hostnames, in NDI Access Manager. If using an IP address, make sure the servers are configured with IP address that do not change, either setup with a static IP or using DHCP reservations. For systems connected to multiple Discovery Servers, all servers will be updated with source registration and availability.



☑ Discovery Servers

IP Addresses 192.168.1.75, 10.210.5.112

In fact, if you have two separate NDI networks, each managed with a separate Discover Server, you can enter both into the NDI Access Manager list, and that system will be able to see all sources across both servers. Discovery Server availability has no effect over existing connections, which will continue to operate without interruption.

Discovery Server overcomes the limitations of mDNS. With proper routing, discovery can occur across multiple subnets. Also, Discovery Server will prevent NDI traffic from being visible on all networks. If you have a system connected to two networks and want to keep NDI contained to a single network, configure Discovery Server on the network in question and configure all the end points to use it.

The time is takes for systems to get the full list of sources on the network is greatly improved as the centralized database nature of Discovery Server make it possible to get this information swiftly, not waiting for the mDNS announcement of all devices on the network to complete.

It is important to understand how Discovery Server handles discovery when mixed with mDNS discovery devices. An endpoint configured to use Discovery Server will only be visible to another endpoint using Discovery Server, but all endpoints will be able to see sources using mDNS discovery on the network. Therefore, older NDI devices that only support mDNS discovery can still be used when the rest of the NDI network is configured for Discovery Server.

How much computing power or bandwidth does Discovery Server need? No NDI signals flow through Discovery Server, it is only used during the discovery process of NDI communication. When a connection is established, the transmission occurs from point-to-point. In general, Discovery Server can run on a low spec system. The most important aspect is that it is constantly available, because an NDI application will need to communicate with it to find other NDI sources on the network. For reference, one of the authors of this paper has used a Raspberry Pi 3B+ single board computer to run Discovery Server without issue.

Discovery Server is recommended for larger NDI networks, cloud-based installations of any size, and more complex networks. You can also use it on smaller size networks, making it easier to grow with more devices in the future.

2.2.4 External Sources

The External Source option allows manual entry of an IP address for device discovery. This allows accessing sources outside the local subnet and requires that NDI applications be using NDI version 2 or higher. No Discovery Server application is required.

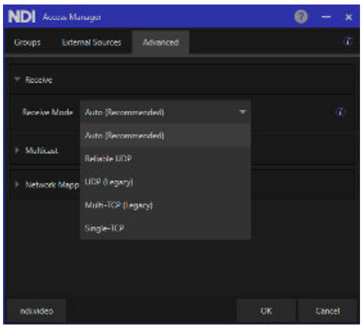
The downside to External Sources is that you will need to manually update each NDI client with the information. As the number of devices on the network increases, the need for manual management will also grow.

External Sources is not a recommended workflow unless you have a very small NDI setup or are using it for workflow testing. The exception is when using devices that are NDI|HXv1, as devices of this level do not support Discovery Server. For NDI|HXv1 devices, using External Sources is the only option to interface across multiple subnets.

2.3 Transmission

2.3.1 What is transmission?

Once you select an NDI signal inside an application, the discovery process is completed, and transmission takes over. This is how the video and audio signal moves from one device to another and when NDI is going to use continuous bandwidth on your network. NDI Access Manager allows transmission control configuration, including different TCP or UDP unicast transfer modes and multicast transmission.



Receive Mode: Auto (Recommended)

2.3.2 Unicast

The default method of NDI transmission is unicast. Unicast is the most common type of transmission of data on any computer network. All traffic on the Internet uses unicast.

Unicast transmission means that it only goes to one destination. If two clients connect to the same NDI source then two unicast transmissions will be sent, one for each destination. Each receiver that connects to an NDI source will increase the consumed network bandwidth. Two receivers will double the bandwidth on the sender, three receivers will triple the bandwidth of the sender, etc.

This is an important factor to account for when using NDI for signal distribution. Having two or three receivers on a source will likely not cause any issues, but having a dozen or more will often overflow the network bandwidth of the sending device. Multicast is one way to avoid this issue and will be discussed later in the document.

Despite this, unicast is the recommended method of transmission in most scenarios. NDI unicast transmission can be performed using one of four different protocols. The video quality is the same for each, they only vary in how the bits are transmitted. To determine which is the best choice for your network, let's discuss each in turn:

- **Single-TCP mode** is the first and original type of NDI transmission going all the way back to NDI version 1. All NDI devices support this mode.

  Single-TCP creates a single connection between devices and transfers information using TCP packets. This protocol uses fewer computing resources than other methods, which is beneficial to low power devices. Single-TCP doesn't fully support systems with multiple NICs on the same network for more network bandwidth. In this case, each NDI signal only uses one network interface, even if there are multiple NICs on the system. This doesn't mean both NICs aren't used, but it happens at the connection level. One receiver only uses one NIC, but two receivers could (but not necessarily) use a different NIC interface each.

  Single-TCP uses automatic repeat request (ARQ) method for error correction, where missing or corrupted data is resent to complete the transmission of all data.

  Single-TCP is the least preferred method of transmission but is useful for troubleshooting issues as it is guaranteed to be supported by all NDI devices. There is no reason you cannot use Single-TCP if determined to work best on your network, but if a different mode works equally as well then it is recommended to use that method instead.

- **UDP mode** was introduced with NDI version 3.5. This protocol method adds two improvements: Forward Error Correction (FEC) and packet level bonding.

  Forward Error Correction will increase your NDI bandwidth by 6.25% which includes the additional error correction data. FEC allows NDI to run on higher latency networks, where TCP re-transmission cannot function correctly. Re-transmission error correction requires that your network latency be low enough for missing data to be handled in less time than a single video frame. If you find that your network latency (ping), is higher than a single video frame, then UDP is a good choice.

Frame Rate	Maximum Latency
59.94 fps	16.7 ms
50 fps	20 ms
29.97 fps	33.4 ms
25 fps	40 ms
23.976 fps	41.7 ms

UDP will also take advantage of multiple NIC connections at the packet level. If you have a sender with two NIC interfaces, half the NDI data will be sent on one interface and half on the other.

- **Multi-TCP** was introduced with NDI version 4. This provides the bonding of multiple interfaces while using TCP packets. Since TCP uses ARQ for error correction, there is no increase in bandwidth as found with UDP. Multi-TCP is best used on systems with multiple 1Gb Ethernet connections to the same network.

- **Reliable UDP mode** was introduced with NDI version 5. This method of transmission improves communication, especially in cases where there are multiple speed fabrics on the network (i.e. 1Gb and 10Gb devices working together). Reliable UDP will multiplex signals together into a single connection between devices, and therefore may not utilize multiple NICs connections as well as UDP or Multi-TCP would. While using UDP packets, Reliable UDP employes a *TCP like* ARQ mechanism, which has been optimized for NDI traffic flows. Reliable UDP is best for networks using multiple speed fabrics.

In NDI Access Manager, the default Receive mode is Auto, which means that all the methods listed above are supported. If you select anything off this list, then you will be limiting transmission to the selection chosen and Single-TCP. Single-TCP can never be removed as a transmission mode, as it provides a fallback for all devices to communicate if a newer mode is not available. It is recommended the Auto mode be used, when possible, which should result in the best transmission mode used between the NDI endpoints.

While the NDI software SDK does support all the transmission modes listed, this may not apply to hardware or FPGA devices. Some hardware devices have lower spec processors which may not support all transmission modes (beyond the required Single-TCP).

If you are mixing NDI versions, this matrix shows what protocol would be used with the *Auto* setting selected in NDI Access manager.

↓ Source / Dest →	NDI v5/v6	NDI 4	NDI 3.5	NDI v1 to v3
NDI v5/v6	Reliable UDP	Multi-TCP	UDP	Single-TCP
NDI v4	Multi-TCP	Multi-TCP	UDP	Single-TCP
NDI v3.5	UDP	UDP	UDP	Single-TCP
NDI v1 to v3	Single-TCP	Single-TCP	Single-TCP	Single-TCP

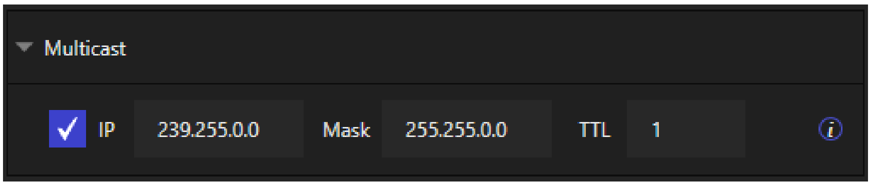
2.3.3 Multicast

Starting with NDI version 3, it is possible to enable multicast sending. This is useful if you need to share a source to many destinations, such as for video distribution type workflows.

Enabling multicast doesn't remove the possibility of unicast being used. If a multicast connection cannot be created, then the connection will try to use unicast as a fallback. For example, if multicast is enabled in NDI but is blocked at the network level, NDI signals can try to operate as unicast connections.

Multicast does require network switches that support **IGMP Snooping**. Make sure **all** switches on the network have at least IGMPv2 enabled before testing multicast. Incorrect switch configuration will cause a broadcast of signals on the network, sending data to endpoints that didn't request the signal. IGMP Snooping requires Layer 2 or Layer 3 managed switches. It is recommended to use switches from the same manufacturer on the network when using multicast, as it makes it easier to troubleshoot and test.

Before multicast is enabled, first verify that NDI devices are running correctly using unicast. It is possible to run a mix-environment of some NDI devices using unicast and others using multicast. In fact, this is the recommended workflow. Only enable multicast on the NDI devices that require one-to-many transmission and configure all other devices to use unicast.



Multicast

☑ IP 239.255.0.0 Mask 255.255.0.0 TTL 1

Activating multicast results in the automatic generation of a random multicast group IP, determined by the IP and Mask settings within the NDI Access Manager. The design does not allow for specifying a single multicast group IP because a device can produce multiple signals depending on what the receiver demands, such as a full bandwidth stream, a proxy bandwidth stream, or an audio-only stream, with each signal requiring its own unique multicast group IP.

The standard mask allows for a span of 65,000 addresses, significantly lowering the odds of two NDI devices choosing the identical IP group address. Should such an overlap occur, packets are tagged to avoid signal interference, although this situation would reduce network efficiency.

In scenarios with substantial multicast traffic—perhaps involving several dozen NDI devices configured for multicast—it's advisable to consider employing varied IP ranges to further minimize the risk of multicast group IP conflicts. You could establish the first multicast group with the default address 239.255.0.0 for some NDI devices, and for an additional group, opt for 239.254.0.0 to allocate a fresh pool of 65,000 addresses for other devices to use.

The Time-To-Live (TTL) setting is at 1 for multicast traffic, which usually isn't forwarded between networks. However, some routers might be set up to route it. To allow multicast traffic to travel across subnets, increment the TTL by one for every router it crosses.

Another way to handle signal distribution is with the NDI Tool **NDI Bridge**. *Local mode* in this application allows the bandwidth of the Bridge system to be used to distribute signals. For example, perhaps you have a Viz Connect Solo (Spark) unit that needs to be shared with 15 end points and multicast isn't possible on the network. With NDI Bridge on a computer that has a 10Gb network interface, there would be enough bandwidth to send signals to all 15 end points, while continuing to use unicast transmission for the entire workflow.

2.4 NDI Compression

Various compression modes are employed by NDI, which can impact different resource aspects of a system or determine the network bandwidth for the signal. NDI Bandwidth charts can be found in section 6.5 of this document or on the ndi.video website.

Bandwidth (ndi.video)

2.4.1 NDI High Bandwidth

NDI High Bandwidth is generally the preferred mode of compression. It offers low latency and high image quality, but it consumes the most bandwidth. This is the type of compression used by almost all NDI software applications. The compression algorithm is comparable to the Apple ProRes codec but optimized for real-time performance.

NDI High Bandwidth targets CPU power for any encoding or decoding of signals. A multi-core, high performance CPU is desirable when working with this compression type.

2.4.2 NDI|HX

NDI|HX is available in three different versions, but the common aspect between all of them is using H.264 or HEVC (H.265) image compression, which leverages GPU hardware acceleration for decoding and encoding of signals when possible. Because of this, an Nvidia GPU is highly recommended for any NDI applications working with HX footage other GPU cards can work but may limit performance.

NDI|HXv1

This version of HX was designed to allow existing legacy hardware products to work with NDI. Because this hardware was built *before* NDI support was intended, it was not possible to implement all NDI features. These devices are compatible with current NDI applications but require NDI Tools to add the appropriate driver for NDI|HXv1 support. All NDI applications running at least NDI version 3 libraries or higher can accept NDI|HXv1 signals.

NDI|HXv1 uses about one tenth the amount of network bandwidth as High Bandwidth NDI, making it suitable for bandwidth constrained connections. HXv1 is a Long-GOP encoded video stream, and by its nature will introduce some latency.

All NDI|HXv1 sources use H.264 image compression.

NDI|HXv2

NDI|HXv2 is similar to v1 in terms of bandwidth, quality and latency, but used where NDI was integrated during the device design process. This makes it possible for the full NDI feature set to be supported. No additional driver is required, but any software application must be using NDI version 4 libraries to have compatibility with NDI|HXv2 sources. HXv2 is a Long-GOP encoded video stream, and by its nature will introduce some latency.

NDI|HXv2 sources can use H.264 or HEVC image compression.

NDI|HXv3

This version uses a similar compression codec as HXv2 but achieves lower latency at the expense of much higher bandwidth. Typically, the device will give you the option to choose between HXv3 and HXv2, so you can choose between lower latency or reduced bandwidth operation. NDI|HXv3 requires NDI 5.5 libraries or higher for compatibility. HXv3 encoding scheme operates with a modified Long GOP structure, where the streams are mostly I-frame only or key-predicted (IP) frame pairs, while removing the bi-predictive (B) frames, which causes most of the latency in Long GOP streams.

NDI|HXv3 sources can use H.264 or HEVC image compression.

2.5 NDI Groups

2.5.1 What are NDI Groups?

NDI Groups allow you to logically segment NDI sources on your network, making them visible only to other devices configured in the same group. A device can be configured with multiple groups if needed, and you can set send and receive groups separately. This makes it possible to receive a signal from one group on a device which can process the signal and then make it available on a different group. A good example of group usage is if you had two studios, where all the equipment is on the same network. NDI Groups can make it so that Studio A cannot see Studio B (and vice versa), but if Studio A needed a resource from Studio B, all it would take is altering the group setting to make it available.

By default, all NDI devices are part of the **Public** send and receive groups. If you happen to delete all grouping information on a device, you will automatically be placed back into the Public group.

NDI Access Manager doesn't auto populate a list of groups to join, you will need to know the names to manually enter. This provides a method to hide sources from other devices on the NDI network, and only by knowing the group name can you gain access.

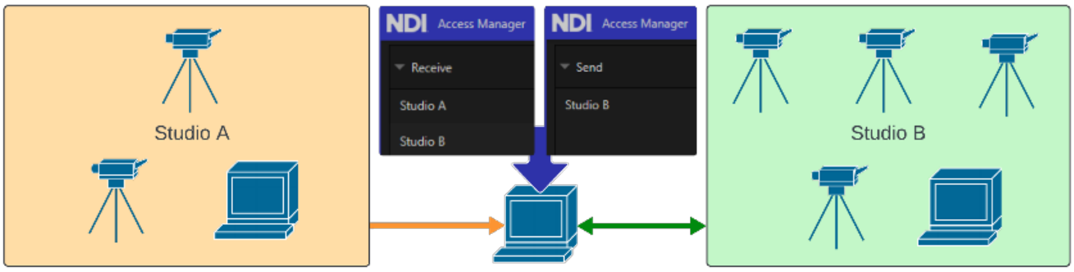
NDI Access Manager will affect all NDI signals generated on a device, unless the application is using the NDI Advanced SDK, which can provide the ability for unique NDI settings per channel output.

2.5.2 Send Groups

This is the group(s) in which signals are available. This is the type of group you configure on an NDI encoding device such as a camera.

2.5.3 Receive Groups

This is the group(s) in which you can see NDI signals. This is the type of group you configure on an NDI decoding device such as a monitor.



With the above NDI Group settings, the center device can receive from both studios and can send signals into Studio B.

2.6 NDI Routing

Some NDI applications use a concept called *routing* to connect signals between systems. These routing applications will create virtual endpoints, which operate a bit differently than normal NDI endpoints. Instead of creating or sending video, these routing applications direct destinations to the source they should pull a signal from. No video passes through the system running the routing application, all communication happens point to point between systems.

Having an NDI routing application on your network can make it easier to manage signal flows between systems. Instead of having to configure each endpoint using its local interface, you can direct NDI connections from a routing application instead. This can provide a centralized location to manage your entire NDI network. Some routing applications allow you to specify the virtual endpoint names, making it easier to select the correct router output.

NDI Tools includes a free NDI Routing tool, which is a viable option for smaller sized NDI networks. Third party routing options are available for larger installations, that can be interfaced with hardware control panels, making it easy for end users to route NDI signals just as they would SDI signals.

3. LAN Configuration

3.1 Switches and network topology

NDI is designed to operate using any switch with at least 1 gigabit Ethernet ports. It is highly recommended that a managed layer 2 switched by is used to handle NDI signals. Layer 2 switches provide additional configuration options that may be useful in many workflow configurations.

Recommended switch default settings and useful features:

- **Non-blocking design** – Describes a performance aspect of a switch. A non-blocking design means that the switch offers enough internal switching bandwidth allowing all network ports to be functioning at maximum capacity at the same time.
- **Ethernet Flow Control** – Recommend enabling. Allows better control of traffic flow to prevent data loss. Some switches provide a selection of symmetric or asymmetric flow control. Symmetric will affect communication in both directions, while asymmetric allows independent adjustment in either direction. Try symmetric first and adjust to asymmetric if needed.
- Jumbo frames needed to be turned off, in every device (senders, switches).
- **IGMP Snooping** – If you are going to use multicast sending, all switches on the network must have IGMP Snooping enabled.
- **EEE Configuration** – the EEE feature reduces power consumption, but some aspects might lower performance. It is recommended to disable EEE features.
- **PoE/PoE+** – If you plan on using devices that support power over Ethernet, having a switch with this feature will provide power to your device via ethernet cable. Verify power budget to make sure switch can supply enough current to all devices.
- **Quality of Service (QoS)** – NDI does not use DiffServ flags as of now, therefore it is recommended that QoS features to be disabled on the switch (or port based rules or similar could be applied to prioritize NDI traffic).
- **Port Configuration -** enables manual adjustment of network ports with settings such as link speed, duplex, jumbo frames size. Link speed should match what is expected, duplex must always be full, and it is not recommended to use jumbo frames with NDI traffic.
- Jumbo frames are any Ethernet packets exceeding the 1500 bytes size for the payload. This has to disabled in every NDI endpoint, and every switch port sending and receiving NDI traffic. While theoretically Jumbo frames will help to send data between sender and receiver units with less overhead, by allowing the sending of more payload data per packet. The additional size of the jumbo frames will take longer to transmit each packet over the wire, increasing the latency on the network.

If a single bit is corrupted due to noise or interference, a retranmission needs to occur. With jumbo frames, this process will take longer due to the larger frame size.

- The port configuration menu typically also gives access to VLAN configurations. It's always advisable to isolate different traffic in the network switch if you can't use different networks for different applications (e.g. NDI, Dante, control network, etc).
- **Port Mirroring** – a helpful troubleshooting feature for doing network packet captures. You can duplicate the incoming and outgoing traffic of Port X to Port Y, while running Wireshark on a device connected to Port Y. This enables you to troubleshoot a device without disconnecting it from its existing port.
- **Maintenance and Administration** – Logging features are essential for troubleshooting. Logs can report issues such as port flapping or dropped packets on a specific port (either inbound or outbound).
- **Simplified topology** – Use a star topology, with core and edge switches whenever possible and try to avoid daisy chaining networking equipment. Always work with the simplest network topology possible. Additional complexity might increase latency, which will affect your production network's performance.
- **Multiple speed fabrics** – While devices with higher speed network interface speeds are becoming more common (such as 10Gbit/s, 5 Gbit/s & 2.5G Gbit/s), make sure your switch also supports those speeds. Otherwise, the negotiated link speed will be limited to the version what both endpoints support instead of the desired speed. Never use different network speed interfaces on the same device to connect to the same subnet / VLAN of the NDI network.
- **Oversize the network switch** – get a larger switch than required for the workflow, this provides additional performance overhead and room for expansion. If the switch is precisely the size needed today, you will often be upgrading to a larger switch in the near future.

3.2 TriCaster

With the different TriCaster models available, let us start with a few common aspects:

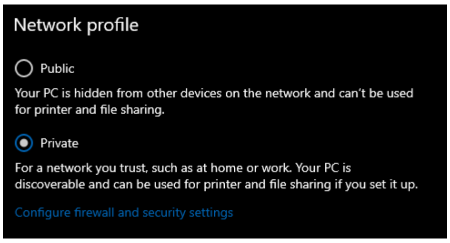
3.2.1 Common

- **Install NDI Tools**. TriCaster systems already have NDI capabilities, but installing NDI Tools on the system can be useful. This will let you do NDI tests outside the TriCaster application. Also, you will need NDI Tools to use any NDI|HXv1 devices.
- **Set network type to Private**. When you connect a Windows computer to a local network, the operating system will display a message asking for the network type. Different releases of Windows might ask this question in different ways, but ultimately this request is trying to determine how the network should be trusted, resulting with the network type to be either be *Public* or *Private*

- *Public* networks are considered less secure, which will cause Windows to increase the firewall protection of the system. In essence, it will try to limit the system to communication only to the Internet and not with any other devices on the local network. Think of the Wi-Fi at a coffee shop, you want to access the Internet, but not any of the other systems on the Wi-Fi at the coffee shop. If Public networking is configured, NDI communication will often be blocked.

- *Private* networks are considered more secure and will allow the computer to communicate easily with other devices on the local network. **This is the type of networking you want configured for all NDI clients.**

If this message is bypassed, the default will be Public. In those cases, open the *Internet & Network Settings* in Windows. Then click on Ethernet, find the interfaces being used and set them to Private.


Network profile

○ Public

Your PC is hidden from other devices on the network and can't be used for printer and file sharing.

● Private

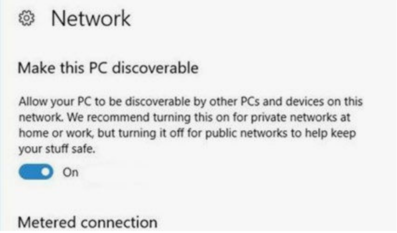
For a network you trust, such as at home or work. Your PC is discoverable and can be used for printer and file sharing if you set it up.

Configure firewall and security settings

Another aspect you may run into is an NDI network that is set up without a gateway (no access to the Internet). NDI can run on these networks, as Internet access is not required for NDI to operate on a local network. However, Windows will have a difficult time trying to determine the network type (which is referenced by the gateway information). In these cases, you will often need to manually adjust Windows to assign the network type correctly.

1. Press Window + R to run a program, type *gpedit.msc* and press return.
2. Navigate *to Computer Configuration > Windows Settings > Security Settings > Network List Manager Policies*
3. Open *Unidentified Networks*.
4. Change *Location type* from *Not configured* too *Private*.

Some versions of Window 10 might not display the Public/Private options as shown above, but instead provide a switch labeled *Find devices and content* or *Make this PC discoverable*, in this case you want to set this option to On, which make the system use a private network.


Network

Make this PC discoverable

Allow your PC to be discoverable by other PCs and devices on this network. We recommend turning this on for private networks at home or work, but turning it off for public networks to help keep your stuff safe.

On

Metered connection

- **Configure NDI Access Manager**. If any specific settings need to be configured for NDI, you can find NDI Access Manager in the Add-On section of the home screen. You can also use NDI Tools (if installed) for this purpose as well. Running either version of NDI Access Manager will update the same NDI configuration file on the system.

- **Update NIC drivers**. You can update TriCaster's NIC drivers, especially for any interfaces that operate at more than 1Gb speed. While 1Gb Ethernet is well established and updating the drivers may not make much difference, higher speed interfaces can benefit from having the most recent drivers.

- **Mixing different speed interfaces**. Many TriCaster models offer Ethernet interfaces at different speeds. For example, TriCaster TC1, 1 Pro and 2 Elite all have a 1Gb and 10Gb interface, while some TriCaster Mini models offer a 1Gb and 2.5Gb Ethernet interface.

  If multiple Ethernet interfaces are going to be connected to the same network subnet, the requirement is that all interfaces must be running at the same speed. Do not connect two dissimilar speed interfaces to the same subnet.

  If using transmission methods like UDP or Multi-TCP that balance the NDI traffic between interfaces, the lower performance interface will cap the amount of traffic that can be balanced between all interfaces. For example, connecting a 1Gb and 10Gb to the same subnet will give you approximately 2Gb of bandwidth (not 11Gb).

  Transmission modes like Single-TCP and Reliable UDP do not attempt to balance traffic between interfaces, but rather send each connection over a single Ethernet interface. The NIC which is chosen is determined by the operating system network stack and may not be the one with the most available bandwidth. This means that more traffic could be directed to the interface with less bandwidth if you have mixed speed interfaces.

  This multi-link connection only applies to the same subnet. If the TriCaster is connecting to two completely different networks, then you can connect the 10Gb NIC to the primary network and the 1Gb port to the secondary network without issue.

- **Expanding TriCaster with more NIC interfaces**. There may be times when additional network interfaces are needed for a TriCaster system. A USB3 to Ethernet interface can be used for this purpose. The most common are USB3 to 1Gb Ethernet, but interfaces do exist for connecting with 2.5Gb or 5Gb Ethernet. Be aware that many of the 5Gb Ethernet USB3 interfaces will be limited by the USB chipset and likely not achieve a full 5Gb of performance.

  When possible, use on-board hardware Ethernet interfaces for NDI workflows before using USB interfaces. USB does add overhead in the network communication, making

PCI Ethernet based interfaces more efficient.

- **Limiting NDI traffic to a specific NIC interface.** When using mDNS discovery, NDI will broadcast signal availability on all connected interfaces to the TriCaster system. This is the inherent operation of mDNS, which is designed to be simple and automatic. If you have multiple networks connected to your TriCaster and you want to keep NDI contained to a specific network, the best option is to use Discovery Server.
  When Discovery Server is enabled, the TriCaster system will no longer broadcast NDI availability to all network interfaces, but instead work through the managed Discovery Server application. Run Discovery Server on your NDI network and configure all clients to connect to it using NDI Access Manager.

  If more control over the network adapters is required, select a network using the *Preferred NIC* option in NDI Access Manager. This option will determine which Ethernet interface should be used for advertising signals. Use this option when connected to a Discovery Server, trying to use this function with mDNS discovery will often not lead to the results desired.

  Even with Discovery Server enabled, systems will continue to see any mDNS discovered NDI sources as well. If the path to an mDNS source is only available on a particular network, it will be used when selected. If you have a Layer 2 switch it may be possible to add an Access Control Entry to block mDNS communication in the switch if you want all devices to only use Discovery Server to find sources.

3.2.2 TriCaster Bandwidth

TriCaster systems have features that can reduce the network bandwidth used when working with NDI, which can often make these systems more efficient than expected. This can be very helpful when TriCaster is used for remote or cloud production, as it allows more sources to be used over the often bandwidth limited WAN connection.

All NDI sources are available in full resolution and proxy resolution. There are some software applications that allow you to choose which resolution is used. NDI Studio Monitor has a *Low Bandwidth* option which will use force the proxy stream for display. Regardless of the source resolution, the proxy stream is 640 pixels wide by the corresponding height as determined by the aspect ratio (360 pixels tall for 16x9 widescreen video).

In the case of TriCaster, bandwidth will dynamically shift between full and proxy resolution sources determined by its use in the switcher. TriCaster will always use the proxy bandwidth from a source, except for these functions that will determine when full resolution video will be requested:

- **Any source directed to a MIX output will use full resolution.** By default, this would be anything on Program output, but will include anything else that is manually directed to a MIX output. If Preview or a M/E is selected for a MIX output, then any sources used in these buses will be full resolution. Also, during a transition, sources on Program and Preview are currently both visible on a MIX output, meaning that all of these sources are full resolution. Keep this in mind with M/Es, if you had two M/Es, each with four NDI video layers, performing a transition between them would place eight NDI channels in full resolution at once for that short period of time.

- **Any source visible in a Multiview over a certain size will use full resolution.** Typically, Multiview viewports on the primary display are less than 640 pixels wide. However, full screen multiviewers at 1080p resolution set for full screen or with a 2x2 display will stream full NDI resolution from each source. When using a 4K monitor on TriCaster, multiviewer viewports will often be more than 640 pixels wide, causing additional NDI sources to stream at full resolution. Make sure that you do not operate more 4K computer displays than are supported on your TriCaster model.

- **Any input source being recorded will use full resolution.** When recording an input source (this doesn't apply for recording MIX outputs), a full resolution NDI stream will be used. This is in addition to the bandwidth for the live video stream. Native NDI recording will spawn an additional task, meaning that there are two destinations pulling the NDI connection: the TriCaster for live video switching and the recording task. If you are recording an input and also have the same source on a MIX output, then double the source bandwidth will be needed during this time.

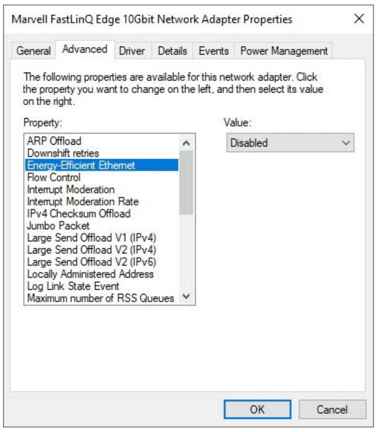
Another bandwidth saving feature on TriCaster is using the same NDI source in multiple inputs. A 4K camera can be connected to multiple TriCaster inputs, which can then be adjusted using the Pan and Scan tool (also called Region of Interest) for creating multiple shots. In this situation, the same NDI source signal is shared between all inputs, allowing a single 4K source to be used 3, 4 or more times without using additional network bandwidth.

NDI Bandwidth charts are available at the NDI website. Bandwidth (ndi.video) NDI uses VBR encoding, the values listed are typically maximums.

Often bandwidth values will be a bit less than shown, but it is possible for short spikes to be above these values. Some NDI devices support adjustable compression values. Check your device to see how it is configured.

3.2.3 TriCaster Specific Settings

- **Marvell 10Gb NIC settings**. If you have a system with a Marvell/Aquantia 10Gb NIC interface (TriCaster TC1, 1 Pro & 2 Elite), the following options should be set in the Ethernet driver. If you perform a driver update, these settings may be reset to defaults.
    - ○ Disable IPv6
    - ○ Disable Downshift retires
    - ○ Disable Recv Segment Coalescing (IPv4)
    - ○ Disable Recv Segment Coalescing (IPv6)
    - ○ Disable Energy Efficient Ethernet
    - ○ Set Transmit Buffer to 4096
    - ○ Set Receive Buffer to 4096


Marvell FastLinQ Edge 10Gbit Network Adapter Properties
General
Advanced
Driver
Details
Events
Power Management
ARP Offload
Downshift retries
Energy-Efficient Ethernet
Flow Control
Interrupt Moderation
Interrupt Moderation Rate
IPv4 Checksum Offload
Jumbo Packet
Large Send Offload V1 (IPv4)
Large Send Offload V2 (IPv4)
Large Send Offload V2 (IPv6)
Locally Administered Address
Log Link State Event
Maximum number of RSS Queues
OK
Cancel

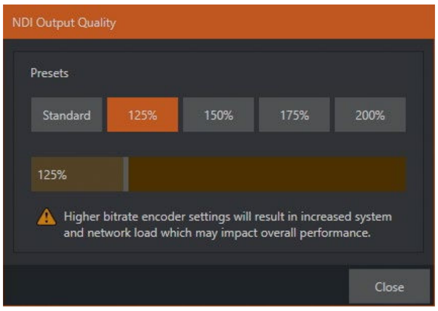
The following properties are available for this network adapter. Click the property you want to change on the left, and then select its value on the right.

Property: Energy-Efficient Ethernet

Value: Disabled

- **TriCaster Mini 4K DHCP Server**. This model comes with a built-in *mini* DHCP server application to allow direct-connect of NDI devices to the system. This service is enabled by default. The mini DHCP server will be disabled on ports when an external DHCP server is detected. But, depending on the network complexity, this may not occur immediately, which will cause the TriCaster Mini to configure devices on the network. You can manually disable the DCHP server from the TriCaster's Administration panel. In this situation, if you are using any of the NIC ports for direct NDI device connection, then you will need to set an appropriate static address for each NIC port and external device.

- **TriCaster Vectar NDI Output Quality.** These systems can adjust the encoding quality of the NDI signal, using the NDI Output Quality slider found on the home screen (inside the NDI-KVM menu). Adjusting value will affect not only the network bandwidth, but also the CPU resources required for encoding and decoding in other NDI devices, it is advisable to perform adequate testing when adjusting this value. This feature is available on TriCaster Vectar systems, which are run in the cloud with high-speed bandwidth connections and on virtual systems which are easier to adjust performance via deployable instance types.



Higher bitrate encoder settings will result in increased system and network load which may impact overall performance.

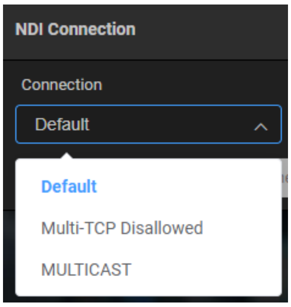
3.2.4 Viz Connect Solo Configuration

Viz Connect Solo units – previously called NewTek Spark Plus – are a popular method to add additional baseband I/O interfaces to a TriCaster.

All Connect Solo units are designed to operate with High Bandwidth NDI signals. This is for both encoding and decoding. The NDI Bridge software, mentioned in more detail later in this document, can be used to convert signals into and out of NDI|HX if that is required for compatibility with Viz Connect Solo.

Connect Solo allows for a selection of NDI transmission options, the charts below will explain what will be used based on the selections. The *Default* selection should work in most cases, but there may be times when choosing a specific mode will improve operation.

For a Connect Solo setup as an *Encoder* mode, the NDI receiver transmission is listed on the left (the setting found in NDI Access Manager) and the Connect Solo send options are listed across the top.



NDI Connection

Connection

Default

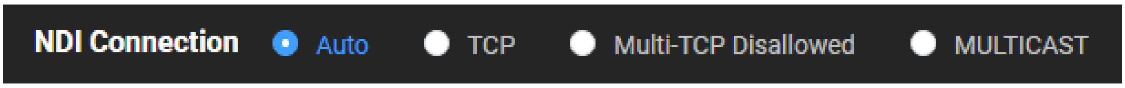
Default

Multi-TCP Disallowed

MULTICAST

	Default	Multi-TCP Disallowed	Multicast
Auto	Multi-TCP	UDP	Multicast
Reliable UDP	Single-TCP	Single-TCP	Multicast
Multi-TCP	Multi-TCP	Single-TCP	Multicast
(Unicast) UDP	UDP	UDP	Multicast
Single-TCP	Single-TCP	Single-TCP	Multicast

If the Connect Solo is configured as a *Decoder*, use the chart below to determine the transmission which will be used. The default option is *Auto* and should work in the majority of cases. The Connect Solo options are listed on the left and the top of the chart shows how the NDI sender has been configured. Notice that it is possible to set a Connect Solo to receive a unicast stream, even if the sender has been enabled for multicast sending.



NDI Connection ● Auto ○ TCP ○ Multi-TCP Disallowed ○ MULTICAST

	Unicast Sender	Multicast Sender
Auto	Multi-TCP	Multicast

TCP	Multi-TCP	Multi-TCP
Multi-TCP Disallowed	Single-TCP	Single-TCP
Multicast	UDP	Multicast

3.2.5 Viz Connect Configuration

Another popular expansion solution with TriCaster is Viz Connect.

On these products, use NDI Access Manager to determine NDI configuration options like send/receive groups, receive selections, or transmission modes.

All Connect units will encode High Bandwidth NDI and have the ability to decode any incoming NDI type (including all NDI|HX formats). Be aware that NDI|HX decoding is accelerated by GPU hardware, while most Connect units were designed for High Bandwidth NDI workflows. The maximum number of NDI|HX decoding channels will depend on the source aspects (video resolution, framerate & codec type).

All Connect units have multiple Ethernet ports, the maximum speed of the NIC will depend on the Connect model. It is recommended to use the fastest Ethernet speed available, especially if you will be working with 4K/UHD video formats.

If you are operating with only HD video formats, workflows of up to 8 HD video channels will typically work over a single 1 Gb Ethernet connection when the destination is a TriCaster. This is due to the proxy/full resolution workflow operation offered by TriCaster.

With any other NDI destination (like a *Viz Connect to Viz Connect* workflow), do not expect more than 6 HD channels of video when using a single 1 GbE port. Should you require more than 6 channels, utilize both Ethernet ports which operate at 1 GbE, or for improved performance, opt for the interface with a higher speed. In the cast of dual 1 gig Ethernet, a receive mode of Multi-TCP or UDP will result in the best transmission of data across both network interfaces.

	Video I/O	Primary NIC Speed	Secondary NIC Speed
Viz Connect Tetra	4x 12G-SDI	2.5 GbE	1 GbE
Viz Connect Studio I/O	2x 12-SGI & 6x 3G-SDI	10 GbE	1 GbE
NewTek Connect NC2	2x 12-SGI & 6x 3G-SDI	10 GbE	1 GbE
NewTek Connect NC1	8x 3G-SDI	1 GbE	1 GbE

4. Remote & Cloud Configuration

4.1 Router

Router configuration is often required when working with remote and cloud configurations. The user interface for routers can vary. A good resource is PortForward.com which can assist with configuration or many different router manufacturers.

4.2 NDI Bridge

4.2.1 Overview

NDI Bridge is utilized to connect NDI between remote locations or to the cloud. One side of NDI Bridge will be configured as the *host* and can accept multiple connections from NDI Bridge systems running in *join* mode. All traffic will go through the host system if multiple sites are joined, so the host should be configured at the location with the most bandwidth capability.

The default port number for NDI Bridge is 5990, or this can be manually set in the NDI Bridge interface. Only one port needs to be configured for any number of NDI connections, from any number of external NDI Bridge join system. All traffic will use UDP packets.

The connections for NDI Bridge are password protected and encrypted.

Other than TriCaster Now installations, NDI Bridge should be run on a system separate from the TriCaster hardware itself.

NDI Bridge running in Join mode can be running on a Viz Connect Tetra system, while all other Connect models should work with NDI Bridge running on a separate computer.

4.2.2 Buffering and Synchronization

NDI Bridge offers a Buffering control to account for high latency WAN connections. A good rule of thumb is to set this value to at least triple the ping value between locations. This will provide additional time for resending of packets to avoid dropping frames.

All sources sent through NDI Bridge are multiplexed into a single transmission. This ensures synchronization across all signals, eliminating concerns about varying effects on individual transmissions across the Wide Area Network.

Any video offset between streams likely is generated this way at the source. If the cameras have a genlock input, use this feature to lock all cameras to a signal master clock. Verify that any unnecessary devices in the signal chain are removed, which could be adding latency to the signal path. If none of these aspects provides a resolution, TriCaster has per input video delay options. Determine which video input is the most delayed and then add additional delay on the other inputs to match.

TriCaster offers separate video and audio delay settings on every input. Don't forget to match the same delay with the audio or use this setting for A/V sync adjustment.

4.3 NDI High Bandwidth & NDI|HX Considerations

NDI Bridge can be configured to support a variety of compression codecs. Which codec to use will depend on the bandwidth and resources available to the system. All traffic will output in the selection chosen. While NDI High Bandwidth is an option, most NDI Bridge connections will use NDI|HX to allow more control over the bandwidth and the maximum number of NDI connections possible.

In planning a system to run NDI Bridge, the following aspects need to be considered:

- **Network Bandwidth** – The Ethernet interfaces on the system need enough bandwidth to handle all incoming and outgoing streams on the local network, along with the WAN connection being able to handle the encoded data between nodes. NDI Bridge does provide an estimate of WAN bandwidth in both directions. Use Windows Task Manager to monitor local network bandwidth.

- **CPU** – any High Bandwidth NDI signals will be encoded/decoded using CPU resources. A fast, multi-core CPU will benefit more of these types of NDI signal being used. In general, an Intel i7 CPU or higher is recommended.

- **GPU** – any NDI|HX signals will be encoded/decoded using GPU. In the case of decoding, a suitable decoder must be available in Windows. If a decoder is not present, the HX signal will be passed through in its native format with no changes.

  Be aware that some GPU's have limits on the number of simultaneous encodes they can process. When possible, Nvidia cards are recommended for NDI Bridge, with most Nvidia Quadro cards having unrestricted encoding and GeForce card being limited to 8 encodes maximum. The maximum amount of encoding possible will depend on the video signal specs (resolution & frame rate) along with the type of GPU being used. If this encode limit is reached, any additional connections will default to High Bandwidth NDI being sent.

- **Bridge Output** – all incoming encoded sources will be sent out of Bridge, on the far side, in the encoded format. NDI Bridge does not perform additional transcoding for these output signals. If you have a device that can only accept High Bandwidth NDI or NDI|HX as an input type, keep this in mind for the workflow design.

4.4 Remote Control Surface Operation

- **TriCaster Flex & Flex Dual** – these control panels operate via NDI. NDI Bridge will provide the required connectivity for the control surface to connect with the TriCaster. Make sure the TriCaster is in a session and NDI Bridge is operating, then you can connect these surfaces just like you would to a local, on premise TriCaster.

- **2-Stripe, 4-Stripe & USB panels** – these panels can operate with remote and cloud-based TriCaster systems using the *Viz Remote Control Surface Utility* available for Windows. Running this tool will allow one of these surfaces to connect using port 5958/TCP. Control surface compatibility with the TriCaster model still applies when using this application.

4.5 Cloud Configurations for Secure and Efficient NDI Workflows

When running NDI workflows in the cloud, it is essential to configure the network infrastructure properly to ensure secure, efficient, and reliable transmission of high-quality video streams. Key considerations include security groups, subnet communication, VPC (Virtual Private Cloud) communication, VPC peering, routing, and data transfer.

4.5.1 Security Groups

Security groups act as virtual firewalls that control inbound and outbound traffic to your instances. Properly configuring security groups is crucial for securing NDI flows:

- **Inbound Rules** – Allow only necessary traffic, such as video and control data, by specifying the appropriate ports (refer to section 6.4 for NDI Network Port List).
- **Outbound Rules** – Restrict outbound traffic to required destinations, reducing the risk of unauthorized data transmission.
- **Granularity** – Use fine-grained rules to define traffic sources and destinations, enhancing security by limiting exposure to only trusted IP addresses and networks.

4.5.2 Subnets Communication

Subnets are segments within a VPC that group instances based on network requirements and security needs:

- **Public vs. Private Subnets** – Place instances running NDI in private subnets to enhance security, limiting direct access from the internet.
- **NAT Gateway** – Use NAT (Network Address Translation) gateways to allow instances in private subnets to access the internet securely for updates or other necessary communications.

- **Subnet Routing** – Ensure that subnets are configured with appropriate route tables to facilitate communication between instances transmitting NDI.

4.5.3 Network Access Control Lists (ACLs)

Network ACLs provide an additional layer of security at the subnet level.

- **Configuration**: Set up ACLs to allow the necessary traffic for NDI flows while blocking unwanted traffic (refer to section 6.4 for NDI Network Port List).
- **Best Practices**: Use network ACLs to complement security groups, providing a more granular level of control over the traffic entering and leaving your subnets.

4.5.4 VPC Communication

VPC communication involves setting up and managing multiple VPCs to handle different segments of your network infrastructure:

- **Inter-VPC Communication** – Use VPC peering or Transit Gateways to enable communication between instances in different VPCs, ensuring low latency and secure data transmission.
- **Isolation** – Keep critical NDI components in separate VPCs to isolate them from less secure or unrelated workloads, enhancing security and manageability.

4.5.5 VPC Peering

VPC peering establishes a direct network route between two VPCs, enabling instances in different VPCs to communicate as if they were within the same network:

- **Peering Connections** – Set up VPC peering connections to facilitate direct, low-latency communication between NDI instances in different VPCs.
- **Routing Tables** – Update route tables in both VPCs to include routes to the peered VPC, ensuring seamless traffic flow.
- **Security Groups and ACLs** – Ensure that security groups and network ACLs (Access Control Lists) allow traffic to flow between the peered VPCs.

4.5.6 Routing

Routing involves configuring the paths that network traffic follows to reach its destination:

- **Route Tables** – Ensure route tables are configured to direct traffic efficiently within and between VPCs. This includes setting up routes for subnets and peered VPCs.
- **Transit Gateways** – Use Transit Gateways for centralized and scalable routing management, especially when connecting multiple VPCs or on-premises networks.

4.5.7 Data Transfer

Data transfer considerations are critical for optimizing performance and cost when running NDI workflows:

- **Intra-Region vs. Inter-Region** – Transfer data within the same region to minimize latency and reduce data transfer costs. Inter-region transfers are subject to higher latency and additional costs.
- **Optimizing Data Flow** – Use services like Direct Connect or dedicated inter-region connectivity options to optimize data transfer for NDI workflows, ensuring high bandwidth and low latency.
- **Monitor Egress Costs** – Some cloud providers charge for data egress, which is transferring data from their network to the Internet or other external networks. Monitoring egress costs is essential as these charges can accumulate quickly, especially with NDI High Bandwidth.

  Understanding and managing egress costs can significantly impact your overall cloud expenses. If necessary, use compression such as NDI-HX to reduce bandwidth usage without compromising quality (see section 2.4.2 – NDI-HX and section 4.2 – NDI Bridge for detailed information).

4.6 mDNS in the Cloud

mDNS does not work in AWS and other cloud provider environments, which has implications for NDI workflows when running in this kind of environment.

Cloud networking is primarily designed to support unicast traffic. Broadcast and multicast traffic, which mDNS relies on, are typically unsupported in cloud environments. This is due to the scalability and security concerns associated with allowing such traffic to propagate across the cloud provider's network infrastructure.

Cloud providers such as AWS, Google GCP, Microsoft Azure, and others operate on highly segmented and isolated network architectures. Instances and services are often separated into different virtual private clouds (VPCs) and subnets, with strict network security rules that prevent the broadcast and multicast traffic required by mDNS.

The following solutions can be implemented in cloud environments:

- **NDI Discovery Server** – As mentioned in section 2.2.3 of this document, the best solution for enabling NDI in cloud environments is to use the NDI Discovery Server. The NDI Discovery Server acts as a centralized directory service, allowing NDI devices to register and discover each other without relying on mDNS. This approach is cloud-friendly and overcomes the limitations imposed by the cloud provider's network architecture.

- **Multicast** – As mentioned in section 2.3.3, creating a Multicast Domain and a Multicast Group within the cloud environment can also be a solution. This involves configuring the cloud network to support multicast traffic in a controlled and secure manner, enabling devices to discover each other through multicast.

4.7 Using NDI in Cloud Providers' Global Networks

4.7.1 Introduction

Cloud providers operate extensive global networks to connect their data centers, availability zones, and regions. These networks ensure high performance, low latency, and robust security for seamless cloud service operations worldwide. Usually, a Global Cloud Provider Network is presented as follows:

- **Data Centers** – These house the physical servers and networking equipment. Strategically located worldwide, they ensure redundancy, high availability, and data sovereignty.
- **Availability Zones (AZs)** – Distinct locations within a region with independent power, cooling, and networking. Multiple AZs per region enhance fault tolerance and service availability.
- **Regions** – Geographic areas containing multiple AZs. Multiple regions offer geographic diversity, compliance with local regulations, and improved latency. Each operates independently to ensure service continuity.
- **Inter-Region Connectivity** – High-speed fiber-optic networks, or backbone networks, connect regions. AWS, GCP, and Azure heavily invest in these networks, ensuring high bandwidth and low latency for data traffic.
- **Edge Locations** – Also known as Points of Presence (PoPs), these reduce latency and improve performance. They support content delivery networks (CDNs), caching, and low-latency access to cloud services globally.

4.7.2 Transmitting NDI from On-Premises to the Cloud using Dedicated Connectivity

When transmitting NDI from on-premises environments to the cloud, dedicated connectivity options such as Google Cloud Interconnect, AWS Direct Connect, and Azure ExpressRoute offer several significant advantages over standard internet connections. These dedicated connections ensure higher performance, reliability, and security, which is crucial for high-quality video transmission.

- **Reduced Latency** – Dedicated connectivity provides low-latency connections by establishing direct links between your on-premises network and the cloud. This minimizes the number of hops and optimizes the routing path, significantly reducing latency. This is particularly beneficial for real-time NDI streams where even minor delays can be critical.

- **Increased Bandwidth** – Dedicated connectivity supports high-bandwidth connections, enabling the transmission of high-resolution NDI streams without heavy compression. This is crucial for maintaining video quality and ensuring smooth transmission. With scalable bandwidth options, these connections allow for the simultaneous transmission of multiple NDI High-Bandwidth streams.
- **Enhanced Security** – Dedicated connectivity ensures data security by using private, dedicated connections, reducing the risk of interception or data breaches that can occur over the public internet. This enhances security by keeping your data off the public internet, making dedicated lines less susceptible to attacks and unauthorized access.
- **Reliability and Consistency** – Dedicated connectivity offers reliable and consistent network performance by avoiding the variability and congestion of the public internet. This ensures a steady stream of NDI data without interruptions, with dedicated connections ensuring reliable transmission of NDI streams, free from the fluctuations common with public internet connections.
- **Scalability** – Dedicated connectivity easily scales to accommodate increasing bandwidth demands, allowing you to transmit more NDI streams or higher-resolution content as business needs grow. With flexible bandwidth options that can be adjusted based on business requirements, these connections support scalability as NDI transmission needs evolve.
- **Cost Efficiency** – Dedicated connectivity can be more cost-effective over time by reducing egress charges associated with data transfer over the public internet, especially for high-volume NDI transmissions. It reduces data transfer costs compared to standard internet usage, providing a more economical solution for large-scale NDI deployments.

4.7.3 Transmitting NDI High Bandwidth with Cloud Inter-Region Connectivity

Transmitting NDI High Bandwidth using inter-region connectivity within cloud providers' internal networks, such as AWS, offers significant advantages over using the public internet. This method ensures higher performance, better security, and lower latency, crucial for high-quality video transmission. However, there are essential factors to consider when optimizing the performance of NDI High Bandwidth streams across different regions.

Same Geographic Area (e.g., US Regions)

The latency is relatively low when transmitting NDI High Bandwidth between regions in the same geographic area, such as AWS US regions. For instance, the latency between us-east-1 (Northern Virginia) and us-west-2 (Oregon) is generally around 60-80 milliseconds. While this is low, even small delays can be significant for NDI. Important considerations:

- **Bandwidth Requirement** – NDI High Bandwidth can require several hundred Mbps, depending on the resolution and frame rate. AWS provides high-bandwidth connections between regions but ensuring that the network can handle the uncompressed NDI stream is crucial.

- **Network Conditions** – The performance of NDI streams depends on network conditions like jitter and packet loss. Cloud Providers like AWS's infrastructure minimize these issues, but they cannot be eliminated. Cloud Providers provide tools such as AWS Reachability Analyzer and CloudWatch to monitor network conditions.
- **Optimize Network Settings** – Instances or VMs can be configured for optimal network performance by enabling enhanced networking features and using instance types that support high network throughput.
- **Monitor and Adjust** – Continuously monitor network performance and make necessary adjustments. AWS CloudWatch provides valuable metrics to help to maintain optimal performance.

Different Geographic Areas (e.g., AWS US to Asia Regions)

When transmitting NDI High Bandwidth across different geographic areas, such as from US regions to Asia regions, the latency increases significantly due to the greater distance and the number of network hops. The estimated latency in this scenario is around 150-200 milliseconds. Important considerations:

- **Bandwidth Requirements** – NDI High Bandwidth can require several hundred Mbps, depending on the resolution and frame rate. Cloud Providers provide a robust infrastructure with high bandwidth capacity, but intercontinental links may pose additional challenges.
- **Latency Impact** – A latency of 150-200 milliseconds can significantly affect real-time video interactions.
- **Network Conditions** – Factors like jitter and packet loss are more pronounced over longer distances. To maintain NDI performance, Cloud Providers provide monitoring tools to verify high-quality network conditions.
- **Optimize Network Settings** – Instances or VMs can be configured for optimal network performance by enabling enhanced networking features and using instance types that support high network throughput.
- **Monitor and Adjust** – Continuously monitor network performance; necessary adjustments are always needed. AWS CloudWatch provides valuable metrics to maintain optimal performance.
- **Consider Compression**: For long-distance transmission, using NDI Bridge for local transcode and employing NDI|HX (compressed streams) might be more suitable. NDI|HX reduces bandwidth requirements while maintaining good quality.

4.7.4 Reducing Latency for NDI Bridge with Edge Cloud Services

Several advanced cloud and edge computing services are specifically designed to bring computing, storage, and networking resources closer to end-users, thereby significantly reducing latency. Key services such as AWS Local Regions, AWS Wavelength Zones (utilizing 5G networks), Google Distributed Cloud Edge (GDC), AWS Outposts, and Akamai Gecko exemplify this approach. By leveraging these services, NDI Bridge performance can be greatly enhanced through minimized latency, improved reliability, and high-quality video transmission.

Below are some examples of NDI Bridge Achieving Low Latency with Cloud Edge Services:

4.7.4.1 AWS Local Regions

AWS Local Regions are smaller geographic areas that provide localized infrastructure to reduce latency for specific regions. Running NDI Bridge in AWS Local Regions offers several benefits:

By processing data closer to the source, AWS Local Regions significantly reduce round-trip time, ensuring smoother and more responsive video streaming.

For example, the AWS Local Zone in Los Angeles (within the larger AWS US West (Oregon) region) provides localized infrastructure to reduce latency for NDI Bridge hosts or join in Southern California.

4.7.4.2 AWS Wavelength Zones

AWS Wavelength Zones integrate AWS compute and storage services within 5G networks, providing ultra-low latency and high bandwidth.

- **Ultra-Low Latency** – Wavelength Zones drastically reduce latency by placing AWS infrastructure at the edge of 5G networks, which is essential for NDI streaming.
- **High Bandwidth** – 5G connectivity offers significant bandwidth, supporting NDI over NDI Bridge without issues.
- **Restricted Access** – Only devices connected via the 5G network can access AWS Wavelength Zones. This restriction provides an added layer of security, as it limits the potential attack surface. Unauthorized devices or those connected through other means cannot access the infrastructure, reducing the risk of malicious activities.
- **Carrier-Level Security** – The integration with 5G networks means that AWS Wavelength Zones benefit from the carrier-grade security measures implemented by telecommunications providers. This includes encryption, authentication, and other security protocols inherent to 5G networks, ensuring that data is protected as it travels from devices to the Wavelength Zone.
- **Mobility** – Ideal for mobile and remote production setups, leveraging the expansive coverage of 5G networks.

MTU Configuration Considerations

When working with NDI in AWS Wavelength Zones, attention to Maximum Transmission Unit (MTU) configurations is crucial. The MTU defines the largest packet size that can be transmitted over a network without needing to be fragmented.

- **Optimizing MTU Settings**: The default MTU size for most networks is 1500 bytes. However, 5G networks and AWS infrastructure might support different MTU sizes. It is essential to verify and adjust the MTU settings to match the network's capabilities, ensuring efficient packet transmission and avoiding fragmentation, which can lead to increased latency and decreased performance.
- **Avoiding Fragmentation**: Packet fragmentation can cause delays and affect the performance of NDI streams. By configuring the MTU correctly, you can ensure that packets are transmitted in their entirety, reducing the likelihood of fragmentation and maintaining the quality and performance of NDI video streams.

4.7.4.3 Google Distributed Cloud Edge (GDC)

Google Distributed Cloud Edge (GDC) extends Google Cloud's services to edge locations, enabling localized data processing:

- **Localized Processing** – GDC processes data close to where it is generated, significantly reducing latency and improving NDI Bridge performance.
- **Scalability** – Seamless integration with Google Cloud services allows for scalable NDI solutions that can grow with demand.
- **Enhanced Security** – Localized data processing enhances data security by minimizing data travel over the public internet.

Example Scenario: Consider a live sports event in a stadium equipped with Google Distributed Cloud Edge (GDC) infrastructure. NDI Bridge can transmit live video feeds with minimal latency to remote production teams using Google GCP Services.

4.7.4.4 AWS Outposts

AWS Outposts bring AWS infrastructure and services to on-premises locations, offering a truly hybrid cloud solution:

- **Consistent Low Latency** – By extending AWS to on-premises environments, Outposts ensure consistent low-latency performance for NDI Bridge.
- **Seamless Integration** – Integration with the full range of AWS services allows for robust and flexible NDI deployments.

- **Customizable** – Outposts can be tailored to specific needs, ensuring optimal performance for demanding NDI applications.

Example Scenario: Imagine a television studio utilizing AWS Outposts to run NDI Bridge for live broadcasts. The low-latency environment provided by Outposts ensures that video feeds from various cameras and production equipment are synchronized in real-time, allowing for seamless live editing and broadcasting without any lag.

The television studio can customize their Outposts deployment to include specific compute, storage, and networking resources tailored to their NDI workflow, ensuring maximum performance and reliability during live productions.

4.7.4.5 Akamai Gecko

Akamai Gecko is a platform designed to deliver low-latency and high-performance computing at the edge:

- **Edge Processing** – Akamai Gecko processes data at edge locations, drastically reducing latency for NDI Bridge.
- **Global Reach** – Akamai's extensive global network ensures reliable, high-quality video transmission across different regions.
- **Performance Optimization** – Advanced routing optimizes the performance of NDI stream flow, ensuring minimal delay.

Akamai Gecko's routing optimization technology ensures that all video streams take the most efficient path through the network. The NDI Bridge can manage and transmit these video feeds with extremely low latency using Akamai Gecko's edge processing.

4.8 NDI Tools Deployment with Infrastructure as Code (IaC)

Infrastructure as Code (IaC) is a powerful approach that allows companies to provision and manage cloud resources using machine-readable configuration files. Deploying NDI tools using IaC brings numerous benefits, including consistency, repeatability, and scalability. NDI Tools are fully supported for IaC, enabling seamless integration into your cloud infrastructure. By treating infrastructure configuration as code, it is possible to automate the deployment of NDI tools, ensuring a streamlined and efficient setup process.

Some tools for IaC are: **Terraform** and **AWS CloudFormation**.

For more detailed information on deploying infrastructure using Cloud Infrastructure as Code (IaC), please refer to the respective cloud provider's or vendor's IaC documentation available online. Detailed resources are available on AWS Cloudformation and Azure IaC

Documentation. Comprehensive documentation for NDI Tools can be found at NDI Tools Documentation, which provides essential guidance on installation, configuration, and best practices for using NDI technology.

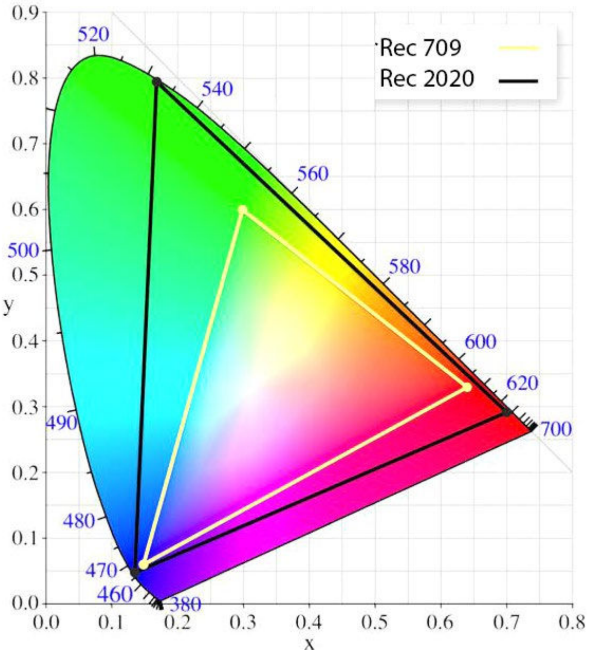
In Viz Now, Vizrt's cloud automation platform, NDI Tools are deployed using Terraform Infrastructure as Code (IaC). This approach ensures consistent, repeatable, and scalable deployments of NDI technology in cloud environments. For more detailed information and documentation, please refer to the Vizrt website.

5. HDR Productions

NDI version 6 adds support for HDR workflows, supporting wide gamut color primaries and HDR transfer functions, including HLG and PQ.

Creating HDR footage requires applications using the NDI Advanced SDK, while either the Advanced SDK or standard SDK can decode footage using NDI 6 libraries. Applications using previous versions of NDI will not be able to display the video, and instead show an image explaining that the HDR footage cannot be decoded until the NDI libraries are updated to version 6.

Currently, TriCaster systems support 8-bit SDR color formats. In the future, we should see TriCaster's with HDR support, at which point, this document will be expanded to include information about live HDR video production.



6. Troubleshooting

6.1 Determining the issue

When an issue occurs using NDI, one of the first aspects that needs to be determined is if it is a discovery, transmission or local computer issue. While there is some overlap, most resolutions are specific to these aspects, understanding which category it falls in can help with finding a solution.

Discovery issues involve seeing the source appear in the list of NDI sources. If you can’t find the source to select, then you are dealing with a discovery issue.

Transmission issues involve anything that occurs after you have selected the source. If after selecting the source on the NDI list, you get no video or the video/audio is incorrect in some manner, then you are likely dealing with a transmission issue.

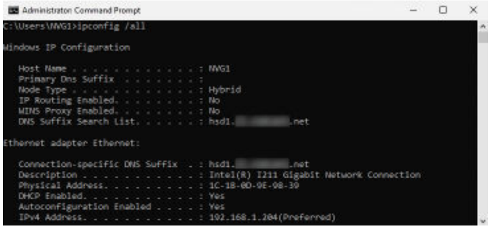
Another possibility could be issues caused by GPU drivers or system configuration. In these cases, NDI is working correctly but something on the local system is preventing the display of the NDI output. A good test is to always try a second, differently configured system. If the signal works somewhere else, then it is a good indication that the NDI source is operating correctly, and the issue is specific with the destination system.

6.2 Troubleshooting Tests

Discovery Issues

Addressing discovery problems is one of the most common topics Vizrt support and Professional Services teams address on a day-to-day basis. Here are many aspects you can look into.

- On Windows based senders, set both the sender and receiver NDI network interface to Private networking. This step alone usually solves most of the discovery issues, sometimes even transmission issues.
- Verify both devices are on the same subnet. Use `ipconfig /all` in the Windows command line, or `ifconfig` in Unix-like operating systems in the terminal to determine if the IP address and subnet of the network interface are in range of each other.



```
C:\Users\NVG1>ipconfig /all

Windows IP Configuration

   Host Name . . . . . . . . . . . . : NVG1
   Primary Dns Suffix  . . . . . . . :
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No
   DNS Suffix Search List. . . . . . : hsd1.[redacted].net

Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . : hsd1.[redacted].net
   Description . . . . . . . . . . . : Intel(R) I211 Gigabit Network Connection
   Physical Address. . . . . . . . . : 1C-1B-0D-9E-98-39
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   IPv4 Address. . . . . . . . . . . : 192.168.1.204(Preferred)
```

- Does mDNS work for other devices? Try to discover other sources on the network, test this by starting a new application on a device where NDI discovery was working fine earlier. If the new source appears, that shows mDNS discovery operates correctly on the network. If not, check whether port 5353 (see the well-known NDI and TriCaster ports below) is blocked by any firewall rules, and that multicasting is not disabled in the network switch.

- Verify the send and receive groups using NDI Access Manager (or the **ndi-config.v1.json** on Linux systems) for both sides of the connection. If the NDI group names do not match between both sides, it will prevent sources from appearing.
- If your devices have Discovery Server as a discovery method enabled, verify that Discovery Server(s) is operating on your network and that no firewalls are blocking the network port.
- Devices with Discovery Server enabled will not generate mDNS messages for other NDI receiver systems. The receivers must also be configured to use Discovery Server.
- Applications using older NDI libraries (before version 4) will not detect Discovery Server sources. Look for software updates for these applications, which may update the NDI version it uses.
- Discovery Server's console output displays information when new clients are added into the database. If using this method of discovering and having issues, check if client information is being registered. If not appearing, it could be a firewall issue, an alternative port number is being used or the application is using older NDI libraries and doesn't support Discovery Server.
- If using mDNS discovery, add the IP of the device you are trying to connect with as an external source in NDI Access Manager. If this fixes discovery, then you know that something is blocking the mDNS communication on the network. Check firewall settings (or temporarily disable the firewall) to see if this resolves the issue or you can operate with this setting left intact.

Transmission Issues

Before looking at transmission issues, an aspect that is important to understand about NDI is how it handles missing packets. For NDI to display a frame of video, it must receive **all** the packets that create the frame. If any single packet does not arrive in time, then the last good frame is held until a new frame is displayed.

What this means is that NDI will not display corrupted, invalid data or a black frame, instead you will see a freeze frame.

These NDI transmission aspects apply to devices using High Bandwidth NDI, NDI|HXv2 and NDI|HXv3. Some NDI|HXv1 devices may use transmission modes that do not offer any error correction (these devices were built previously to NDI), and it is possible to get video artifacts to appear in the video frame when using these sources.

- If seeing studders or jumpy video, check the bandwidth of the Ethernet port on the system. In Windows, you can use Task Manager to see the current bandwidth for each network interface. Always allow some traffic overhead (at least 20%), with a

1Gb Ethernet connection if you are using 800 megabits or more, your Ethernet interface is likely saturated with too much traffic to work effectively. In these cases, either connect a second similar speed NIC, or upgrade to allow faster Ethernet speeds.

- If the network connection is not oversaturated with traffic, then try changing the receive modes using NDI Access Manager. Try each mode and use the one that works best.
- If the video appears in low resolution, verify that the application is not pulling the proxy or low bandwidth version of the NDI source. All NDI sources are available in a full resolution and proxy/low resolution version. Some applications offer a choice to switch between these resolutions.
- Verify that the network interface is running at the expected speed. If possible, force the network speed in the NIC and switch to only operate at the speed desired. This can be useful, as a poor Ethernet cable might jump between different speeds, causing what appears to be issues only part of the time. If you force the speed setting, then the port will go offline during these times, making it easier to diagnose the issue.
- If none of these issues appear to fix studder/jumpy video, then use NDI Studio Monitor and create a recording of the video. Then load this clip into an NLE application and you can step through the video frames to see if they exist or not. It is possible that the computer is not able to keep up with the NDI display, the frames are arriving, but the system cannot display them fast enough. If a recording shows the frames existing, then the data was received on the system and system performance needs to be increased.
- Video stuttering or jumping due to network congestion can impact all downstream connections, identify the root cause to resolve this issue. Starting troubleshooting with the source is often better than with the receiver.
- While troubleshooting a continuous ping command initiated between the sender and receiver in command prompt might help to see the health of that network path (ping x.x.x.x /t) – or a more detailed test via NDI Analysis will help to identify the problem.
- Use the iPerf performance testing tool setup between a sender device and receiver to test the throughput of the entire network path. On the server that will be receiving data open an elevated command window and run the following command.
    - Enter: `iperf.exe -s`

      (-s tells iPerf to run as a server, waiting for connections from other systems runing in client mode).
    - On the second system enter: `iperf -c x.x.x.x -w 2m -t 30s -i 1s`

      (-c means runing iPerf in client mode, x.x.x.x is the iperf server IP address, -w 2m sets the socket size to 2Mbyte, -t 30 sets the test lenght for 30 seconds, and -i 1s means iPerf will report back every second).
    - This command will run with TCP transmission and determines your throughput from server to client. You can run UDP transmission tests as well,

but that will need different parameters. Please refer to the iperf.fr website for documentation.

- If having issues with NDI|HXv1 source video, try using NDI Bridge setup in Local mode to regenerate the signal. This solution works best if the HXv1 devices are segmented to a separate network, allowing the only access to them through NDI Bridge which is spanning both networks. All signals processed through NDI Bridge will include error correction on the output, which some HXv1 devices lack.
- NDI Analysis is a free command tool available for Windows. It can help determine performance statistics of NDI sources over your network. It is useful to verify the quality of NDI connections along with specifications of the NDI source, like video resolution, framerate, number of audio channels, compression codec, NDI version of the sender and more. Adding the /framecheck argument in NDI Analysis v6 will include information like SDR/HDR color information.

  Download access and documentation on this tool can be found here: NDI Analysis

6.3 NDI Analysis used for troubleshooting

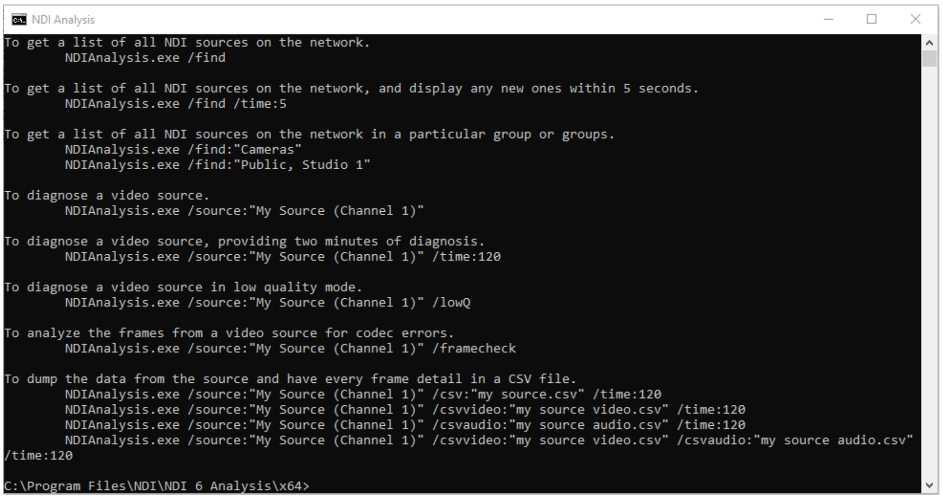
When you install NDI Analysis, you will find the full documentation of the NDI Analysis software in the installation folder. For deeper instructions on how the NDI analysis works and all its functions, please refer to that user manual.

The NDI Analysis tool is provided to help you diagnose and understand network issues. While it produces a deluge of detail about network connections and NDI streams, the human eye is by far the best arbiter of video performance and quality.

Ultimately your eye provides a superior real-time analysis, and one which is inherently more tolerant. In certain cases when subjective issues are visible, lower-level analysis of the root problem may be required to understand the underlying factors.

NDI Analysis will be most useful on the same network path as the struggling receiving device you are troubleshooting. If possible, replace the receiver with the device you are running NDI Analysis on (literally remove the receiver, and connect with the same cable to the NDI Analysis computer).

When you run NDI Analysis, it will open a command prompt window.



```
To get a list of all NDI sources on the network.
        NDIAnalysis.exe /find

To get a list of all NDI sources on the network, and display any new ones within 5 seconds.
        NDIAnalysis.exe /find /time:5

To get a list of all NDI sources on the network in a particular group or groups.
        NDIAnalysis.exe /find:"Cameras"
        NDIAnalysis.exe /find:"Public, Studio 1"

To diagnose a video source.
        NDIAnalysis.exe /source:"My Source (Channel 1)"

To diagnose a video source, providing two minutes of diagnosis.
        NDIAnalysis.exe /source:"My Source (Channel 1)" /time:120

To diagnose a video source in low quality mode.
        NDIAnalysis.exe /source:"My Source (Channel 1)" /lowQ

To analyze the frames from a video source for codec errors.
        NDIAnalysis.exe /source:"My Source (Channel 1)" /framecheck

To dump the data from the source and have every frame detail in a CSV file.
        NDIAnalysis.exe /source:"My Source (Channel 1)" /csv:"my source.csv" /time:120
        NDIAnalysis.exe /source:"My Source (Channel 1)" /csvvideo:"my source video.csv" /time:120
        NDIAnalysis.exe /source:"My Source (Channel 1)" /csvaudio:"my source audio.csv" /time:120
        NDIAnalysis.exe /source:"My Source (Channel 1)" /csvvideo:"my source video.csv" /csvaudio:"my source audio.csv"
/time:120

C:\Program Files\NDI\NDI 6 Analysis\x64>
```

The first command you should type in is `NDIAnalysis.exe /find` – this will find all available NDI sources in the Public NDI group.

It will return with a list of devices, and information like the following list

```
1: name="MINI-89ABD276 (MIX 1)"
  host="192.168.1.188"
  port="5963"
  type="NDI"
2: name="NDI-PTZ1 (Chan 1)"
  host="192.168.1.148"
  port="3705"
  type="NDI|HX (NewTek PTZ Camera)"
3: name="PTZ3-560436 (PTZ)"
  host="192.168.1.6"
  port="5961"
  type="NDI"
```

Every entry will have the following attributes – the device and NDI channel name (*name*), the device IP address (*host*), the port to receive the NDI stream (*port*), and the *type* of NDI stream available from the sender. Only the NDI-HXv1 sources will be marked as NDI|HX. All other types, NDI|HXv2, NDI|HXv3 & High Bandwidth NDI, sources will show up as type = “NDI”.

After successfully mapping out your NDI network, please use the following command to start your test on.

```
NDIAnalysis.exe /source:"NDI Device (NDI Channel name)"
```

You must enter the full NDI name between the quotation marks, just the device name or the channel name won’t identify the source you are looking for. In our example, the sender is going to be a TriCaster Mini 4K, and the tested output is the MIX1 so the command will look like this:

```
NDIAnalysis.exe /source:"MINI-89ABD276 (MIX 1)"
```

Please don’t use sources for testing like the NDI Tools’s Screen Capture application, or the Test Pattern Generator. These tend to send 1 frame only, or send new frames when there is a change in the video content, and the readings will be misleading.

You can specify any combination of these additional flags for the test:

`/time:60` – defines period of time the analysis runs in seconds.

`/videoonly` – only pulls the video source, the audio connection won’t be established

`/audioonly` – only pulls the audio source, the video connection won't be established

`/lowQ` – pulls the low bandwidth video feed for the test

`/find:groupname` – by default only the sources from the public NDI group is going to be “found”. If you’d like to list all other sources, specify the group name with the find flag.

`/framecheck` – you can test the validity of the encoded video frames. If you experience receiver crashes or hangs caused by the same sender repeatedly, it’s probably worth running the framecheck option.

Frame by frame this analysis will print something similar:

```
23:37:16.604: Analyzing 1920x1080 frame with timecode 133634909076994678
23:37:16.625: Analyzing 1920x1080 frame with timecode 133634909077194678
23:37:16.645: Analyzing 1920x1080 frame with timecode 133634909077394678
```

Results like these report that the frames from the sender are okay. Sometimes it could take hours to catch the problem, so make sure you are running an extensive test.

In case of a corrupted frame being received, you will see something like below:

```
DecodeLegacyField : Field: 0, Worker: 3, Bit Offset: 1100400, Previous: 826992, Bytes decoded:
34176, Worker Total: 34385
```

In case of an issue, check if there is software/firmware update available, update the device and try the test again. If that doesn’t resolve the issue, contact the manufacturer of the specific sender device with the framecheck log.

After starting the test with the /source flag, the NDI Analysis will open a connection from the receiver to the sender, and print out something like the following entries, with 5-second-long intervals as measurement time for Minimum, Maximum and Average rates.

```
17:47:50.821: Video receiver creation succeeded.
```

```
17:47:50.821: Audio receiver creation succeeded.
17:47:54.827: Audio connection opened.
17:47:54.827: Video connection opened.
17:47:54.837: Source is using NDI 6.0.1.0 on the WIN64 platform. (Note 1)
17:47:54.837: Capabilities, Recordable=false (Note 2)
17:47:54.837: Product, Long name=TriCaster Mini (Note 3)
17:47:54.837: Product, Short name=TCMINI4K (Note 3)
17:47:54.837: Product, Manufacturer=Vizrt AB. (Note 3)
17:47:54.837: Product, Version=8-3 (Note 3)
17:47:54.837: Product, Session name=IBC_webinar (Note 3)
17:47:54.853: Audio format changed. 48000Hz, codec=NewTek Mode 0, 2 channels.(Note
4)
17:47:54.858: Video format changed. 1920x1080, codec=SpeedHQ Mode 2, progressive,
aspect ratio=1.78, frame-rate=50.00, no alpha channel. (Note 5)
17:47:55.826: Video data rate (Mbps). Avg=100.78 (Note 6)
17:47:55.826: Video size (KB). Min=233.58 Avg=243.00 +/- 8.03 Max=256.45 (Note 7)
17:47:55.826: Video recv (ms). Min=18.77 Avg=19.99 +/- 0.39 Max=20.72 (Note 8)
17:47:55.827: Video send (ms). Min=19.27 Avg=20.01 +/- 0.36 Max=20.76 (Note 9)
17:47:55.827: Audio data rate (Mbps). Avg=1.54 (Note 10)
17:47:55.827: Audio size (KB). Min=2.82 Avg=2.82 +/- 0.00 Max=2.82 (Note 10)
17:47:55.827: Audio recv (ms). Min=13.65 Avg=15.00 +/- 0.49 Max=16.60 (Note 10)
17:47:55.827: Audio send (ms). Min=13.65 Avg=15.00 +/- 0.51 Max=16.61 (Note 10)
17:48:00.832: Video data rate (Mbps). Avg=98.35
17:48:00.832: Video size (KB). Min=232.77 Avg=239.82 +/- 4.87 Max=248.91
17:48:00.832: Video recv (ms). Min=18.45 Avg=20.00 +/- 0.42 Max=21.23
17:48:00.832: Video send (ms). Min=18.37 Avg=20.00 +/- 0.44 Max=21.28
```

Note 1: After establishing connection between the NDI Analysis and sender. Gets the source's platform, and the NDI SDK used by the sender. This can help determine why certain sources may not accept a requested NDI transmission mechanism (e.g. RUDP is not available, so the sender falls back to the commonly known version – Single-TCP).

Note 2: As NDI carries control capabilities available by the sender. This can list features like recordable, PTZ control, KVM control, Web Interface, etc.

Note 3: Some products will share the Vendor's name, product name, serial number, or in case of the TriCaster even the Session name what you actively use while you run your test.

Note 4: This is the first entry about audio receiving. The sender will send a change note every time the format changes. The sender provides the codec type, encoded channel count and sample rate.

Note 5: The first video frame has been received**.** As with audio, in case of a format change, this information gets printed in again. The sender provides details about the resolution, codec (SpeedHQ means I-frame only High Bandwidth NDI, while H.264 or HEVC will refer to a NDI-HX sender. HXv2 and HXv3 will differ in overall bandwidth), whether the source is progressive or interlaced, aspect ratio, framerate and if an alpha channel in the stream is present.

Note 6: In the 5 second interval, the average bandwidth of the received NDI video was 100.78 Mbit/s in our example.

Note 7: In the period of 5 seconds, the minimum size of a single video frame, and the maximum size for received video frame printed in, including with the average. In our example the `Min=233.58 Kbyte Avg=243.00 Kbyte +/- 8.03 Kbyte Max=256.45 Kbyte`. The "`+/-`" term is the standard deviation. Consider this as a measurement against how stable the frame size is, or other words how "relatively" constant the video content is. If the video content changes a lot, this number is going to be higher.

Note 8 Video recv (ms) – Probably the most important measurement. Represents the framerate of data received from the network. If we are measuring a 50P (50 Hz) video, like in our example, the average is going to point to the timeframe we must present the next video frame. This will be 20 ms.

Our example average number looks great – 19.99 ms, while the maximums do not exceed 21 ms. This is the highest latency measured for the receiver in that 5 second period to receive and decode video frames.

As we usually have other computers on the network, or even the same computer sending and receiving other traffic as well, the variance number is +/- 0.39 ms, way below the size of a frame. If that variance number goes higher, that might mean dropped frames.

All NDI applications should be designed to support frames within the Nyquist sampling limit, which would mean that any frame variance in the range of half the average should be the lowest reliable limit that would not require additional buffering or latency to display smoothly. At 50 Hz, this would require that the maximum reliable zero latency transfer rates would be `20.00 ms +/- 10.00 ms`.

The maximum times are also important; thus the largest delivery time (which is less then 21 ms in the example row, but during this short test it won't go higher than 22 ms) provides a very important diagnostic tool. If you see values that are over 90-100 ms or so, it is very likely that network frames were dropped, and the receiver had to re-request (RUDP), or rely on the ARQ mechanism in TCP. You should take steps to ensure that there is no packet loss on the network.

For advanced diagnosis of problems, the video receipt time can be combined with the video sending times.

Some key reference times related to video timing are listed below (in milliseconds):

Frame Rate	Maximum Latency
60Hz	16.66 ms
59.94 Hz	16.67 ms
50 Hz	20 ms
30 Hz	33.33 ms
29.97 Hz	33.37 ms
25 Hz	40 ms

Note 9 Video send (ms): This line provides diagnosis of the performance of the sender, before the packets hit the network. If the frame variance is high, the sender might be struggling to send video at the proper framerate to the network. If the high maximum numbers are occurring in these lines, you can expect sender side issues, even if the receiver line also shows high "max" numbers. In other words, if you don't see high variant and maximum numbers on the send side, but you do see it with the receive side, most likely the problem is occurring with network level, and not with the sender.

Note 10 Audio data rate, audio frame size, receive and send time – very similar to how the video side works. The audio data rate is self-explanatory. The audio frame size represents the average size in kilobytes of the audio frames received.

There is one difference however, that the number of bytes taken by each audio packet will depend on the number of audio samples in each audio packet. For instance, sending 800 sample packets will on average result in audio packets that are half the size of sending 1600 sample packets.

6.4 Network Port List

Name	Port Number	Protocol
TriCaster HTTP API / NDI Web UI Access / HTTP	80	TCP
RTMP / RTMPS / Flash Streaming	1935	TCP
mDNS / Bonjour	5353	UDP
TriCaster TCP Server API	5951	TCP
TriCaster Secondary HTTP API	5952	TCP
Viz Remote Control Surface Utility	5958	TCP
NDI Discovery Server	5959	TCP

NDI Messaging Server (list of signals available on device)	5960	TCP
NDI Reliable UDP Transmission (add one port for each connection)	5960+	UDP
NDI Single-TCP Transmission (add one port for each signal)	5961+	TCP
NDI Bridge	5990	UDP
Viz Engine Control	6100	TCP
NDI UDP Receiver (add one port for each signal)	6960+	UDP
NDI Multi-TCP Receiver (add one port for each signal)	6960+	TCP
NDI UDP Sender (add one port for each signal)	7960+	UDP
NDI Multi-TCP Sender (add one port for each signal)	7960+	TCP
VISCA PTZ Camera Control	52381	UDP

6.5 NDI Bandwidth

Bandwidth information for many common video formats using either High Bandwidth NDI (HB NDI), HX2 and HX3 formats, in full and proxy resolutions. All values are in megabits per second (all values have been rounded up).

Charts with additional video formats are available at the NDI.video website. [Bandwidth (ndi.video)]

Format	HB NDI	HB Proxy	HX2 H.264	HX2 HEVC	HX2 Proxy	HX3 H.264	HX3 HEVC	HX3 Proxy
2160 60p	250	66	30	21	6	110	84	3
2160 50p	224	66	27	19	6	92	70	3
1080 60p	133	66	16	11	6	62	50	3
1080 50p	126	66	15	10	6	52	41	3
1080 60i	113	19	11	8	4	31	25	3
1080 50i	103	19	10	7	4	26	20	3

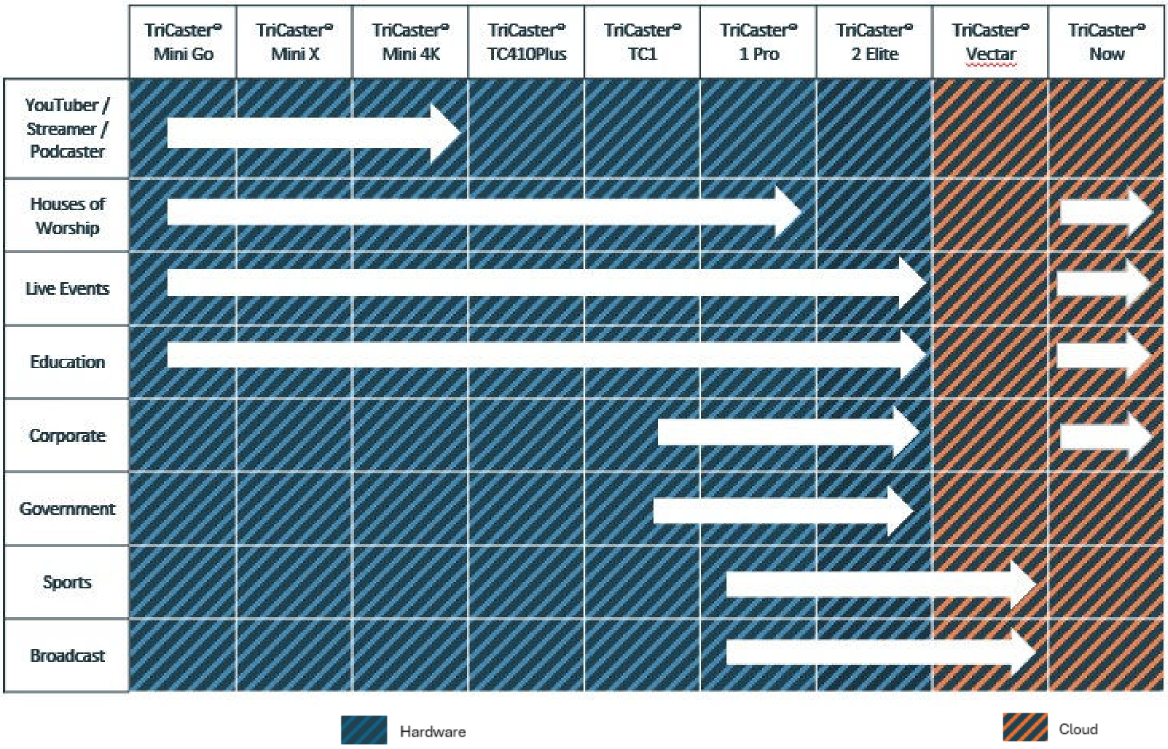
7. Additional Information

7.1 TriCaster

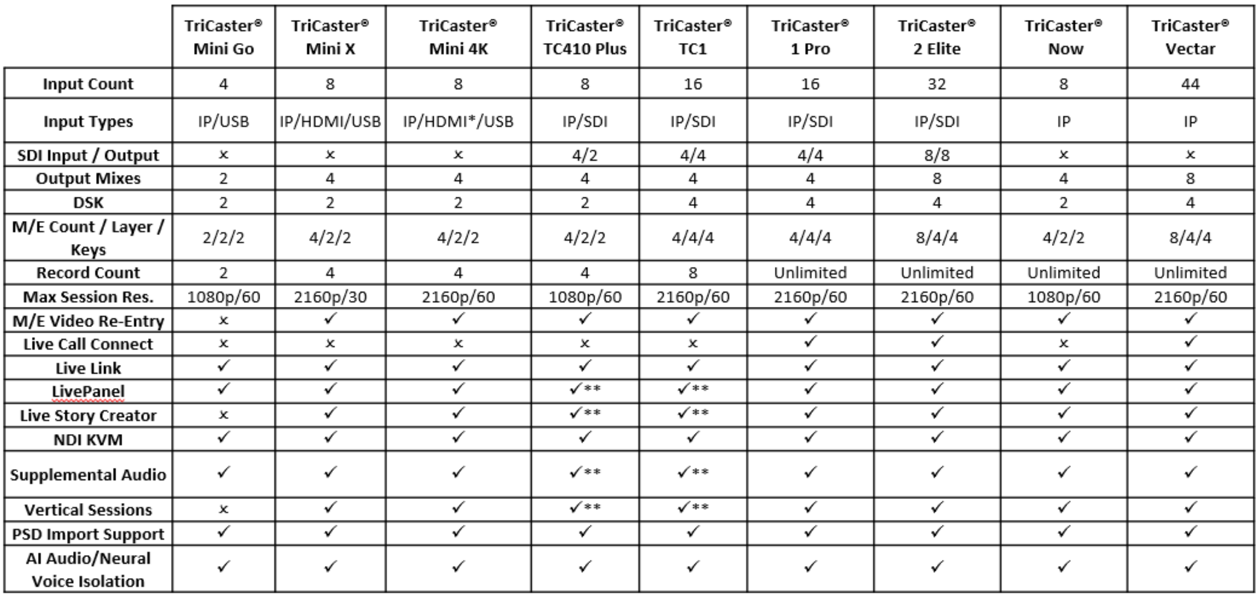
TriCaster is a sophisticated video mixing and live production solution available in sizes to suit any production level, and all built with NDI at their core. TriCasters cater to a range of uses from podcaster to broadcaster.

No matter the TriCaster, users can be certain of a scalable live production solution that caters to their evolving needs – from a desktop solution for small live productions with a handful of cameras to massive, high-end productions with countless signal flows.

Here is an outline of how broadly TriCasters can be used across diverse sectors:



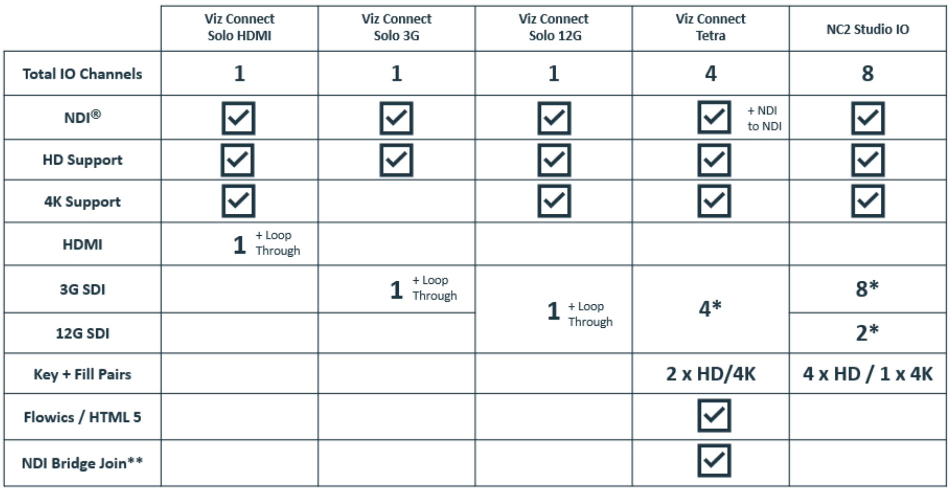
Below is an outline of how the technical specifications compare across our TriCaster Family:



	TriCaster® Mini Go	TriCaster® Mini X	TriCaster® Mini 4K	TriCaster® TC410 Plus	TriCaster® TC1	TriCaster® 1 Pro	TriCaster® 2 Elite	TriCaster® Now	TriCaster® Vectar
Input Count	4	8	8	8	16	16	32	8	44
Input Types	IP/USB	IP/HDMI/USB	IP/HDMI*/USB	IP/SDI	IP/SDI	IP/SDI	IP/SDI	IP	IP
SDI Input / Output	×	×	×	4/2	4/4	4/4	8/8	×	×
Output Mixes	2	4	4	4	4	4	8	4	8
DSK	2	2	2	2	4	4	4	2	4
M/E Count / Layer / Keys	2/2/2	4/2/2	4/2/2	4/2/2	4/4/4	4/4/4	8/4/4	4/2/2	8/4/4
Record Count	2	4	4	4	8	Unlimited	Unlimited	Unlimited	Unlimited
Max Session Res.	1080p/60	2160p/30	2160p/60	1080p/60	2160p/60	2160p/60	2160p/60	1080p/60	2160p/60
M/E Video Re-Entry	×	✓	✓	✓	✓	✓	✓	✓	✓
Live Call Connect	×	×	×	×	×	✓	✓	×	✓
Live Link	✓	✓	✓	✓	✓	✓	✓	✓	✓
LivePanel	✓	✓	✓	✓**	✓**	✓	✓	✓	✓
Live Story Creator	×	✓	✓	✓**	✓**	✓	✓	✓	✓
NDI KVM	✓	✓	✓	✓	✓	✓	✓	✓	✓
Supplemental Audio	✓	✓	✓	✓**	✓**	✓	✓	✓	✓
Vertical Sessions	×	✓	✓	✓**	✓**	✓	✓	✓	✓
PSD Import Support	✓	✓	✓	✓	✓	✓	✓	✓	✓
AI Audio/Neural Voice Isolation	✓	✓	✓	✓	✓	✓	✓	✓	✓

7.2 Viz Connect

With versatile functionality ranging from SDI to IP video conversion and I/O channel expansion, to 4K UHD 60p connectivity and IP interoperability, Viz Connect solutions are designed to help you build the workflow you need. Our current Viz Connect family is as follows:



	Viz Connect Solo HDMI	Viz Connect Solo 3G	Viz Connect Solo 12G	Viz Connect Tetra	NC2 Studio IO
Total IO Channels	1	1	1	4	8
NDI®	☑	☑	☑	☑ + NDI to NDI	☑
HD Support	☑	☑	☑	☑	☑
4K Support	☑		☑	☑	☑
HDMI	1 + Loop Through				
3G SDI		1 + Loop Through	1 + Loop Through	4*	8*
12G SDI					2*
Key + Fill Pairs				2 x HD/4K	4 x HD / 1 x 4K
Flowics / HTML 5				☑	
NDI Bridge Join**				☑	

*Mix and match up to the maximum number of supported IO Channels
**note requires download of latest NDI® Tools from NDI.video/tools, NDI|HX 4K support limited to 1x stream at launch

7.3 Vizrt PTZ Cameras

Offering exceptional HD or UHD picture quality, 20-30x optical zoom, great low-light performance, and phantom-powered audio, the PTZ3 PLUS and PTZ3 UHD PLUS cameras combine quality hardware with intelligent production-enhancing features, including AI presenter tracking and the world's first FreeD tracking data embedded via NDI|HX; all in a sleek, discrete body that blends into any space.

Here are the Vizrt PTZ Cameras in a nutshell:

	Vizrt PTZ3 Plus	Vizrt PTZ3 UHD Plus
Lens	20 x Optical & 2 x Digital Zoom	30 x Optical & 2 x Digital Zoom
Sensor	Panasonic ½.86" CMOS	Sony 1/1.8" 12MP Sensor
Video Output	NDI|HX 3, NDI|HX2, 3G-SDI, HDMI	
NDI Codecs	H.264 or HEVC	

For more detailed information and technical specifications visit our website.