µMPX Codec
Technical Guideline

Best practices when using the µMPX codec

Technical Guideline

February 2024

Your audio. Our solution.

2wcom

Manufacturer

2wcom Systems GmbH • Am Sophienhof 8 • 24941 Flensburg • Germany
Phone (+49) 461 662830-0 • Fax (+49) 461 662830-11
contact@2wcom.com • www.2wcom.com

Contents

1 Introduction

The scope of this document is to inform users about best practices using the µMPX codec with 2wcom devices, as well as possible limitations.

2 Prerequisites

The use of µMPX requires licensing per channel and type. Encoding requires an µMPX encoder license, decoding a µMPX decoder license.

3 Limitations

A hardware unit (i.e. MPX-1c, MPX-2c, MPX-2ds and MPX-1g) can only run two instances of µMPX in parallel due to CPU processing power limitations. In this regard µMPX version 3 instances take up more computing power than version 4 instances, further limiting the parallel use of µMPX. If a unit can handle two type version 3 instances, it could also handle one type of version 3 and one type of version 4 instance instead, if both are required to run in parallel. This yields the following maximum number of µMPX instances in the following devices:

- MPX-1c and MPX-1g
  - µMPX version 3: one encoder or one decoder
  - µMPX version 4: one encoder and one decoder
- MPX-2c:
  - µMPX version 3: two encoders or one decoder, one encoder and one decoder
  - µMPX version 4: two encoders or one decoder, one encoder and one decoder
- MPX-2ds:
  - µMPX version 3: one decoder
  - µMPX version 4: two decoders

4 Input considerations

The µMPX codec expects to make use of the full dynamic range of the input signal, meaning that it expects signals can be up to 0dBFS (without clipping, obviously). A complete MPX signal fed into a MPX series encoder should make use of the full range available, thus using up to 0dBFS for digital XLR inputs or up to 15dBu on analog inputs (BNC or XLR). Internally the units consider 15dBu to be equal to 0dBFS.

As this is vitally important for the best results, users should make sure that their input signals make use of the available range and do not apply any gains on the input signals either, no matter If they are positive or negative. **MPX is a complex signal, by applying gains to it, this impacts not just the audio in the signal but also the pilot and RDS levels. µMPX will recreate the same levels on the output it was presented with on the input of the encoder!** Reducing the input gain for MPX could also negatively the MPX decoding leading to less than optimal reconstruction or missing RDS on the output.

5 Compatibility considerations

Native communication for µMPX codecs is based on UDP packets only. **This is important because in order for 2wcom devices to interoperate with other vendors equipment, either the input source has to be configured to expect UDP on decoders or the encoder output has to be configured to stream as UDP only on encoders.** In between 2wcom devices, also UDP/RTP is supported. RTP enables encoder and decoders to exchange information about the stream via RTCP (real time control protocol) which is send using the port one greater than the streaming port, i.e. if the destination port for the stream is 5000, RTCP uses 5001. **This in important to know, as firewalls have to be configured to let both ports pass!**

Benefits of using UDP/RTP over just UDP:

- Decoders can send receivers reports via RTCP, which enables to monitor the streams performance on the encoder side
- RTP enables the use of ProMPEG-FEC and/or RIST to protect the stream further. This will work in addition to µMPX own FEC, meaning that both or all three can be used in parallel, if the user so chooses (more on that below)
- RTP also enables tracking of missing, duplicated and/or reordered packets, as well as jitter
- RTCP is also used to relay the contents of the stream to a decoder, so it can be setup in automatic mode

6 FEC considerations

There are three available error protection protocols and an additional further method to protect against packet loss available to protect a µMPX stream, two of which are only available in between 2wcom equipment.

6.1 µMPX FEC

The µMPX codec has a built-in forward error correction which is setup in the encoder. It allows user to insert n corrections packet every m standard packets. The µMPX FEC packets are send "inband" using the same UDP port as the main stream, thus increasing it ~94-95packets/second by (n+m)/m percent, i.e. n=1 and m=10 will result in 104-105 packets/second. FEC packets send, to our understanding, do not increase the codecs maximum bitrate, instead they use spare bandwidth. This type of FEC works best for links that only work in one direction, i.e. satellite transmission and can be considered "fire and forget", meaning, if there are too many packets lost, error will occur.

6.2 ProMPEG FEC

This by now "classic" error protection is only available in between 2wcom devices and only when using UDP/RTP. It creates an outer protection by organizing main stream packets into a 2-dimensional array and creating FEC packets by XOR packets over row and/or columns. The row and/or column streams are send using different ports numbers and as mentioned above for RTP streams, also employ RTCP on RTP port +1, something to consider when setting up firewalls.

Like the µMPX native FEC, this FEC is "fire and forget" as well, so it is best employed in unidirectional links. Compared to µMPX FEC, ProMPEG-FEC enables users to keep the main stream and repair streams separate using networking rules to also possible route them over different links. ProMPEG FEC can however create larger delays, increasing with larger encoding array dimensions. However, since ProMPEG FEC are normally distributed differently compared to µMPX FEC, it might be more resilient to burst error losses.

6.3 RIST

In more modern setups with solid bidirectional links, RIST might be the best option. RIST uses RTP and RTCP to create something similar to TCP, where receivers can inform senders about missing packets, to requests a resend of the missing packet, without the constant acknowledging that bogs down TCP connections. As long as there are no issues, a RIST protected stream does not increase the bandwidth requirements constantly like the other two FECs do. RIST only increases bandwidth requirements when it requests packet to be resend. This is a boon, but can also break a network, so care needs to be taken when setting up RIST. To avoid one or multiple receivers causing congestions by excessively requesting packets to be resend, the encoder should be setup to use RIST bandwidth limiting, which is the maximum bandwidth each decoder output is allowed to use for the main stream and possible resends, before it will no longer resend packets to avoid congestions.

As a rule of thumb: if you can expect ~5% packet loss on your link, it is good practice to allow at least twice that to be resend, i.e. if a µMPX stream with 320kbps is send, adding another 32-48kbps for RIST should suffice. Note: RIST bandwidth limiter looks at the total bandwidth of an encoder output, hence for the scenario above It needs to be set to 352-368kbps. Main stream packets will always be send, only repair (requested) packets are run against this number.

6.4 Dual streaming

Users can of course opt to send a redundant stream from encoder to decoder, which is something that µMPX supports natively for UDP streams. For RTP streams, 2wcom equipment can also use packets of one stream to fill gaps in the other stream, so this type of error protection via redundancy is available for both UDP and RTP.

7 Troubleshooting and support

Adherence to these guidelines should help to setup and operate solid MPX over IP connections using the µMPX codec. If you have further questions about using µMPX and 2wcom equipment, please do not hesitate to contact us directly, we will try to help you!