

Application Note DT-AN-IP-3

SMPTE 2022-7 Seamless Protection Switching Using the DTA-2162 Network Adapter

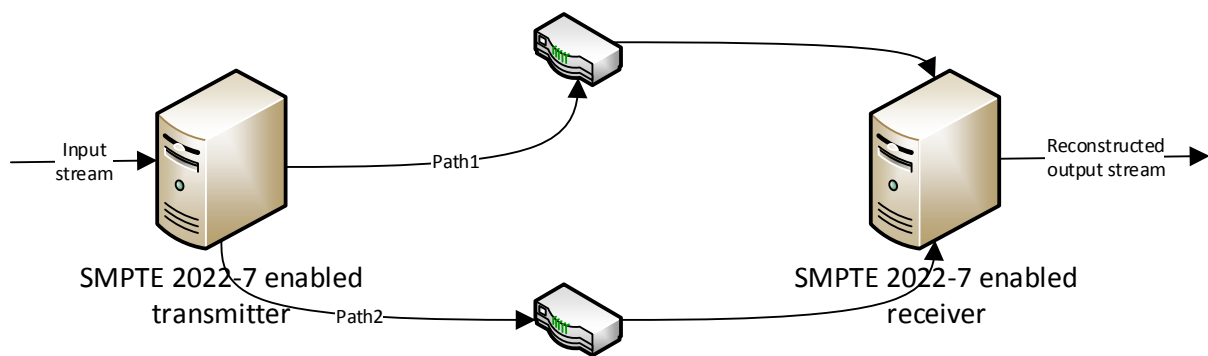
1. Introduction

When streaming data over an IP network, several methods exist to recover from lost IP packets and to reconstruct the original stream.

- One method is to append Forward Error Correction (FEC) data to the transmitted stream, and reconstruct lost packets at the receiver site. The SMPTE 2022-1 and SMPTE 2022-5 standards use this method.
- Another method is to generate two streams with the same data using different routes to the destination. SMPTE 2022-7 ("Seamless Protection Switching") specifies the reconstruction of the original stream in case packets are lost in any of the paths. Switching from one path to the other occurs without impact on the content of the stream.

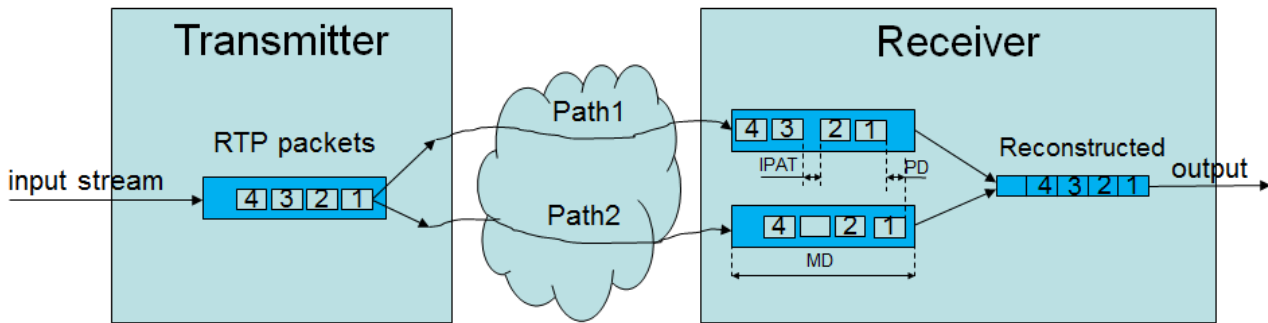
All DekTec network products support SMPTE 2022-1 and SMPTE 2022-5. The DTA-2162 network adapter also support SMPTE 2022-7.

2. Concept



The concept is straightforward. A SMPTE 2022-7-enabled transmitter duplicates the input stream and sends it via two different paths to the destination receiver. The receiver (also SMPTE 2022-7 enabled) combines the streams from both paths and reconstructs the original stream. If a packet was lost on path 1, the packet is taken from path 2. In case path 1 is completely gone, the entire stream is taken from path 2 and vice versa.

3. Details



The transmitter duplicates each IP packet and transmits it both to path 1 and to path 2. The Ethernet, IP and UDP headers of the IP packet may be different, but the UDP payload should be exactly the same. At the receiver site both streams are received and the stream is reconstructed.

As both streams use different routes from transmitter to receiver, the packets will generally not be received at the same time. Due to other network traffic extra network jitter may also be introduced. To be able to switch between path 1 packets and path 2 packets seamlessly (without interrupting the output stream), some buffering is needed at the receiver site to deal with this delay difference and/or jitter.

To measure delay and jitter, statistical information is collected at the receiver site.

- PD (Path Differential) is the delay between receiving the same packet from path 1 and from path 2.
- IPAT (Inter Packet Arrival Time) is the time between two consecutive packets of one path.

MD (Maximum Differential) must be configured at the receiver site using the IP transmission profile settings; this “m_MaxSkew” configuration is explained in the next section.

4. Using SMPTE 2022-7 with DTAPI

4.1. Setup channel

To use SMPTE 2022-7 Seamless Protection Switching mode with DTAPI is relatively straightforward. You only need to set the mode to SMPTE 2022-7 and configure the IP parameters for the redundant link in addition to the normal IP settings.

This can be done using the **SetIpPars** function with a **DtIpPars** object as argument.

The **DtIpPars** structure contains identical members for the redundant link as you have for the primary link. The members for the redundant link have the number 2 as suffix. Below is a list of these members.

Primary link	Redundant link	Description
m_Ip	m_Ip2	IP Address (IPv4/IPv6)
m_Port	m_Port2	UDP port number
m_SrcFltIp	m_SrcFltIp2	Source Filter IP Address (IPv4/IPv6)
m_SrcFltPort	m_SrcFltPort2	Source Filter UDP port number
m_VlanId	m_VlanId2	VLAN Identifier
m_VlanPriority	m_VlanPriority2	VLAN Priority

For details of these members, see the DTAPI documentation.

Additional to these redundant link parameters, you have to setup the IP transmission profile for the receive site. The IP transmission profile describes the maximum expected bitrate and the maximum skew between path 1 and path 2 packets. The skew value is the PD (Path Differential) as described in SMPTE 2022-7.

The IP transmission profile is contained in the **DtIpPars** structure in the *m_IpProfile* member. With the IP transmission profile you can select a predefined profile or choose a user-defined profile (**DTAPI_IP_USER_DEFINED**). When choosing the user-defined profile, you have to fill the *m_MaxBitrate* and *m_MaxSkew* members yourself. With the predefined profiles, these members are fixed.

The following pre-defined profiles are defined:

Profile Type	DTAPI Profile Constant	Member Values	
		m_MaxSkew	m_MaxBitrate
Low bitrate profile	DTAPI_IP_LBR_LOW_SKEW	10ms	10 Mbps
	DTAPI_IP_LBR_MODERATE_SKEW	50ms	10 Mbps
	DTAPI_IP_LBR_HIGH_SKEW	450ms	10 Mbps
Slow bitrate profile	DTAPI_IP_SBR_LOW_SKEW	10ms	270 Mbps
	DTAPI_IP_SBR_MODERATE_SKEW	50ms	270 Mbps
	DTAPI_IP_SBR_HIGH_SKEW	450ms	270 Mbps
High bitrate profile	DTAPI_IP_HBR_LOW_SKEW	10ms	3Gbps
	DTAPI_IP_HBR_MODERATE_SKEW	50ms	3Gbps
	DTAPI_IP_HBR_HIGH_SKEW	450ms	3Gbps
Default profile	DTAPI_IP_PROF_NOT_DEFINED	50ms	270Mbps

Please find below a code snippet that shows setting up a **DtIpPars** structure for transmission in SMPTE 2022-7 mode to destination IP address 192.168.1.100 port 12345 for the primary link, and to destination IP address 192.168.2.100 port 4321 for the secondary link.

```

DtIpPars IpPars;           // IP parameters used for Tx

// Set up destination IP address and port for the primary link
IpPars.m_Ip[0] = 192;
IpPars.m_Ip[1] = 168;
IpPars.m_Ip[2] = 1;
IpPars.m_Ip[3] = 100;
IpPars.m_Port = 12345;

// Set up destination IP address and port for the redundant link
IpPars.m_Ip2[0] = 192;
IpPars.m_Ip2[1] = 168;
IpPars.m_Ip2[2] = 2;
IpPars.m_Ip2[3] = 100;
IpPars.m_Port2 = 4321;

// Set the transmission mode to SMPTE 2022-7
IpPars.m_Mode = DTAPI_IP_TX_2022_7;
    
```

```
// Set up other parameters
IpPars.m_NumIpPerIp = 7;
IpPars.m_Protocol = DTAPI_PROTO_RTP;
IpPars.m_Flags = DTAPI_IP_V4;

// Set the parameters in the output channel.
// This assumes TsOutp is a DtOutputChannel object that has been
// attached to the hardware (e.g. DTA-2162)
TsOutp.SetIpPars (IpPars);
```

And a code snippet that shows setting up a **DtIpPars** structure for receiving using the SMPTE 2022-7 mode with IP address 192.168.1.100 port 12345 for the primary link and 192.168.2.100 port 4321 for the secondary link, with a maximal bitrate of 270Mbps and a maximum skew of 40ms.

```
DtIpPars IpPars;          // IP parameters used for Rx

// Set up destination IP address and port for the primary link
IpPars.m_Ip[0] = 192;
IpPars.m_Ip[1] = 168;
IpPars.m_Ip[2] = 1;
IpPars.m_Ip[3] = 100;
IpPars.m_Port = 12345;

// Set up destination IP address and port for the redundant link
IpPars.m_Ip2[0] = 192;
IpPars.m_Ip2[1] = 168;
IpPars.m_Ip2[2] = 2;
IpPars.m_Ip2[3] = 100;
IpPars.m_Port2 = 4321;

// Set the receive mode to SMPTE 2022-7
IpPars.m_Mode = DTAPI_IP_RX_2022_7;

// Set the max. bitrate and max. skew
IpPars.m_IpProfile.m_Profile = DTAPI_IP_USER_DEFINED;
IpPars.m_IpProfile.m_MaxBitrate = 270000000; // bps
IpPars.m_IpProfile.m_MaxSkew = 40;         // ms

// Set up other parameters
IpPars.m_NumIpPerIp = 7;
IpPars.m_Protocol = DTAPI_PROTO_AUTO;
IpPars.m_Flags = DTAPI_IP_V4;

// Set the parameters in the input channel.
// This assumes TsInp is a DtInpChannel object that has been
// attached to the hardware (e.g. DTA-2162)
TsInp.SetIpPars (IpPars);
```

4.2. Get statistical information

To get an indication of the quality of the network and to see the delay and jitter of the network, you can get statistical information from the DTAPI.

This statistical information can be retrieved using the GetIpStat function on an input channel attached to an IP port.

The GetIpStat function returns a DtlpStat structure with the statistics. Some statistics are only valid when using the SMPTE 2022-7 mode, others are also available in the normal mode.

The following statistics are available:

DtlpStat members	Description
m_TotNumIpPackets	Total number of IP packets that the stream should contain. Lost packets are included in this counter.
m_LostIpPacketsBeforeFec	Number of IP packets lost before FEC reconstruction. For SMPTE 2022-7 mode this is counter after the path1 and path2 merge.
m_LostIpPacketsAfterFec	Number of IP packets lost after FEC reconstruction.
m_NumIpPacketsReceived1/ m_NumIpPacketsReceived2	Number of IP packets received primary link/redundant link. Lost IP packets are not included in this counter.
m_NumIpPacketsLost1/ m_NumIpPacketsLost2	Number of IP packets lost primary link/redundant link.
m_QosStatsLastSec	This member contains the DtlpQosStats struct with extra statistics calculated over a time interval of 1 second. See the table below for the details of these statistics.
m_QosStatsLastMin	This member contains the DtlpQosStats struct with extra statistics calculated over a time interval of 1 minute. See the table below for details of these statistics.
DtlpQosStats members	Description
m_MinSkew/m_MaxSkew	The skew is the minimal (m_MinSkew) and maximal (m_MaxSkew) difference over time in arrival time between IP packets of the primary link compared to the IP packets of the redundant link. If the skew is positive, the primary link has a longer delay than the redundant link. If the skew is negative the redundant link has a longer delay. Note: PD as defined in SMPTE 2022-7 is the absolute value of the skew. The skew is measured in milliseconds.
m_Per1/m_Per2/ m_PerAfterFec	Packet Error Rate for the main link (m_Per1), redundant link (m_Per2) and the resulting stream (m_PerAfterFec). The m_PerAfterFec is the PER after FEC reconstruction (if also enabled). The
m_DelayFactor1/ m_DelayFactor2	The delay factor is an indication of the jitter of the IP stream. It is defined as the maximum difference between the actual arrival time of a UDP/RTP packet and the ideal (jitter less) arrival time of that packet. The delay factor is measured in microseconds.
m_Minlpat1/ m_Minlpat2	Minimal “Inter Packet Arrival Time” of two consecutive IP packets in milliseconds for the primary link and redundant link.
m_Maxlpat1/ m_Maxlpat2	Maximal “Inter Packet Arrival Time” of two consecutive IP packets in milliseconds for the primary link and redundant link.

The code snippet below can be used to get statistical information.

```

DtIpStat IpStat;           // IP statistics

// Get the statistics from the input channel.
// This assumes TsInp is a DtInpChannel object that has been
// attached to the hardware (e.g. DTA-2162)
TsInp.GetIpStat (&IpStat);
    
```

The DtJitter application can display these statistics in an easy-to-use interface. See the screenshot below for an example.

