XENA™
Test. Improve. Repeat.

APPLICATION NOTE

# Freely Programmable Test Packets

## The Custom Data Field and Extended Payload Features Explained

For many R&D and lab applications, flexible definition of Ethernet test packet is necessary to verify the behavior of a Device Under Test (DUT) in all kinds of traffic scenarios. This application note explains how you can use the Custom Data Field (CDF) and Extended Payload features in selected Xena high-speed test modules to compose a wide variety of test packets to test the DUT.

## Contents

Application Note ..... 3
Freely Programmable Test Packets ..... 4
Custom Data Field (CDF) ..... 4
Enabling the CDF Feature ..... 6
Defining Total Packet Size and Padding ..... 6
Configuring CDFs ..... 7
Disable the FCS ..... 7
Disable the TPLD ..... 7
Six Steps to Program Test Packets Freely ..... 8
Sequence of Test Packets ..... 8
Does My Test Module Support CDF? ..... 10
CDF Specifications ..... 10
CDF Memory Allocation ..... 11
Extended Payload ..... 11
Enabling the Extended Payload Feature ..... 11
Configuring the Extended Payload ..... 11
Disable FCS and TPLD ..... 13
Does My Test Module Support Extended Payload? ..... 13
Extended Payload Memory Allocation ..... 13
Modifiers ..... 13

# APPLICATION NOTE

When testing Ethernet devices and systems you typically send test traffic as data packets with an Ethernet header – and in many cases also other protocol headers – together with a payload to construct test packets with the required content and size. Ethernet testers will provide their users a rich selection of payload types, which in many cases are fully sufficient. However, for some R&D and lab applications it is necessary to generate freely programmable test packets – maybe even without the normal protocol headers – to verify the behavior of a Device Under Test (DUT) with all kinds of traffic types.

The Xena Networks Layer 2-3 testers – the ValkyrieBay and ValkyrieCompact equipped with selected high-speed test modules – allow advanced, flexible generation of test packets at 100 Gbps, 50 Gbps, 40 Gbps, 25 Gbps and 10 Gbps data rates. This application note explains how the Custom Data Field (CDF) and Extended Payload features works, and how you can use these features to freely program test packets to verify the behavior of a DUT.

Figure 1: The versatile and powerful Xena Networks Layer 2-3 testers ValkyrieBay and ValkyrieCompact

## FREELY PROGRAMMABLE TEST PACKETS

When testing communication devices like NPU (Network Processor Units) during product development, flexible generation of test packets – where the user programs every byte in the packet – can be of great importance.

NPUs are used in communication network equipment e.g. switches, routers, firewalls, intrusion detection devices and intrusion prevention devices. The NPU will have various tasks in the network equipment like:

- Identification of bit patterns in the data packets
- Table lookup of routing information
- Change fields in the data packets when required
- Management of queues and buffers

During development it must be verified that the NPU has a predictable behavior not only when expected data packets are processed, but also when odd or abnormal packets arrive at the NPU. Development may also involve new message formats that require a specific action from the NPU. Freely programmable test packets will assist the developers to generate test traffic that can uncover how the device is impacted by abnormal packets, new message formats and other traffic conditions.

Development of other devices and communication equipment can also require freely programmable test packets. Furthermore, generating normally formatted Ethernet packets and relevant protocol headers on top (IP, TCP/UDP etc.) with very long, user defined payload can help developers when implementing new features in devices and communication equipment.

### CUSTOM DATA FIELD (CDF)

The Custom Data Field (CDF) feature is available on selected Xena Networks high-speed test modules. It allows definition of a sequence of custom data fields for streams of test traffic. Each CDF is sent as a separate test packet in the order they are defined in the stream. The CDFs can have both different sizes and different content. The CDFs are inserted into the test packets at an offset specified for the stream; all CDF definitions for a given stream uses the same offset value.

When a set of protocol header segments has been specified for a stream and the CDF offset is placed within the area occupied by the headers, the CDF data will overwrite the protocol header data. If the offset is 0, the CDFs will overwrite all protocol headers defined for the test packet. If you set the offset to 0 and also disable the Frame Check Sequence (FCS) and the Xena proprietary *Test PayLoad Data* (TPLD) field, you can freely program all the bytes in the test packet. How to remove the FCS and the TPLD is explained in subsequent sections.

The CDF memory for each stream is split into CDF blocks, each containing a single CDF. The block sizes are identical for all CDFs in the stream and dimensioned to fit the largest CDF. The block size is $2^n$ bytes, ranging from 64 bytes to the Maximum Transmission Unit (MTU) for the port. If the CDF is shorter than the block size, the remaining part of the block is filled with padding, which is defined as the “Payload Type” in the “Packet Content” section of the Stream Properties panel.

Figure 2 illustrates test packets sent in a stream, which defines two CDFs overwriting a part of the MAC header. Each CDF is sent as a separate test packet. Offset and Protocol headers are common for the test packets defined in the stream. If the packet headers need to be visible in each test packet, the offset must be set to the length of the packet header section as a minimum. In case different definitions for offset and Protocol headers are needed, the CDFs must be defined in separate streams.

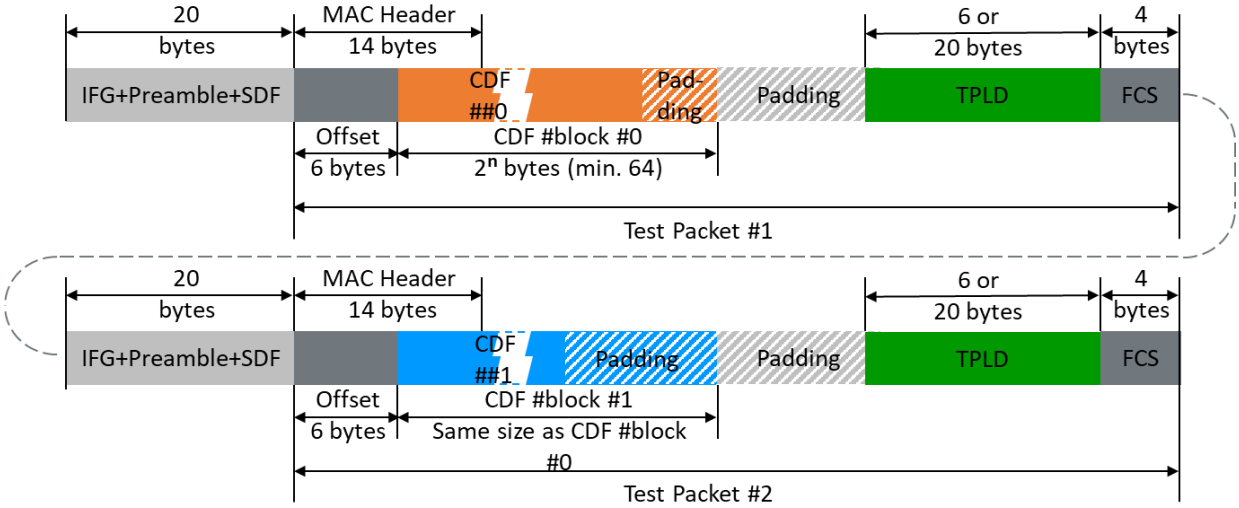



Figure 2: Stream definition with two CDFs overwriting a part of the MAC header

The total length of the test packets is defined in the "Packet Content" section of the Stream Properties panel (see figure 5).

- If the total length of the test packet is longer than the size of the Offset + CDF block + PTLD + FCS, padding is added between the CDF block and the TPLD – figure 2 shows two test packets where padding is added both after the actual CDF and after the CDF block.
- If the total length of the test packet is shorter than the size of the Offset + CDF block + PTLD + FCS, the CDF block is cut off to fit into the test packet from the end of the CDF block (i.e. where padding may be located).
- If the CDF data offset is set to a position after the end of the protocol header segments, padding is added between the protocol headers and the CDF (see figure 3).

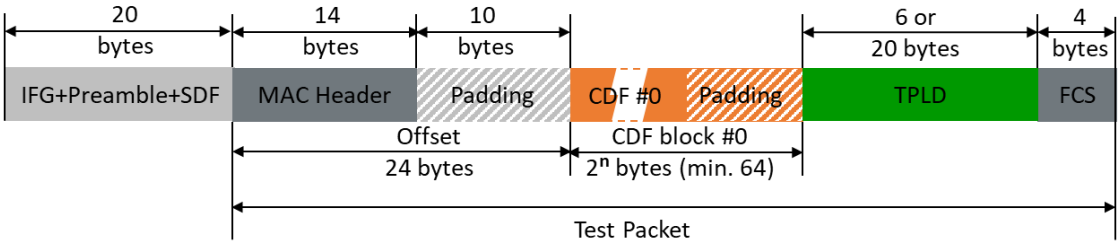



Figure 3: Test packet with one CDF, starting 10 bytes after the protocol header (in this case a MAC header)

## Enabling the CDF Feature

The CDF feature is enabled by selecting “Custom Data Field” in the “Payload mode” field of the parent Port Properties panel (see figure 4). This enables the feature for all streams on this port.

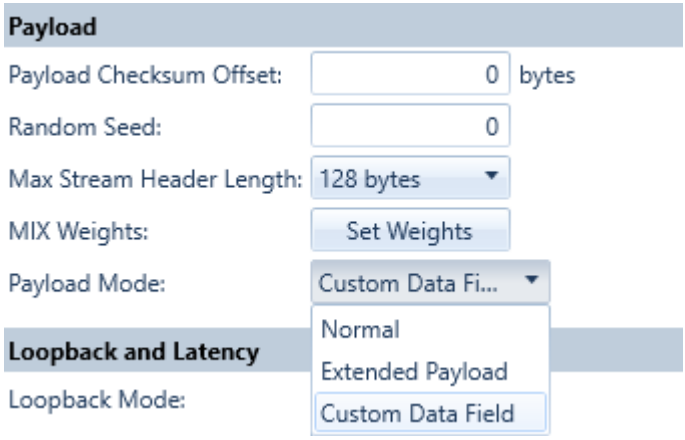



Figure 4: Payload Mode field

## Defining Total Packet Size and Padding

In the Packet Content section of the Stream Properties panel, several parameters relevant for the CDFs can be defined:

- The Packet Size Type
- Information specifying the total length of the test packets in the stream
- The Payload Type, which defines the padding that may be used before, after and within the CDF block in the test packet

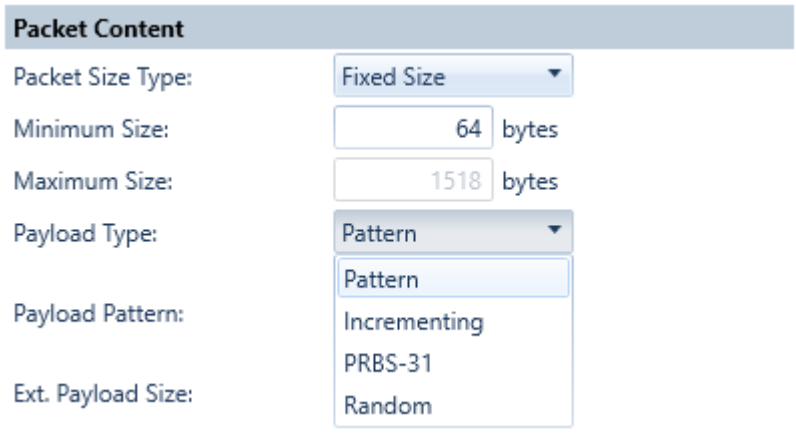



Figure 5: The selected Payload Type will be used as padding

## Configuring CDFs

When the CDF function has been enabled on the parent port of a stream, a new Custom Data Fields section is enabled in the Stream Property page, as shown in figure 6.

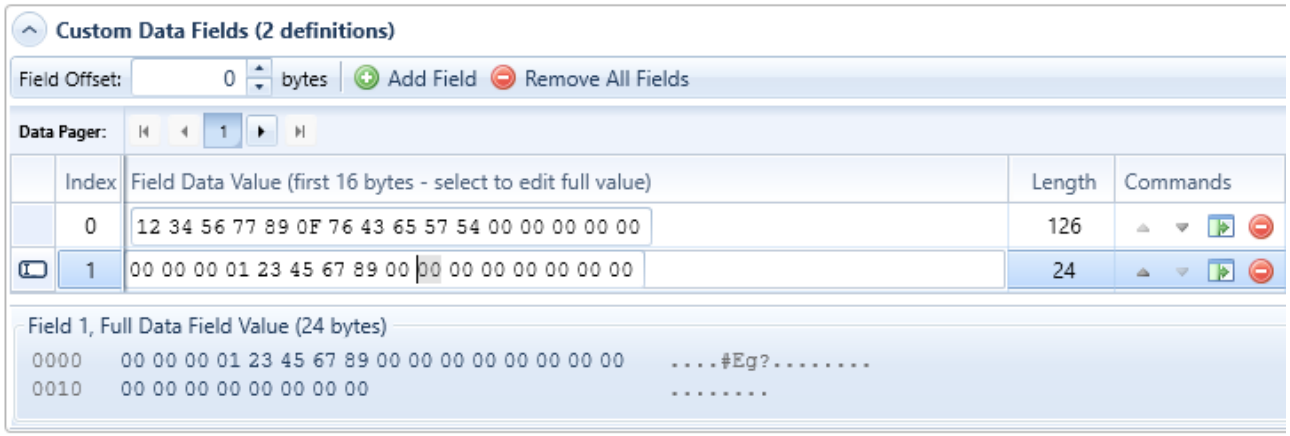



Figure 6: CDF configuration

Click Add Field to add another CDF. When you add a new CDF, you define its length. The length can be changed at a later time by clicking .

The offset (i.e. the starting point in the packet) for the CDFs in the stream is set in the “Field Offset” field.

Enter the desired contents of the CDF into the Field Data Value.

## Disable the FCS

The Frame Check Sequence (FCS) can be omitted from the test packet by unchecking the “Insert Frame Checksum (FCS) check box in the “Error Handling” section of the Stream Properties panel for the stream.

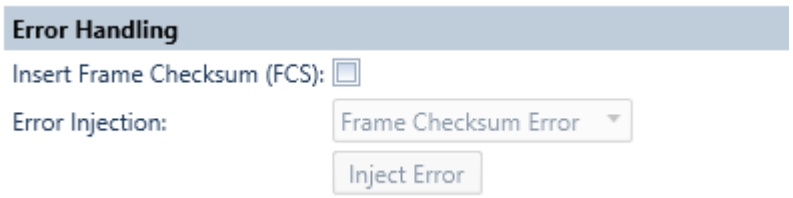



Figure 7: Error Handling section of the Stream Properties panel

## Disable the TPLD

At the end of standard Xena test packet a special proprietary data area called the *Test PayLoad Data* (TPLD) is added by default (see figures 2, 3 and 12). The TPLD contains various information about the packet and permits the Xena tester to detect anomalies like lost packets or mis-ordered packets and to measurelatency and packet jitter on Ethernet links. When the CDF feature is enabled, you may want to remove the TPLD from the test packets in the transmitted stream to have full control over the contents of the packets – please observe however that when the TPLD is omitted the above-mentioned statistics will not be available.

When you create a stream, the stream is assigned a stream number and a Test Payload ID (TID), which identifies the TPLD. If you remove the Test Payload ID as indicated in figure 8, the TPLD is omitted.

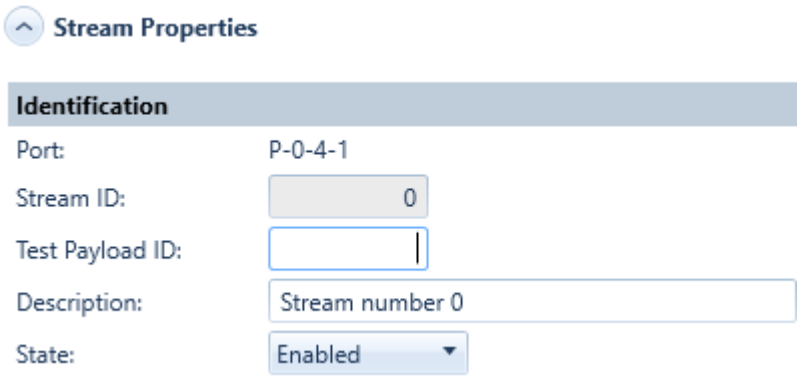



Figure 8: Identification section of the Stream Properties panel

## Six Steps to Program Test Packets Freely

In summary: To enable freely programmable test packets, where the user can define all bytes of the packet on Xena high-speed test modules that support CDFs, the following steps must be performed:

1. Activate the CDF feature on the test port (see figure 4)
2. Set packet size to exactly match the length of the wanted test packet (see figure 5)
3. Set offset to 0 (see figure 6)
4. Enter the required content into the CDF (see figure 6)
5. Disable FCS (see figure 7)
6. Disable TPLD (see figure 8)

If CDFs with different sizes are needed and you don't want padding to be added, a new stream must be defined for each CDF size.

## Sequence of Test Packets

By default, the streams are treated independently, and are merged into a combined traffic pattern for the port, considering the Transmission Profile defined for each stream (see figure 10, Normal Port TX Mode).

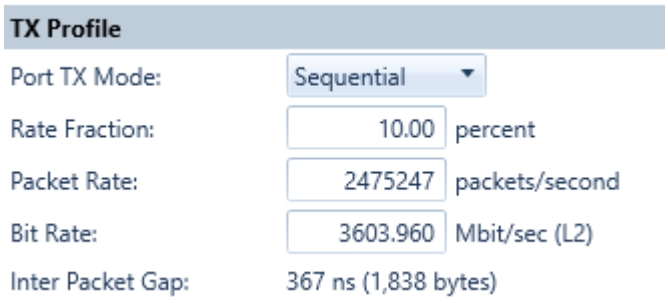
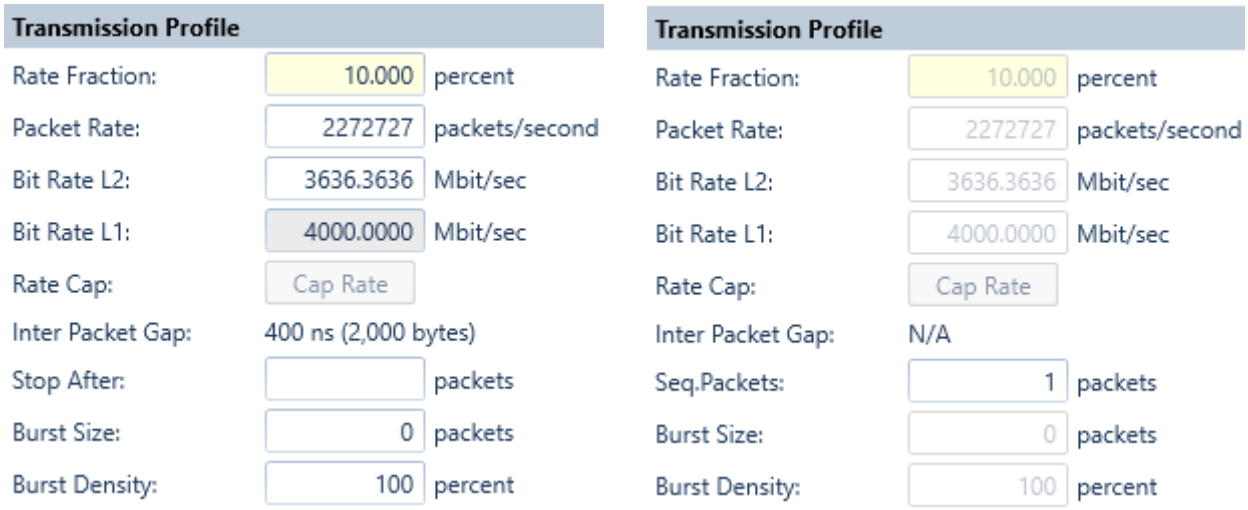



Figure 9: Identification section of the Stream Properties panel

As an alternative the test packets can be sent sequentially if Port TX Mode in the parent Port Properties panel is set to Sequential (figure 9): Each stream in turn contribute one or more packets, before continuing to the next stream, in a cyclical pattern. The number of packets that a stream contributes is entered in the Seq.Packets field in the Transmission Profile for the stream (see figure 10, Sequential TX Mode). The individual stream rates are ignored; the overall rate is determined at the port-level in the TX Profile section (figure 9). This in turn determines the rates for each stream, considering their packet lengths and counts. The maximum number of packets in a cycle (i.e. the sum of Seq.Packets for all enabled streams) is 500.




Normal Port TX Mode (default) Sequential TX Mode

Figure 10: Stream Transmission Profile definitions

- If the number in the Seq.Packets field matches the numbers of CDFs in a stream, all the CDFs from the stream are sent out before sending CDFs from the next stream.
- If the Seq.Packets number is higher than the number of CDFs in a stream, the CDFs in that stream are repeated until the Seq.Packets number is reached; then CDFs from the next stream are sent.
- If the Seq.Packets number is lower than the number of CDFs in a stream, that number of CDFs are sent in the order they are defined within the stream when CDFs from the stream can be sent.

Figure 11 illustrates how a traffic pattern is affected by different settings of Seq.Packets. In all examples two CDFs are defined for stream 1, while one CDF is defined for streams 0, 2 and 3.

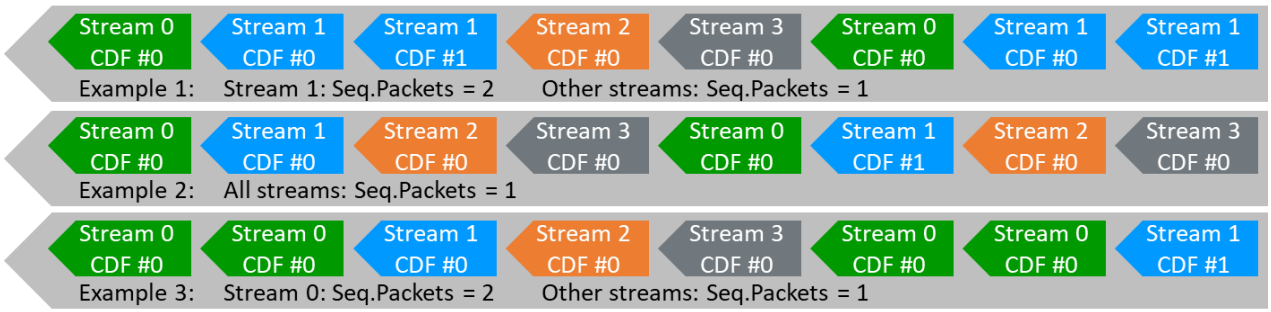



Figure 11: Sequential test packet flow examples with different Seq.Packets settings

## Does My Test Module Support CDF?

In the Port Capabilities part of the Port Properties panel you can see if the test module supports CDFs: If "Custom Data Fields Supported" is "True", the test module supports CDFs (see figure 12).

| | |
|---|---|
| Custom Data Fields Supported | True |
| Ext. Payload Supported | True |

Figure 12: CDF and Extended Payload support status

## CDF Specifications

The total area available for the CDF function depends on the test module type and port configuration.

Basic CDF specifications per 100G port:

- Max number of CDFs: 4096
- Max CDF size: MTU for the port.
- Max CDF offset = 2032
- Max CDF Memory: 262144 bytes (256 Kbytes)

The MTU for the port is found in the Port Capabilities part of the port properties panel as the "Max Packet Length" (please observe that different data rates of a port may have different MTU):

| | |
|---|---|
| Max Packet Length (bytes) | 12,288 |

Figure 13: Max Packet Length (bytes) is the MTU for the port (in this case 12288 bytes)

A future version of the M1QSFP28SFP28 test module will support the following at 100G:

- Max number of CDFs: 14336
- Max CDF size: MTU for the port.
- Max CDF offset = 2032
- Max CDF Memory: 917504 bytes (892 Kbytes)

When operating the module at lower port speeds, the ports share the above CDF memory and max. number of CDFs per port as follows:

CDF Memory per port:

- 40G/50G port: Max 100G CDF Memory / 2
- 25G port: Max 100G CDF Memory / 4
- 10G port: Max 100G CDF Memory / 8

Max. number of CDFs per port:

- 40G/50G port: Max 100G number of CDFs / 2
- 25G port: Max 100G number of CDFs / 4
- 10G port: Max 100G number of CDFs / 8

### CDF Memory Allocation

The amount of CDF memory used by a single stream is calculated in three steps:

1. **Calculate the CDF size**
   The size of a CDF is a combination of the number of data bytes it contains and its offset into the packet. It is calculated by the following formula:
   CDF size = datasize (in bytes) + (offsetbytes_value *modulo 8*).
   I.e. a CDF with an offset of 3 and a data size of 234 bytes will have a CDF size of 237 bytes; if the offset is 11, the CDF size is also 237 bytes (11 *modulo 8* is 3).
2. **Calculate the CDF block memory size**
   The CDF memory for each stream is split into blocks, each containing a single CDF. The CDF block sizes are identical for all CDFs on the stream and $2^n$ bytes long, dimensioned to fit the largest CDF (see step 1). The block sizes can range from 64 bytes to the MTU for the port.
   As an example: If the *largest* CDF entry defined for a stream is 237 bytes, the block size will be 256 bytes. Hence, *all* CDFs for that stream will use 256 bytes of memory, regardless of their individual sizes.
3. **Calculate the CDF memory usage for a single stream**
   Based on the two steps above, the formula for calculating how much CDF memory a single stream uses is:
   Stream CDF memory usage = CDF_memory_block_size * number_of_CDFs
   Hence, if the stream described in the example above has 500 CDFs, the total used CDF memory for the stream is: 256 bytes * 500 = 128000 bytes.

## EXTENDED PAYLOAD

The Extended Payload feature is another feature available on selected Xena Networks high-speed test modules. The Extended Payload allows very long user defined payloads. The Extended Payload will occupy the entire payload field immediately after the protocol headers section in the test packet.

### Enabling the Extended Payload Feature

The Extended Payload feature is enabled by selecting “Extended Payload” in the “Payload mode” field of the parent port panel (see figure 4). This enables the feature for all streams on this port.

### Configuring the Extended Payload

Once the extended payload is enabled, it is possible to set the desired size of the Packet Content area for the stream shown in figure 14 in the “Ext. Payload Size” field. Please observe however that the total test packet size (which is also defined in figure 14) must be minimum the size of the Protocol Headers defined for the stream + Extended Payload + TPLD + FCS.

- If the total packet size is shorter than that, the Extended Payload is cut to fit the test packet
- If it is longer, padding will be inserted between the Extended Payload and the TPLD

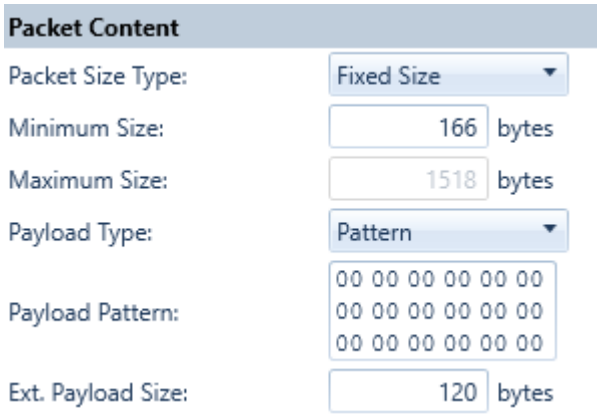



Figure 14: Set the Extended Payload size

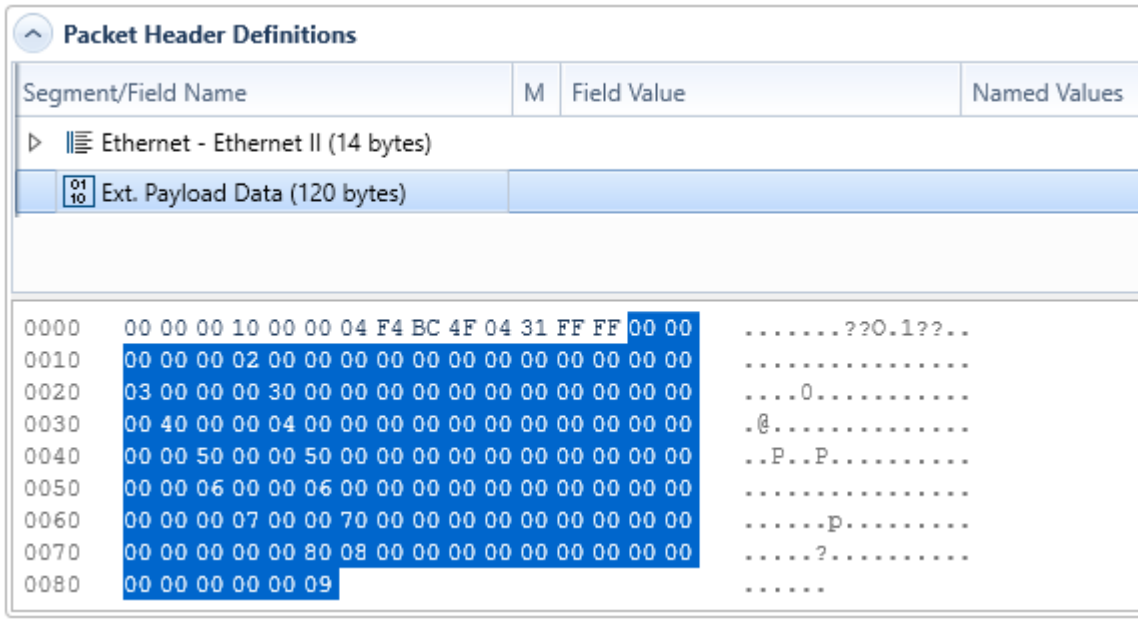



Figure 15: Stream protocol header editor

When the size is set, the equivalent data area will be available in the stream protocol header editor, as illustrated in figure 15. The user can fill the data area with the required content.

Like a CDF, the Extended Payload is fit into a block, which is $2^n$ bytes long. The block sizes can range from 64 bytes to MTU for the port; the smallest size where the Extended Payload fits in is used. Bytes in the block that is not used by the Extended Payload is filled with padding like CDF blocks.

Figure 16 illustrates a test packet with 120 bytes Extended Payload in a 166 byte test packet. For the test packet in figure 16 this means that there is room for the Extended Payload plus the remaining 8 bytes in the memory block that the Extended Payload occupies.

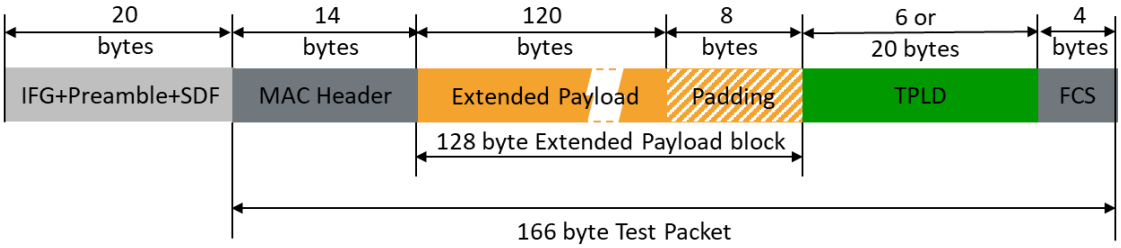



Figure 16: Test packet with 120 bytes Extended Payload

### Disable FCS and TPLD

As for CDFs, the FCS and the TPLD can be omitted from the test packets with Extended Payload as shown in figures 7 and 8.

### Does My Test Module Support Extended Payload?

In the Port Capabilities part of the port properties panel you can see if the test module supports Extended Payload. If “Ext. Payload Supported” is “True”, the test module supports Extended Payload (see figure 9).

### Extended Payload Memory Allocation

The Extended Payload feature is a special application of the CDF feature with an offset matching the headers in the test package. Therefore, the sections on CDF specifications and CDF Memory Allocation also apply for the Extended Payload feature.

## MODIFIERS

It is possible to set a modifier in the Extended Payload area just as it is for a normal protocol field.

Modifiers cannot be set on CDFs as the feature itself can be viewed as having some of the same characteristics as the modifier option but on a much larger scale.