Keywords: Interoperability, TDMoP, TDMoIP, SAToP, CESoP, MPLS, MEF, UDP, RTP, DS34T101, DS34T102, DS34T104, DS34T108, DS34S101, DS34S102, DS34S104, DS34S108

APPLICATION NOTE 4158

Interoperability of Maxim's TDM-over-Packet (TDMoP) Devices with Other Vendors' TDMoP Devices

Dec 20, 2007

Abstract: This application note provides the requirements for using Maxim's TDM-over-packet (TDMoP) devices with other vendors' TDMoP devices. The Maxim TDMoP devices covered by this application note are the DS34T101, DS34T102, DS34T104, DS34T108, DS34S101, DS34S102, DS34S104, and DS34S108.

Requirements for Interoperability

Interoperability is the ability of a system to work with other vendors' systems with little or no intervention from the system operator. The interoperability of systems makes it possible to provide services to, and accept services from, other systems and enables different vendors' systems to operate properly together.

This application note describes how to implement interoperability between Maxim's TDM-over-packet (TDMoP) devices and other vendors' TDMoP devices. The Maxim TDMoP devices covered by this application note are the DS34T101, DS34T102, DS34T104, DS34T108, DS34S101, DS34S102, DS34S104, and DS34S108.

The packet stream created by Maxim's TDMoP devices may not have the same packet header information as that created by other vendors' TDMoP devices. In order to make Maxim devices interoperable, the user needs to know the setup type. The setup may be one of the following:

- IP/UDP/RTP/SAToP
- IP/UDP/RTP/CESoPSN
- MEF/CESoETH-unstructured (i.e., MEF/SAToP)
- MEF/CESoETH-structured locked (i.e., MEF/CESoPSN)

Each setup has different packet headers. In order to be interoperable, the packet header from Maxim's TDMoP devices must be formatted the same as the packet header from other vendors' TDMoP devices. The user needs to compare the packet headers from Maxim's TDMoP devices with the packet headers from the other TDMoP devices. If there are format differences, this application note will show how to modify packet header values for Maxim's TDMoP devices by using Maxim's user application.

TDM-over-Packet (TDMoP)

This section defines the functional description for TDM-over-packet modules.

TDMoP Packet Format

To transport TDM data through packet-switched networks, the TDMoP device encapsulates TDM data into

Ethernet packets, as depicted in **Figure 1**.

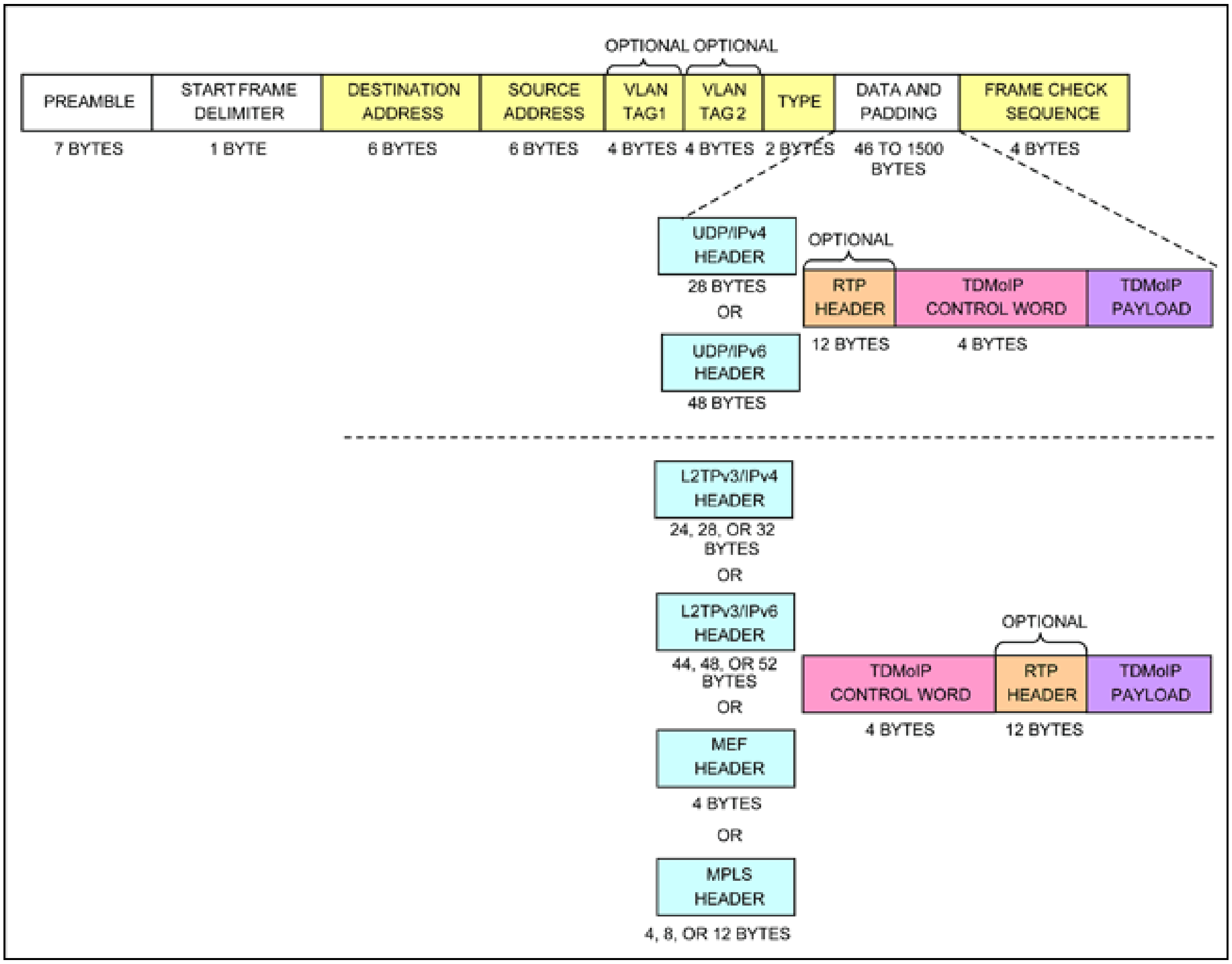


Figure 1. TDM-over-packet encapsulation in an Ethernet packet.

Table 1. Ethernet Packet Structure

Field	Description
Preamble	A sequence of 56 bits (alternating 1 and 0 values) used for synchronization; gives components in the network time to detect the presence of a signal.
Start Frame Delimiter	A sequence of 8 bits (10101011) that indicates the start of the packet.
Destination and Source Addresses	The Destination Address field identifies the station or stations that are to receive the packet. The Source Address identifies the station that originated the packet. A Destination Address may specify either an individual address destined for a single station, or a multicast address destined for a group of stations. A Destination Address of all 1 bits refers to all stations on the LAN and is called a broadcast address.
Type	Ether type
Data and Padding	This field contains the data transferred from the source station to the destination station or stations. The maximum size of this field is 1500 bytes. If the size of this field is less than 46 bytes, then padding is used to bring the packet size up to the minimum length. A minimum Ethernet packet size is 64 bytes from the Destination Address field through the Frame Check

	Sequence.
Frame Check Sequence	This field contains a 4-byte cyclical redundancy check (CRC) value used for error checking. When a source station assembles a packet, it performs a CRC calculation on all the bits in the packet from the Destination Address through the Pad fields (that is, all fields except the Preamble, Start Frame Delimiter, and Frame Check Sequence). The source station stores the value in this field and transmits it as part of the packet. When the destination station receives the packet, it performs an identical check. If the calculated value does not match the value in this field, the destination station assumes an error has occurred during transmission and discards the packet.

VLAN Tag

As specified in IEEE® Standard 802.1q, the 12-bit VLAN identifier's tagged packets enable the construction of a maximum of 4,096 distinct VLANs. For cases in which this VLAN limit is inadequate, VLAN stacking provides a two-level VLAN tag structure, which extends the VLAN ID space to over 16 million VLANs. Each packet can be sent without VLAN tags, with a single VLAN tag, or with two VLAN tags (VLAN stacking). **Figures 2** and **3** show a single VLAN tag and stacked VLAN tags, respectively.

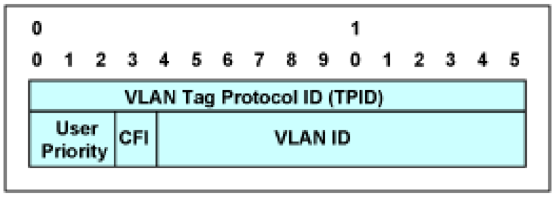


Figure 2. Single VLAN tag.

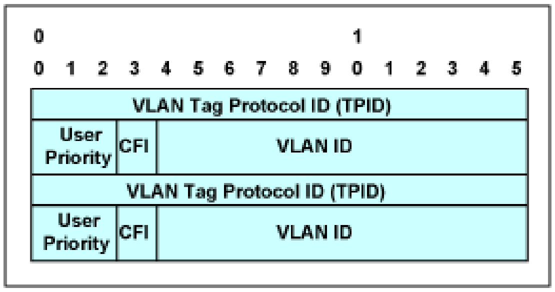


Figure 3. Stacked VLAN tags.

The VLAN tag's Protocol ID (TPID) used to identify the VLAN tag can be either 0x8100 or a value configured in the vlan_2nd_tag_identifier configuration register.

- The User Priority field is used to assign a priority level to the Ethernet packet
- The CFI (Canonical Format Indicator) field indicates the presence of a Router Information field
- The VLAN ID field uniquely identifies the VLAN to which the Ethernet packet belongs

The figures below show the header of different protocols.

- **Figure 4** shows the **UDP/IPv4** header structure
- **Figure 5** shows the **UDP/IPv6** header structure
- **Figure 6** shows the **MPLS** header structure
- **Figure 7** shows the **MEF** header structure
- **Figure 8** shows the **L2TPv3/IPv4** header structure
- **Figure 9** shows the **L2TPv3/IPv6** header structure
- **Figure 10** shows the **Control Word** header structure
- **Figure 11** shows the **RTP** header structure

The tables below describe the different fields of the header structure.

- **Table 2** describes the different fields of the **IPv4** header structure
- **Table 3** describes the different fields of the **UDP** header structure
- **Table 4** describes the different fields of the **IPv6** header structure
- **Table 5** describes the different fields of the **MPLS** header structure
- **Table 6** describes the different fields of the **MEF** header structure
- **Table 7** describes the different fields of the **L2TPv3/IPv4** header structure
- **Table 8** describes the different fields of the **L2TPv3** header structure
- **Table 9** describes the different fields of the **L2TPv3/IPv6** header structure
- **Table 10** describes the different fields of the **Control Word** header structure
- **Table 11** describes the different fields of the **RTP** header structure

UDP/IPv4 Header

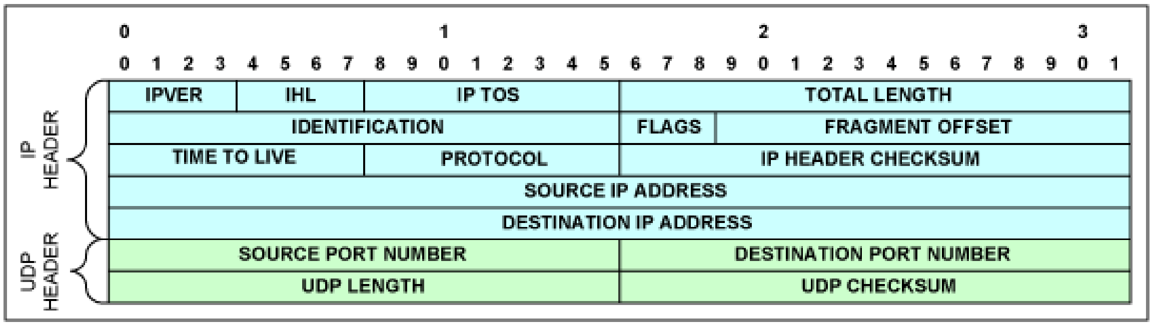


Figure 4. UDP/IPv4 header.

Table 2. IPv4 Header Structure

Field	Description
IPVER	IP version number; for IPv4 IPVER = 4
IHL	Length in 32-bit words of the IP header, IHL = 5
IP TOS	IP type of service
Total Length	Length in octets of IP header and data
Identification	IP fragmentation identification
Flags	IP control flags; must be set to 010 to avoid fragmentation
Fragment Offset	Indicates where in the datagram the fragment belongs; not used for TDM-over-packet
Time to Live	IP Time-to-Live field; datagrams with zero in this field are to be discarded
Protocol	Must be set to 0x11 to signify UDP
IP Header Checksum	Checksum for the IP header
Source IP Address	IP address of the source
Destination IP Address	IP address of the destination

Table 3. UDP Header Structure

Field	Description
Source Port Number, Destination	Either the Source or the Destination Port Number holds the bundle identifier. The unused field can be set to 0x85E (2142), which is the user port number assigned to TDM-over-packet by the Internet Assigned Numbers Authority (IANA). For UDP/IP-specific OAM packets, the

Port Number	bundle identifier is all ones.
UDP Length	Length in octets of UDP header and data
UDP Checksum	Checksum of UDP/IP header and data; if not computed, it must be set to zero

UDP/IPv6 Header

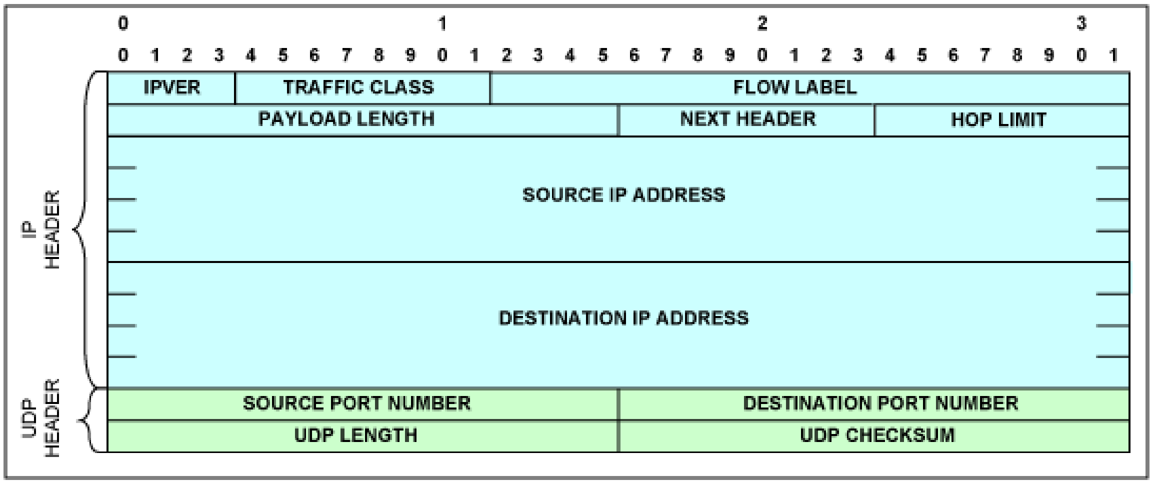


Figure 5. UDP/IPv6 header.

Table 4. IPv6 Header Structure

Field	Description
IPVER	IP version number; for IPv6 IPVER = 6
Traffic Class	An 8-bit field similar to the type-of-service (ToS) field in IPv4
Flow Label	The 20-bit Flow Label field can be used to tag packets of a specific flow to differentiate the packets at the network layer.
Payload Length	Similar to the Total Length field in IPv4, this field indicates the total length of the IP header and data in octets.
Next Header	Similar to the Protocol field in IPv4, this field determines the type of information following the basic IPv6 header. It must be set to 0x11 to signify UDP.
Hop Limit	Similar to the Time-to-Live field in IPv4
Source IP Address	Similar to the Source Address field in IPv4, except that this field contains a 128-bit source address for IPv6 instead of a 32-bit source address for IPv4.
Destination Address	Similar to the Destination Address field in IPv4, except that this field contains a 128-bit destination address for IPv6 instead of a 32-bit destination address for IPv4.

MPLS Header

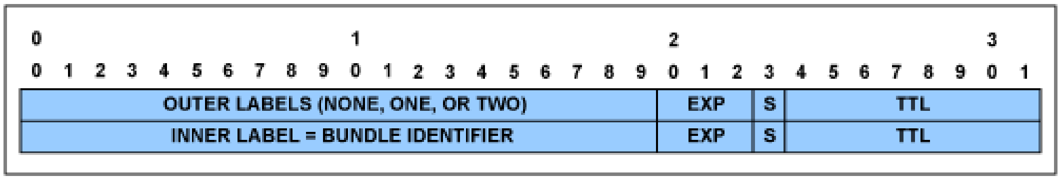


Figure 6. MPLS header.

Table 5. MPLS Header Structure

Field	Description
Outer Labels	These MPLS labels identify the MPLS LSP used to tunnel the TDMoMPLS packets through the MPLS network. They are also known as tunnel labels or transport labels. The label number can be assigned either manually or via the MPLS control protocol. There can be zero, one, or two outer labels.
EXP	Experimental field
S	Stacking bit: 1 indicates stack bottom; S = 0 for all outer labels
TTL	MPLS time to live
Inner Label	The MPLS Inner Label (also known as the PW label or the interworking label) contains the bundle identifier used to multiplex multiple bundles within the same tunnel. It is always at the bottom of the MPLS label stack, and hence its stacking bit is set.

MEF Header

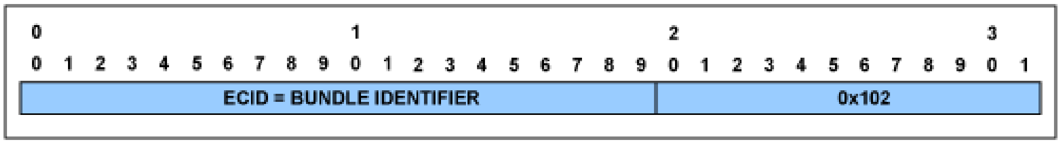


Figure 7. MEF header.

Table 6. MEF Header Structure

Field	Description
ECID	The Emulated Circuit Identifier (ECID) contains the bundle identifier.

L2TPv3/IPv4 Header

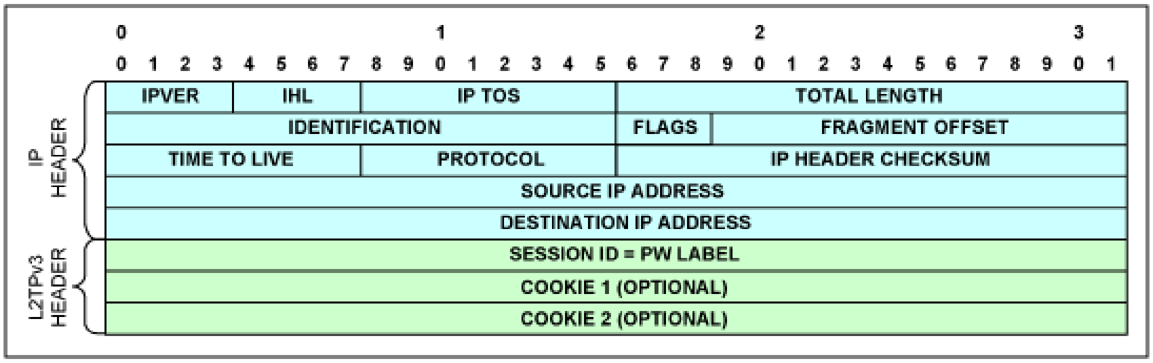


Figure 8. L2TPv3/IPv4 header.

Table 7. L2TPv3/IPv4 Header Structure

Field	Description
IPVER	IP version number; e.g., for IPv4 IPVER = 4
IHL	Length in 32-bit words of the IP header, IHL = 5
IP TOS	IP type of service
Total Length	Length in octets of header and data
Identification	IP fragmentation identification
Flags	IP control flags; must be set to 010 to avoid fragmentation

Fragment Offset	Indicates where in the datagram the fragment belongs; not used for TDM-over-packet
Time to Live	IP Time-to-Live field; datagrams with zero in this field are to be discarded
Protocol	Must be set to 0x73 to signify L2TPv3
IP Header Checksum	Checksum for the IP header
Source IP Address	IP address of the source
Destination IP Address	IP address of the destination

Table 8. L2TPv3 Header Structure

Field	Description
Session ID (32 Bits)	Locally significant L2TP session identifier, also contains the bundle identifier; all bundle identifiers are available for use except 0, which is reserved
Cookie (32 or 64 Bits)	Optional field that contains a randomly selected value used to validate association of the packet with the expected bundle identifier

L2TPv3/IPv6 Header

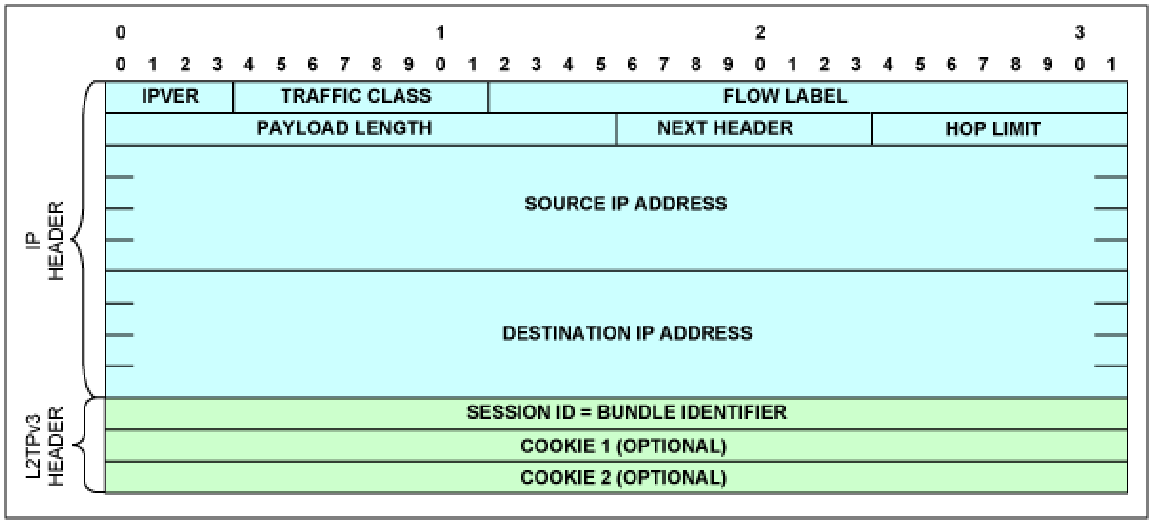


Figure 9. L2TPv3/IPv6 header.

Table 9. L2TPv3/IPv6 Header Structure

Field	Description
IPVER	See Table 4
Traffic Class	
Flow Label	
Payload Length	
Next Header	Must be set to 0x73 to signify L2TPv3
Hop Limit	See Table 4
Source Address	
Destination Address	

See Table 8 for the L2TPv3 header structure.

Control Word

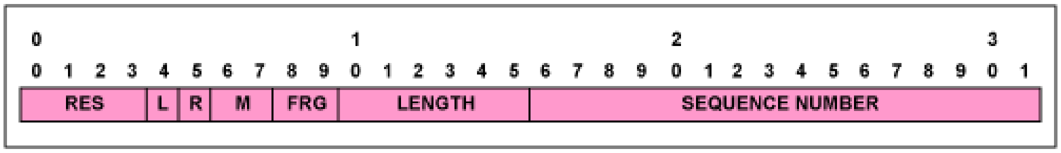


Figure 10. Control Word.

Table 10. Control Word Structure

Field	Description
RES	Reserved bits—must be set to zero
L	Local loss-of-sync (LOS) failure. This bit is set by the CPU. A set L bit indicates that the source has detected, or has been informed of, a TDM physical layer fault that impacts the data to be transmitted. This bit can be used to indicate physical layer LOS that should trigger AIS generation at the far end. Once set, if the TDM fault is rectified, the L bit must be cleared.
R	Remote receive failure. This bit is set by the CPU. A set R bit indicates that the source is not receiving packets at the Ethernet port (i.e., there is a failure in the direction of the bidirectional connection). This indication can be used to signal congestion or other network-related faults. A remote failure indication may trigger fallback mechanisms for congestion avoidance. The R bit must be set after a preconfigured number of consecutive packets are not received, and must be cleared once packets are received again.
M	Defect modifier failure. These bits are set by the CPU. This field is optional. When used, it supplements the L-bit meaning.
FRG	Fragmentation field. This field is used for fragmenting multiframe structures into multiple packets in case of CESoPSN structured with CAS bundles. The field is used as follows: 00 - Indicates that the entire (unfragmented) multiframe structure is carried in a single packet 01 - Indicates the packet carrying the first fragment 10 - Indicates the packet carrying the last fragment 11 - Indicates a packet carrying an intermediate fragment
Length	Includes control word, payload, and RTP header (if it exists), unless it is a UDP/IP packet. It is used when this sum is less than 64 bytes. Otherwise, set to zero.
Sequence Number	TDM-over-packet sequence number. This value is defined separately for each bundle and incremented by one for each TDMoP packet sent for that bundle. The initial value of the sequence number is random (unpredictable) for security purposes, and the value is incremented in wrap-around manner separately for each bundle. It is used by the receiver to detect packet loss and restore packet sequence. The HDLC payload type machine supports three different modes for this field: always zero, incremented in wrap-around manner, or incremented in wrap-around value, but skips zero value. For OAM packets (see TDM-over-packet payload), it uniquely identifies the message. Its value is unrelated to the sequence number of the TDMoP data packets for the bundle in question. It is incremented in query messages, and replicated without change in replies.

RTP Header

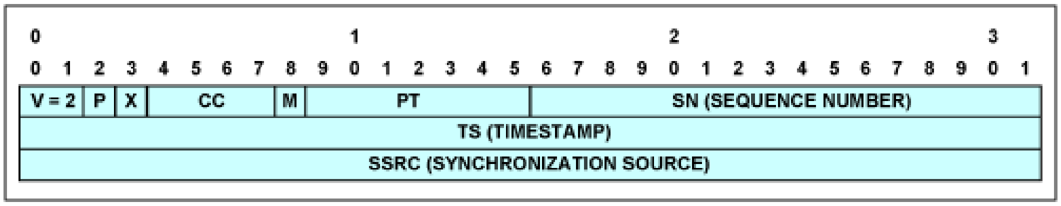


Figure 11. RTP header.

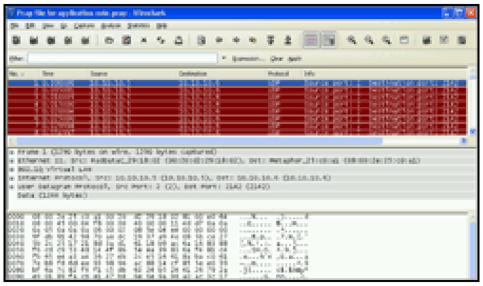
Table 11. RTP Header Structure

Field	Description
V	RTP version—must be set to 2
P	Padding bit—must be set to 0
X	Extension bit—must be set to 0
CC	CSRC count—must be set to 0
M	Marker bit—must be set to 0
PT	Payload Type. One PT value MUST be allocated from the range of dynamic values for each direction of the bundle. The same PT value MAY be reused for both directions of the bundle, and also reused between different bundles.
SN	The sequence number, identical to the sequence number in the control word
TS	Timestamp. The RTP header can be used in conjunction with the following modes of timestamp generation: Absolute mode: the chip sets timestamps using the clock recovered from the incoming TDM circuit. As a consequence, the timestamps are closely correlated with the sequence numbers. The timestamp is incremented by one every 125µs. Differential (common-clock) mode: The two chips at bundle edges have access to the same high-quality clock source, and this clock source is used for timestamp generation.
SSRC	Identifies the synchronization source. This identifier should be chosen randomly, with the intent that no two synchronization sources within the same RTP session will have the same SSRC identifier.

How to Know the Packet Contents from Other Vendors' TDMoP Devices

Software is available to analyze the Ethernet packet headers. Wireshark® software was used for this application note. The user can download this freeware from www.wireshark.org/download.html. For more information about Wireshark, go to Wireshark Frequently Asked Questions.

In order to confirm that the user is sending the right packets with the right protocol, the user needs to make sure that the two TDMoP system boards from other vendors are in sync with each other. After that, the user needs to capture the packets using the Wireshark program. The screenshot of this program is shown in **Figure 12**.



More detailed image (PDF)

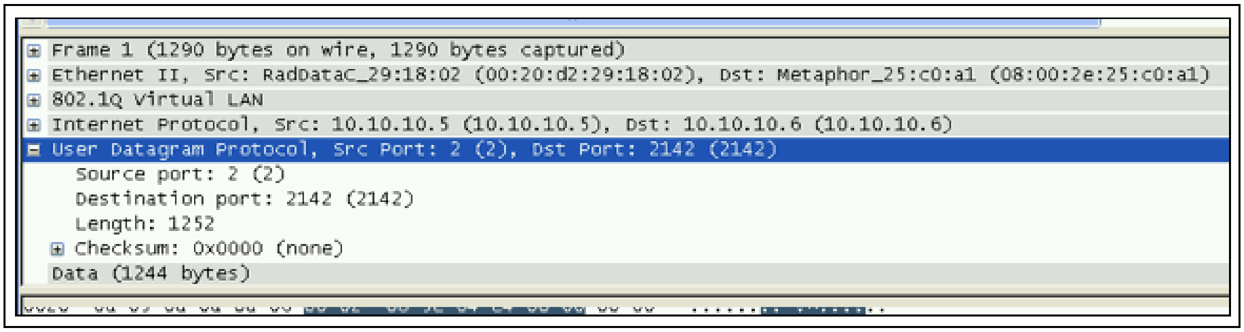
Figure 12. A screenshot of the Wireshark program used to analyze Ethernet packet headers.

The following requirements must be considered to make the systems interoperable.

1. **Source Port Number and Destination Port Number**
2. **Total Number of Packet Bytes, IP Length, UDP Length, and Data Bytes**
3. **Ether Type**

Source Port Number and Destination Port Number

The TDMoIP_Port_Number is used by the Packet Classifier block to identify the UDP/IP of TDM-over-packets. Two different values can be configured as the TDMoIP_Port_Number. Although Maxim's TDMoP devices have two TDMoIP_Port_Number registers, in most cases both registers should have the default value (0x085E) as assigned by IANA for TDM-over-packet. Either the Source or the Destination Port Number holds the bundle identifier. The unused field can be set to 0x85E (2142 in decimal), which is the user port number assigned to TDM-over-packet by the IANA. As shown in Figure 4, Maxim's devices insert Source Port Number first and then TDMoP Destination Port Number. **Figure 13** shows the contents of the User Datagram Protocol (UDP) where Source Port Number is set as 2 and the TDMoIP Destination Port Number is set as 0x85E (2142 in decimal).



```
⊞ Frame 1 (1290 bytes on wire, 1290 bytes captured)
⊞ Ethernet II, Src: RadDataC_29:18:02 (00:20:d2:29:18:02), Dst: Metaphor_25:c0:a1 (08:00:2e:25:c0:a1)
⊞ 802.1Q Virtual LAN
⊞ Internet Protocol, Src: 10.10.10.5 (10.10.10.5), Dst: 10.10.10.6 (10.10.10.6)
⊟ User Datagram Protocol, Src Port: 2 (2), Dst Port: 2142 (2142)
    Source port: 2 (2)
    Destination port: 2142 (2142)
    Length: 1252
  ⊞ Checksum: 0x0000 (none)
  Data (1244 bytes)
```

Figure 13. UDP Source and Destination Port Numbers.

Some vendors insert 0x85E as the UDP Source Port Number and UDP Destination Port Number. In this scenario, the user must use the preconfiguration menu to configure the system. The default Maxim SW menu is shown below:

```
                PreConfig Configuration

1.   Link Type                          E1
2.   Bundle Number ID Location         Port in DST, Bundle in SRC UDP Port
3.   UDP Mask                           1FFF
4.   VCCV OAM Mask [0 - 4]              0
5.   VCCV OAM Value                     1FFF
6.   MEF Ethernet Type                  88D8
7.   MEF OAM Type                       0
8.   TDMoIP Port Number 1               85E
9.   Oscillator Type                    OCXO (Stratum 3E)
10.  RTP Clock Source                   ABSOLUTE
```

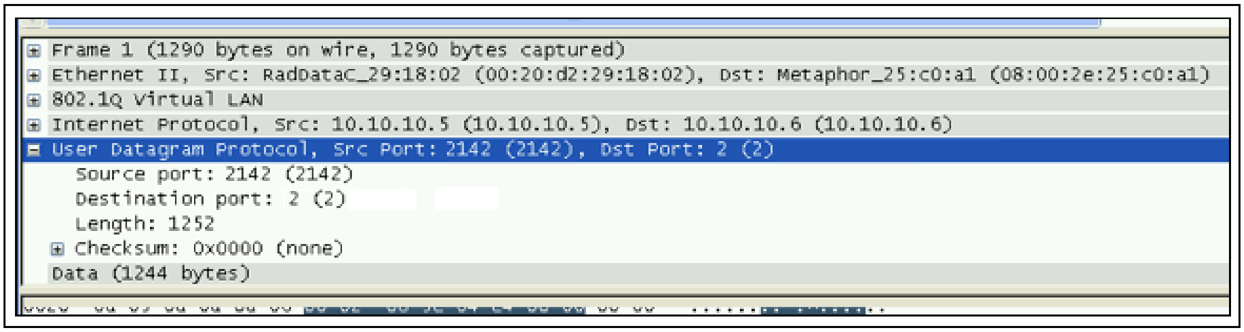
```
11.  Common clock Rate                19440000
12.  IP Version                       IPv4
13.  Clock Recovery Smart Statistics  Enable
14.  One or Two Clock Mode            One
```

Item 2 from the Maxim SW menu is used to select the desired Bundle Number ID Location. Item 2 from the above menu provides the following section:

```
                   Bundle Number ID Location

1: Ignore port, Bundle in SRC UDP PORT,
2: Port in DST, Bundle in SRC UDP PORT
3: Port in SRC, Bundle in DST UDP PORT,
4: Ignore Port, Bundle in DST UDP PORT
```

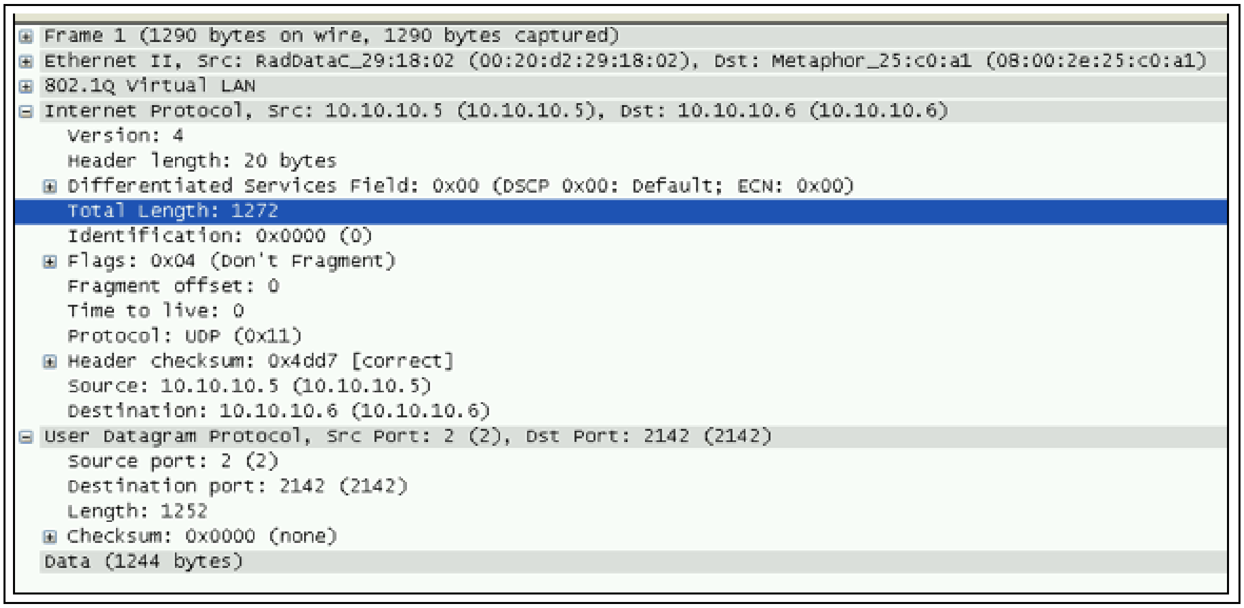
Maxim's devices default Bundle Number ID Location is Item 2 in the above menu: *"Port in DST, Bundle in SRC UDP PORT."* To enable Maxim's devices to work with other vendors' devices, the user needs to select Item 1, 3, or 4 as appropriate. For example, one of the TDMoP device vendor's devices inserts Destination port at Source (SRS) location and the bundle port number at Destination (DST). If the user chooses Item 3 from the above menu, then the UDP Source port Bundle Number ID Location will be set to 0x85E (2142 decimal) and the UDP destination port will have 2 as shown in **Figure 14**. This will match that vendor's TDMoP packet header and, thus, will enable interoperability.



```
⊞ Frame 1 (1290 bytes on wire, 1290 bytes captured)
⊞ Ethernet II, Src: RadDataC_29:18:02 (00:20:d2:29:18:02), Dst: Metaphor_25:c0:a1 (08:00:2e:25:c0:a1)
⊞ 802.1Q Virtual LAN
⊞ Internet Protocol, Src: 10.10.10.5 (10.10.10.5), Dst: 10.10.10.6 (10.10.10.6)
⊟ User Datagram Protocol, Src Port: 2142 (2142), Dst Port: 2 (2)
    Source port: 2142 (2142)
    Destination port: 2 (2)
    Length: 1252
  ⊞ Checksum: 0x0000 (none)
  Data (1244 bytes)
```

Figure 14. UDP Source and Destination port numbers reversed to that of Figure 13.

Total Number of Packet Bytes, IP Length, UDP Length, and Data Bytes



```
⊞ Frame 1 (1290 bytes on wire, 1290 bytes captured)
⊞ Ethernet II, Src: RadDataC_29:18:02 (00:20:d2:29:18:02), Dst: Metaphor_25:c0:a1 (08:00:2e:25:c0:a1)
⊞ 802.1Q Virtual LAN
⊟ Internet Protocol, Src: 10.10.10.5 (10.10.10.5), Dst: 10.10.10.6 (10.10.10.6)
    Version: 4
    Header length: 20 bytes
  ⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 1272
    Identification: 0x0000 (0)
  ⊞ Flags: 0x04 (Don't Fragment)
    Fragment offset: 0
    Time to live: 0
    Protocol: UDP (0x11)
  ⊞ Header checksum: 0x4dd7 [correct]
    Source: 10.10.10.5 (10.10.10.5)
    Destination: 10.10.10.6 (10.10.10.6)
⊟ User Datagram Protocol, Src Port: 2 (2), Dst Port: 2142 (2142)
    Source port: 2 (2)
    Destination port: 2142 (2142)
    Length: 1252
  ⊞ Checksum: 0x0000 (none)
  Data (1244 bytes)
```

Figure 15. Captured packets showing different packet length information.

Figure 15 shows the content of the packets with different packet length information numbered as 1.

The user must consider the following lengths:

A. **Data Bytes**: Figure 15 shows that packet 1 has 1244 bytes. In the bundle configuration, we used the IP/UDP/CESoPSN protocol. We were sending E1 TDM data using 31 timeslots. Each timeslot had 40 frame bytes. The total number of TDM data frame bytes is 40 × 31 = 1240 frame bytes. The addition of 4 bytes of control words makes it 1244 bytes. One of the many advantages of using Maxim's TDMoP devices is that in adaptive clock-recovery mode, the default mode does not use a RTP (real-time protocol) header in the packets, thus freeing some BW for payload data. Most of the other vendors use 12 bytes of RTP. If we used RTP in the TDMoP packets, then the data bytes would be 1256 (1244 + 12). After knowing the total number of TDM data bytes, which is 1240 bytes in this case, the user needs to program the Maxim device so that it also generates 1240 bytes of TDM data, or the number that was found with the Wireshark program.

B. **UDP Length**: Figure 15 shows that packet 1 has a UDP length of 1252 bytes, which is 1244 bytes of data and 8 bytes of UDP protocol.

C. **IP Length**: Figure 15 shows that packet 1 has an IP length of 1272 bytes, which is 1244 bytes of data, 20 bytes of IP header, and 8 bytes of UDP protocol header.

D. **Total Number of Frame Bytes**: Figure 15 shows that packet 1 contains 1290 bytes. This is 1244 bytes of data with 20 bytes of IP header, 8 bytes of UDP protocol header, 2 bytes of Ether type, 4 bytes of VLAN tag, and 12 bytes of source and destination MAC address.

Interoperability requires that all the packet lengths match. If these lengths are not the same, the user must use the SW menu to configure Maxim's TDMoP devices to have the same packet lengths.

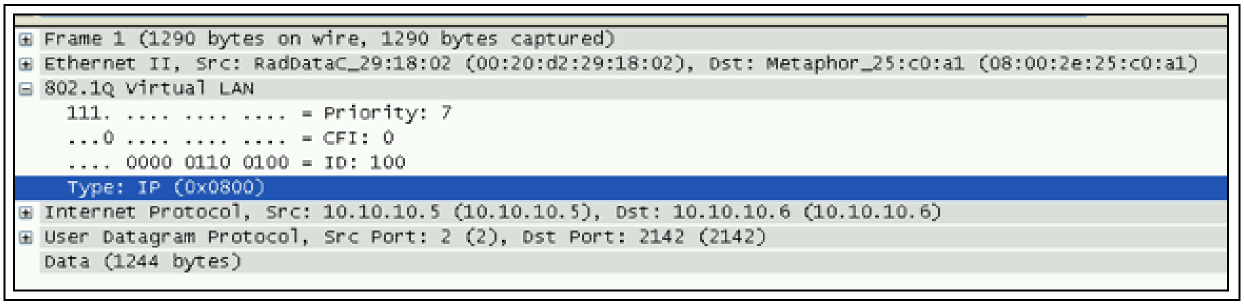
Ether Type

Maxim's TDMoP devices consider the following Ether types as known Ether types:

1. IPv4 (0x800)

2. IPv6 (0x86DD)
3. MPLS unicast (0x8847)
4. MPLS multicast (0x8848)
5. ARP (0x806)
6. MEF Ether type as configured in the Mef_ether_type configuration register
7. MEF OAM Ether type as configured in the Mef_oam_ether_type configuration register
8. Specific Ether type as configured in the CPU_dest_ether_type configuration register

To make Maxim's TDMoP devices interoperable, the user must determine the Ether type of the incoming packets from the other TDMoP devices. The type is located after the VLAN ID header bytes. **Figure 16** shows that the incoming packet Ether type is 0x800, which indicates the packet to be IPv4.



```
⊞ Frame 1 (1290 bytes on wire, 1290 bytes captured)
⊞ Ethernet II, Src: RadDataC_29:18:02 (00:20:d2:29:18:02), Dst: Metaphor_25:c0:a1 (08:00:2e:25:c0:a1)
⊟ 802.1Q Virtual LAN
    111. .... .... .... = Priority: 7
    ...0 .... .... .... = CFI: 0
    .... 0000 0110 0100 = ID: 100
    Type: IP (0x0800)
⊞ Internet Protocol, Src: 10.10.10.5 (10.10.10.5), Dst: 10.10.10.6 (10.10.10.6)
⊞ User Datagram Protocol, Src Port: 2 (2), Dst Port: 2142 (2142)
  Data (1244 bytes)
```

Figure 16. The Ether type value is 0x800, which indicates that it is IPv4.

Once the Ethernet type is determined, the user must configure Maxim's TDMoP devices to generate the same Ether type packets. The Ether type is selected from the Bundle Configuration menu by changing the PSN type. A portion of the Bundle Configuration menu is shown below.

```
Main Menu>Bundle Configuration>CES Bundle Configuration

... (P)
11. VLAN ID 1[1 - 4095]                      ... (100)
12. VLAN Priority[0 - 7]                     ... (7)
13. IP Tos[0 - 255]                          ... (0)
14. IP TTL[0 - 255]                          ... (128)
15. PSN Type                                 >   (IP)
```

Item 15 from the above menu provides the following section:

```
Main Menu>Bundle Configuration>CES Bundle Configuration>PSN Type ()

 1. IP
 2. MPLS
 3. L2TPV3
 4. Ethernet
```

By choosing the desired combination from the Bundle Configuration menu, the Ether type of the captured packets is matched.

Conclusion

Interoperability refers to the ability of diverse systems and organizations to work together (interoperate). Products achieve interoperability with other products either by adhering to published interface standards or by allowing configuration changes that convert one product's interface into another product's interface "on the fly." By knowing the contents of the packets generated by other TDMoP devices, Maxim's devices can be configured easily to match packet configurations of other TDMoP devices.

If you have further questions on TDMoP products or any other aspect of using Maxim's telecom products, please contact the telecom products applications support team by email at telecom.support@maximintegrated.com or by phone at 972-371-6555.

IEEE is a registered service mark of the Institute of Electrical and Electronics Engineers, Inc.
Wireshark is a registered trademark of the Wireshark Foundation, Inc.

Related Parts		
DS34S101	Single/Dual/Quad/Octal TDM-Over-Packet Transport Devices	Free Samples
DS34S102	Single/Dual/Quad/Octal TDM-Over-Packet Transport Devices	Free Samples
DS34S104	Single/Dual/Quad/Octal TDM-Over-Packet Transport Devices	Free Samples
DS34S108	Single/Dual/Quad/Octal TDM-Over-Packet Transport Devices	Free Samples
DS34T101	Single/Dual/Quad/Octal TDM-Over-Packet Chip	Free Samples
DS34T102	Single/Dual/Quad/Octal TDM-Over-Packet Chip	Free Samples
DS34T104	Single/Dual/Quad/Octal TDM-Over-Packet Chip	Free Samples
DS34T108	Single/Dual/Quad/Octal TDM-Over-Packet Chip	Free Samples

More Information
For Technical Support: http://www.maximintegrated.com/support
For Samples: http://www.maximintegrated.com/samples
Other Questions and Comments: http://www.maximintegrated.com/contact

Application Note 4158: http://www.maximintegrated.com/an4158
APPLICATION NOTE 4158, AN4158, AN 4158, APP4158, Appnote4158, Appnote 4158