Keywords: Jitter Buffer, Jitter, Buffer, TDMoP, TDMoIP, SAToP, CESoP, CESoPSN

APPLICATION NOTE 4115

How to Use Jitter Buffers on TDMoP Products to Compensate for Packet-Delay Variation (PDV)

Sep 24, 2007

Abstract: The DS34T10x and DS34S10x TDM-over-Packet (TDMoP) devices utilize jitter buffers to compensate for the packet-delay variation (PDV) that is present in packet networks. This application note explains PDV and how it affects communication quality. The function and types of jitter buffers are discussed. The article also describes how to set the jitter buffer controller's parameters in TDMoP devices to minimize the effects of PDV.

Introduction

The DS34T10x and DS34S10x families of TDM-over-Packet (TDMoP) devices use jitter buffers to compensate for the packet-delay variation (PDV) that is present in packet networks. These buffers are independently configurable on a per-bundle or per-connection basis. Additionally, they are dynamically adjustable, allowing them to be adapted in real-time to changes in the performance characteristics of the packet network. This application note discusses the jitter buffer controller and how to set its parameters to minimize the effects of PDV during TDM clock recovery.

DS34T10x comprises the DS34T101, DS34T102, DS34T104, and DS34T108; DS34S10x comprises the DS34S101, DS34S102, DS34S104, and DS34S108.

Timing in a TDM Network

TDM networks have a single-clock source used by all devices in the network. Destination TDM devices derive the clock from incoming data and use it for transmitting data (loopback timing), as shown in **Figure 1**.

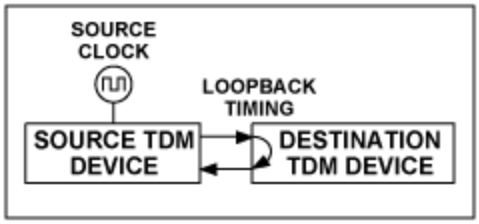

Figure 1. Loopback timing in a TDM network.

Variations in packet arrival time, called jitter, occur because of network congestion, timing drift, or route

changes. Thus, when replacing the physical TDM connection with an IP/MPLS network and two TDMoP devices (as shown in **Figure 2**), the receiving TDMoP device (slave) receives TDMoP packets with variable delays in arrival time.

After processing the packets, the device should send TDM data to the TDM side at the constant rate of the TDM network to minimize the effects of this jitter. To achieve this constant data rate, the device works in clock-recovery mode to reconstruct the source TDM clock so that the destination TDM device can still work in loopback timing mode.

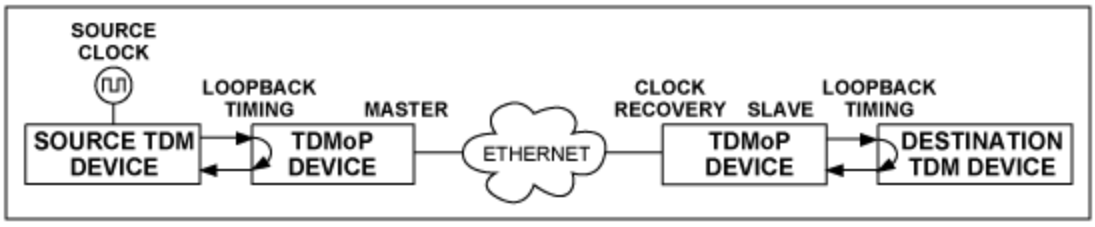

Figure 2. Timing in a TDM-over-Packet network.

Jitter Buffers

The DS34T10x/DS34S10x utilize jitter buffers to minimize the effects of PDV on communication quality. A jitter buffer is a shared memory area where TDM packets can be collected, stored, and sent to the circuit emulation engine in evenly spaced intervals. Located at the sending and receiving ends of the TDM connection, the jitter buffer intentionally delays the arriving packets so that the end user experiences a clear connection with very little to no sound distortion.

There are two kinds of jitter buffers: static and dynamic. The static jitter buffer is hardware-based and is configured by the manufacturer. The dynamic jitter buffer is software-based and can be configured by a network administrator to adapt to changes in the network's delay and PDV.

The DS34T10x/DS34S10x feature dynamic jitter buffers, located in the SDRAM. These jitter buffers have two main roles:

- Compensating for packet-delay variation
- Reconstructing the far-end TDM clock on a TDMoP slave device

Data enters the buffer at a variable rate derived from the received Ethernet packet's arrival time. Data leaves the buffer at a constant TDM rate. In clock-recovery mode, the level of the jitter buffer provides an indication of the clock-recovery mechanism.

For TDMoP protocols (CESoPSN, SAToP, and TDMoIP), bundles can comprise any number of 64kbps timeslots originating from a single E1 or T1. Bundles are single-direction streams, frequently coupled with bundles in the opposite direction to enable full-duplex communications. More than one bundle can be transmitted between two TDMoP edge devices.

The DS34T10x/DS34S10x provide large jitter buffers (up to 64 bundles) that are configurable on a per-bundle basis to compensate for the delay variation introduced by the IP/MPLS/Ethernet network. Each bundle can be assigned to any TDM port on a payload-type machine or CPU. All bundles feature the following independently configurable functions:

- Transmit and receive queues
- Receive-jitter-buffer depth
- Optional connection-level redundancy (for both SAToP and CESoPSN)

Each device provides internal bundle-crossconnect functionality with DS0 resolution. Additionally, each bundle can be configured to: transport all the T1/E1 data from a port; transmit a configured number of bytes in each packet (SAToP); or transport specific timeslots of the T1/E1 data, with up to 24 timeslots for T1 and 31 timeslots for E1 (CESoPSN). A bundle can contain data from any timeslot within a single TDM port. However, a timeslot can only be assigned to a single bundle. For the SAToP and CESoPSN bundles, TDMoP devices perform packet reordering within the range of the jitter buffer. Packet loss is compensated by inserting either a preconfigured conditioning value or replaying the last received value.

The resolution of SAToP and CESoP jitter buffers is different. For SAToP, precision is in bytes, so the variables can be sized in increments around 4µs for E1 and 5µs for T1. For CESoP, precision is in frames, so the variables can be sized by 125µs increments for both E1 and T1. Because the entire packet must be stored before determining whether or not the packet is good, the minimum precision for the jitter buffer is a function of packet size. If it is a small packet (1 byte), then the minimum is 1 byte. If it is a large CES packet (1500 frames), then the minimum is 187.5ms, regardless of PDV.

Configuring the DS34T10x/DS34S10x Jitter Buffers

Configuring the jitter buffer's parameters correctly avoids underrun and overrun situations. Underrun occurs when the jitter buffer is empty (the entering rate is lower than the exiting one). When an underrun event occurs, the chip transmits conditioning data instead of actual data to the TDM interface. Overrun occurs when the jitter buffer is full and there is no room for new data to enter (the entering rate exceeds the exiting one). Underrun and overrun require special treatment from the IC's hardware, depending on the bundle type.

The DS34T10x/DS34S10x allocate separate areas in the external SDRAM for data and for signaling.

- In 8-port low-speed mode, both data and signaling areas are divided into eight identical sections, one for each E1/T1/Nx64 interface.
    - In E1/T1 structured mode, each data section contains the data of 32 timeslots for E1 or 24 timeslots for T1; a single E1/T1 timeslot is allocated a maximum of 4kB of space. There is a total of 256 timeslots and 1024kB of space for all eight interfaces.
    - Each signaling section is divided into multiframe sectors, with each multiframe sector containing the signaling nibbles of up to 32 timeslots. There is a total of 64kB for all eight interfaces.
    - In serial or E1/T1 unstructured mode, there is no per-timeslot allocation. The jitter buffer is divided into eight identical sections, one for each interface. Each section is 512kB per HDLC bundle, or 128kB otherwise.
- For high-speed mode (E3/T3, STS-1), the jitter buffer acts as one large buffer, without division into sections and with a total of 512kB.

The jitter buffers have the following depths:

- E1: up to 256ms
- T1 unframed: up to 340ms
- T1 framed: up to 256ms
- T1 framed with CAS: up to 192ms

The jitter buffer's maximum depth in units of time is calculated according to the following formula:

½ × Buffer area per interface x (8 / Rate) *(Eq. 1)*

where:

½ = Two halves of the buffer

Buffer area per interface	= 512kB for a single high-speed interface, or 128kB for a low-speed interface
8	= Number of bits per byte
Rate	= Transmission rate (e.g. 2.048Mbps)

For T1 framed with CAS, multiply the result of Equation 1 by 0.75.

The jitter buffer's depth is defined by the `Rx_max_buff_size` parameter found in the Bundle Configuration tables. When the jitter buffer level reaches the value of `Rx_max_buff_size`, an overrun situation is declared.

The `Rx_pdvt` parameter (also found in the Bundle Configuration tables) defines the amount of data to be stored in the jitter buffer to compensate for network-delay variation. **Figure 3** shows the jitter buffer parameters. The `Rx_pdvt` parameter has two implications:

- `Rx_pdvt` defines the IC's immunity to the Ethernet network-delay variation
- The data arriving from the network is delayed by `Rx_pdvt` before it is sent to the TDM line

`Rx_pdvt` should be smaller than `Rx_max_buff_size`. Also, the difference between `Rx_max_buff_size` and `Rx_pdvt` must be larger than the time that it takes to reconstruct a packet; otherwise, an overrun may occur when the packet arrives. Typically, the recommended value for the `Rx_max_buff_size` is **2 × `Rx_pdvt` + PCT** (packet creation time). This provides equal immunity for both delayed and bursty packets.

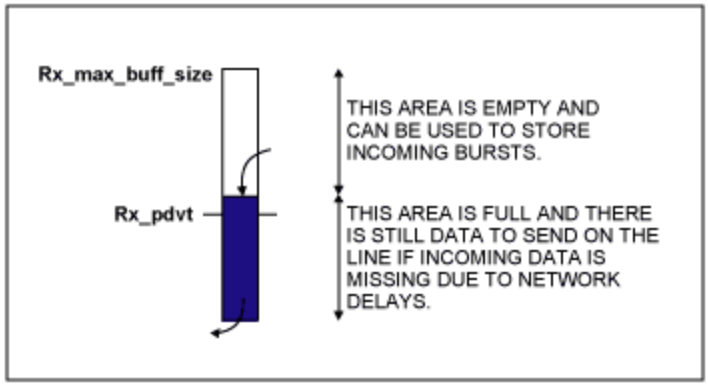

Figure 3. Jitter buffer parameters.

The jitter buffer controller (JBC) uses a 64-bit by 32-bit Bundle Timeslots table to identify the assigned timeslots of each active bundle. The index to the table is the bundle number. The software must configure each active bundle entry. For unstructured bundles, the whole bundle entry (32 bits) must be set—setting a bit means that the corresponding timeslot is assigned to this bundle.

JBC statistics are stored in a 128-entry table. Each TDM port has 32 dedicated entries—one per timeslot. The Jitter Buffer Status table stores the statistics of an active jitter buffer for each active bundle. A configurable parameter called `Jitter_buffer_index` (located in the TSA tables) defines the entry in the table in which jitter buffer statistics are stored, and from which they are read.

The software accesses the Jitter Buffer Status table according to the `Jitter_buffer_index`. The status table contains the current jitter buffer status, such as the jitter buffer level and state (e.g. ok, underrun, or overrun). It also contains two variables, which report the jitter buffer minimum and maximum levels. These variables provide information about network characteristics for the user. For example,

using these values, the user can calculate the margins from the top (Rx_max_buff_size) and the bottom of the jitter buffer. If there is spare capacity, the user may want to reduce Rx_pdvt to limit the latency added by the jitter buffer to the incoming data.

The user should define the value of the Jitter_buffer_index as follows:

- For AAL1/HDLC/RAW structured bundles, the Jitter_buffer_index is the concatenation of the interface number (2 Msbits) and the number of the lowest timeslot in the bundle. For example, if the bundle consists of timeslots 2, 4, and 17 on the third interface, the Jitter_buffer_index is 10_00010[bin], i.e., 42[hex].
- For unstructured bundles, the jitter_buffer_index is the concatenation of the interface number (2 Msbits) and five '0's.
- For AAL2 bundles, each timeslot data is stored in its own jitter buffer; therefore, the Jitter_buffer_index is the concatenation of the interface number (2 Msbits) and the timeslot number. For example, if the bundle consists of timeslots 2, 4, and 17 on the first interface, there are three jitter buffers—one for each timeslot—and the Jitter_buffer_index values are 2[hex], 4[hex] and 11[hex], respectively.

The payload-type machine detects whether or not a packet was lost, either by sequence number error (AAL1/RAW) or by UUI error (AAL2). If packets are lost, conditioning data is inserted into the jitter buffer to compensate for the lost data and to maintain bit integrity. In simple terms, the number of bits inserted into the jitter buffer must equal the number of bits transmitted by the far end.

If a packet is misordered in a RAW bundle—for example, the packet with the sequence number N arrives after the packet with sequence number N+1—it is reordered by the RAW payload-type machine. The packet's data is inserted into the appropriate location in the jitter buffer, assuming that the data in this location has not been transmitted to the TDM domain yet.

Conclusion

A jitter buffer temporarily stores arriving packets to minimize delay variations. If packets arrive too late, then they are discarded. Sometimes jitter buffers are misconfigured so that they are either too small or too large. If a jitter buffer is too small, then an excessive number of packets may be discarded, which can lead to call quality degradation. If a jitter buffer is too large, then the additional delay can lead to conversational difficulty. Configuring the jitter buffer parameters correctly avoids these underrun and overrun situations.

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 01-972-371-6555.

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 4115: http://www.maximintegrated.com/an4115
APPLICATION NOTE 4115, AN4115, AN 4115, APP4115, Appnote4115, Appnote 4115