A communications system, comprising a first device including a first scan table and a second device including a second scan table. The first device is configured to select the first scan table and transmit scan table identification. The second device is configured to receive the scan table identification from the first device and select the second scan table based on the received scan table identification.
COMMUNICATIONS SYSTEM WITH SCAN TABLE IDENTIFICATION

Background

Computer system speeds continue to increase and more computer systems are connected to communicate with other computer systems daily. As the volume of digital data communicated between computer systems increases, there is a need to develop higher bandwidth communication links. Often, these communication links are part of a network, such as a local area network (LAN), metro area network (MAN), or a wide area network (WAN).

A network includes network nodes that provide network related functions. Each network node is a grouping of one or more network elements, such as computer systems, and each network node includes one or more communication links connected to the network. In addition, each network node is administered as a single entity. Network elements in a network node can be at one or more sites and a single site may contain more than one network node. Network elements on a network communicate with other network elements on the network by employing some type of suitable network communication, such as synchronous communication and/or asynchronous communication.

In synchronous communication, transmission system payloads are synchronized to a master clock that is traceable back to a highly stable reference clock. The digital transitions in signals occur at essentially the same rate. Synchronous communication often uses time division multiplexing (TDM) as a mechanism for dividing the bandwidth of a communication link into separate channels or time slots. In TDM, multiple data streams are put into a single signal by separating the signal into many segments, each having a very short duration. Each individual data stream is reassembled at the receiving end based on the timing. A synchronous TDM interface transmits and receives data traffic at a constant bit rate.

One type of synchronous network is a synchronous optical network and synchronous digital hierarchy (SONET/SDH) ring network. SONET and SDH are a set of related standards for synchronous data transmission over fiber optic
networks. SONET is the United States standard published by the American National Standards Institute (ANSI) and SDH is the international standard published by the International Telecommunication Union (ITU). A SONET/SDH ring network is one medium of choice for delivering services over a MAN and/or a WAN.

In asynchronous communication, transmission system payloads are not synchronized to a master clock. Instead, each network element operates from its own clock. Data packets or frames are transmitted asynchronously as the packets become available for transmission.

In some networks, some network elements communicate with other network elements through one or more synchronous TDM interfaces and one or more asynchronous packet interfaces. Data traffic is transmitted at a constant bit rate through the synchronous TDM interfaces and data packets are transmitted through the asynchronous packet interfaces to maintain the data rate of each channel on the synchronous TDM network. Channel information is transmitted with the data through the asynchronous interfaces. However, transmitting channel information with data through an asynchronous interface can lead to the loss of data bandwidth through the asynchronous interface. Also, communication errors can occur if the channel information and data are misaligned or lost as the channel information and data are transmitted through the asynchronous interface.

For these and other reasons there is a need for the present invention.

**Summary**

One aspect of the present invention provides a communications system, comprising a first device including a first scan table and a second device including a second scan table. The first device is configured to select the first scan table and transmit scan table identification. The second device is configured to receive the scan table identification from the first device and select the second scan table based on the received scan table identification.

**Brief Description of the Drawings**
Figure 1 is a diagram illustrating one embodiment of a resilient packet ring network.

Figure 2 is a diagram illustrating one embodiment of a network node according to the present invention.

Figure 3 is a diagram illustrating one embodiment of a west reconciliation media access control (RMAC) device and an east RMAC device in a communications system according to the present invention.

Figure 4 is a diagram illustrating one embodiment of a segment packet.

Figure 5 is a diagram illustrating one example of segmenting and framing channel payloads into segment packets in one embodiment of a device.

Figure 6 is a diagram illustrating one embodiment of a segment status packet.

Figure 7 is a diagram illustrating one embodiment of a transmit (TX) manager.

Figure 8 is a diagram illustrating one embodiment of a receive (RX) manager.

Figure 9A is a diagram illustrating pointer operation in a TX scan table.

Figure 9B is a diagram illustrating pointer operation in an RX scan table.

Figure 10A is a diagram illustrating a transmit scan table add/drop register.

Figure 10B is a diagram illustrating a receive scan table add/drop register.

**Detailed Description**

In the following Detailed Description, reference is made to the accompanying drawings, which form a part hereof, and in which is shown by way of illustration specific embodiments in which the invention may be practiced. In this regard, directional terminology, such as “top,” “bottom,” “front,” “back,” “leading,” “trailing,” etc., is used with reference to the orientation of the Figure(s) being described. Because components of embodiments of the present invention can be positioned in a number of different orientations, the directional terminology is used for purposes of illustration and is in no way limiting. It is to be understood that other embodiments may be
utilized and structural or logical changes may be made without departing from
the scope of the present invention. The following detailed description, therefore,
is not to be taken in a limiting sense, and the scope of the present invention is
defined by the appended claims.

Figure 1 is a diagram illustrating one embodiment of a resilient packet
ring (RPR) network 40. RPR 40 includes network nodes A-D at 42a-42d,
respectively, a clockwise communications path 44, and a counter clockwise
communications path 46. Each of the nodes A-D at 42a-42d includes one or
more network elements and is communicatively coupled to clockwise
communications path 44 and to counter clockwise communications path 46.
RPR 40 is a synchronous network. In one embodiment, RPR 40 is a
synchronous TDM network such as a SONET/SDH network including TDM of
data.

Node A at 42a includes a west device 48 and an east device 50. West
device 48 is communicatively coupled to clockwise communications path 44 and
to counter clockwise communications path 46. Also, east device 50 is
communicatively coupled to clockwise communications path 44 and to counter
clockwise communications path 46. In addition, west device 48 is
communicatively coupled to east device 50 via communications path 52. West
device 48 communicates with east device 50 in asynchronous packet
communication via communications path 52. Each of the other nodes B-D 42b-
42d is similar to node A at 42a. In other embodiments, each of the nodes A-D
42a-42d can be different and include any suitable devices.

West device 48 and east device 50 receive and transmit data
synchronously in TDM channels or timeslots via clockwise communications
path 44 and counter clockwise communications path 46. West device 48 and
east device 50 communicate with each other in asynchronous packet
communication via communications path 52. West device 48 and east device 50
are configured to convert between synchronous and asynchronous
communication. In one embodiment, west device 48 and east device 50 are
configured to maintain a constant data rate or channel bandwidth in each of the
TDM channels of RPR 40.
West device 48 transmits and receives data to service one or more network elements. West device 48 receives data synchronously in TDM channels or timeslots via clockwise communications path 44 and west device 48 receives data asynchronously in packets from east device 50 via communications path 52. West device 48 can consume data received via clockwise communications path 44 and/or transmit the data asynchronously in packets to east device 50 via communications path 52. East device 50 can transmit data received via communications path 52 to other nodes, such as nodes B-D at 42b-42d, synchronously in TDM channels or timeslots via clockwise communications path 44. West device 48 can transmit data received in packets from east device 50 via communications path 52 to other nodes, such as nodes B-D at 42b-42d, synchronously in TDM channels or timeslots via counter clockwise communications path 46. In one embodiment, west device 48 can consume data received in packets from east device 50. In one embodiment, west device 48 can transmit data packets received from east device 50 back to east device 50 via communications path 52 and east device 50 can transmit the data from the data packets synchronously in TDM channels or timeslots on clockwise communications path 44.

East device 50 transmits and receives data to service one or more network elements. East device 50 receives data synchronously in TDM channels or timeslots via counter clockwise communications path 46 and east device 50 receives data asynchronously in packets from west device 48 via communications path 52. East device 50 can consume data received via counter clockwise communications path 46 and/or transmit the data asynchronously in packets to west device 48 via communications path 52. West device 48 can transmit data received via communications path 52 to other nodes, such as nodes B-D at 42b-42d, synchronously in TDM channels or timeslots via counter clockwise communications path 46. East device 50 can transmit data received in packets from west device 48 via communications path 52 to other nodes, such as nodes B-D at 42b-42d, synchronously in TDM channels or timeslots via clockwise communications path 44. In one embodiment, east device 50 can consume data received in packets from west device 48. In one embodiment, east device 50 can transmit data packets received from west device 48 back to west
device 48 via communications path 52 and west device 48 can transmit the data from the data packets synchronously in TDM channels or timeslots on counter clockwise communications path 46.

In one example operation, west device 48 of node A at 42a receives data from a network element at another one of the nodes B-D at 42b-42d via clockwise communications path 44. West device 48 receives the data synchronously in TDM channels via clockwise communications path 44 and reassembles the original data from each channel of the TDM channels. The reassembled data from a channel is stored in a first-in-first-out memory corresponding to that channel. West device 48 consumes data from a channel if the data is intended for a network element serviced by west device 48. West device 48 transmits the data to other network elements in other nodes, such as nodes B-D at 42b-42d. To transmit the data to other nodes, west device 48 transmits the data in packets asynchronously to east device 50 via communications path 52. East device 50 receives the packets and assembles the data from each channel. The assembled data from a channel is stored in a first-in-first-out memory corresponding to that channel. East device 50 transmits the data synchronously in TDM channels via clockwise communications path 44.

In another example operation, the roles of west device 48 and east device 50 are reversed. East device 50 of node A at 42a receives data from a network element at another one of the nodes B-D at 42b-42d via counter clockwise communications path 46. East device 50 receives the data synchronously in TDM channels via counter clockwise communications path 46 and reassembles the original data from each channel of the TDM channels. The reassembled data from a channel is stored in a first-in-first-out memory corresponding to that channel. East device 50 consumes data from a channel if the data is intended for a network element serviced by the east device 50. East device 50 transmits the data to other network elements in other nodes, such as nodes B-D at 42b-42d. To transmit the data to other nodes, east device 50 transmits the data in packets asynchronously to west device 48 via communications path 52. West device 48 receives the packets and assembles the data from each channel. The assembled data from a channel is stored in a first-in-first-out memory corresponding to that
channel. West device 48 transmits the data synchronously in TDM channels via counter clockwise communications path 46.

Figure 2 is a diagram illustrating one embodiment of a network node 100 according to the present invention. Node 100 is similar to node A at 42a (shown in Figure 1) and part of an RPR network similar to RPR 40 of Figure 1. Node 100 is communicatively coupled to clockwise communications path 102 and counter clockwise communications path 104. Clockwise communications path 102 is similar to clockwise communications path 44 and counter clockwise communications path 104 is similar to counter clockwise communications path 46. In other embodiments, node 100 can be part of any suitable network.

Node 100 includes a west device 106, west network elements 108, an east device 110 and east network elements 112. West device 106 is similar to west device 48 and east device 110 is similar to east device 50. West device 106 is communicatively coupled to clockwise communications path 102 and to counter clockwise communications path 104. Also, west device 106 is communicatively coupled to west network elements 108 via west network elements communications path 114. East device 110 is communicatively coupled to clockwise communications path 102 and to counter clockwise communications path 104. Also, east device 110 is communicatively coupled to east network elements 112 via east network elements communications path 116. In addition, west device 106 and east device 110 are communicatively coupled via device communications path 118 to communicate bi-directionally with each other. Device communications path 118 is similar to communications path 52 (shown in Figure 1).

West device 106 includes a west WAN physical layer 120 and a west reconciliation media access control (RMAC) device 122. West WAN physical layer 120 includes a system packet interface (SPI) 124 that is communicatively coupled to west RMAC device 122 via west device SPI communications path 126. Also, west WAN physical layer 120 is communicatively coupled to clockwise communications path 102 and to counter clockwise communications path 104.

West RMAC device 122 includes a west RMAC SPI 128 that is communicatively coupled to SPI 124 via west device SPI communications path
126. Also, west RMAC device 122 includes a west RMAC mate interface 130 and west RMAC network elements SPI 132. West network elements 108 include west network elements SPI 134 communicatively coupled to west RMAC network elements SPI 132 via west network elements communications path 114. West RMAC mate interface 130 is communicatively coupled to east device 110 via device communications path 118.

West WAN physical layer 120 is similar to physical layer 1 of the Open System Interconnection (OSI) model, which is the standard description or reference model that defines a framework for implementing protocols to communicate messages in a communications system. West WAN physical layer 120 conveys a bit stream through the RPR network at the electrical and mechanical level. The bit stream can be conveyed through electrical impulses, radio signals, light, or any suitable transmission technology. West WAN physical layer 120 provides the hardware for sending and receiving data on a carrier, including cables, cards, and other physical aspects. In one embodiment, west WAN physical layer 120 includes a physical layer for interfacing to a SONET/SDH network including clockwise communications path 102 and counter clockwise communications path 104.

West RMAC device 122 includes at least part of the media access control (MAC) sub-layer of the data link layer 2 of the OSI model. West RMAC device 122 controls sharing west WAN physical layer 120 among several west network elements 108, including how the west network elements 108 gain access to data and permission to transmit data.

East device 110 includes an east WAN physical layer 136 and an east RMAC device 138. East WAN physical layer 136 includes an SPI 140 that is communicatively coupled to east RMAC device 138 via east device SPI communications path 142. Also, east WAN physical layer 136 is communicatively coupled to clockwise communications path 102 and to counter clockwise communications path 104.

East RMAC device 138 includes an east RMAC SPI 144 that is communicatively coupled to SPI 140 via east device SPI communications path 142. Also, east RMAC device 138 includes an east RMAC mate interface 146 and east RMAC network elements SPI 148. East network elements 112 include
east network elements SPI 150 communicatively coupled to east RMAC network
elements SPI 148 via east network elements communications path 116. East
RMAC mate interface 146 is communicatively coupled to west RMAC mate
interface 130 and west device 106 via device communications path 118.

East WAN physical layer 136 is similar to physical layer 1 of the OSI
model. East WAN physical layer 136 conveys a bit stream through the RPR
network at the electrical and mechanical level. The bit stream can be conveyed
through electrical impulses, radio signals, light, or any suitable transmission
technology. East WAN physical layer 136 provides the hardware for sending
and receiving data on a carrier, including cables, cards, and other physical
aspects. In one embodiment, east WAN physical layer 136 includes a physical
layer for interfacing to a SONET/SDH network including clockwise
communications path 102 and counter clockwise communications path 104.

East RMAC device 138 includes at least part of the MAC sub-layer of
the data link layer 2 of the OSI model. East RMAC device 138 controls sharing
east WAN physical layer 136 among several west network elements 108,
including how the west network elements 108 gain access to data and permission
to transmit data.

SPI 124 and west RMAC SPI 128 communicate in an SPI interface that
includes an exchange of data packets between west WAN physical layer 120 and
west RMAC device 122. The SPI interface is an asynchronous, parallel data bit
interface that includes control signals, a transmit clock, and a receive clock, such
as the SPI interface described in Implementation Agreement: OIF-SPI3-01.0,
titled “System Packet Interface Level 3 (SPI-3): OC-48 System Interface for
Physical and Link Layer Devices”, by The Optical Internetworking Forum (OIF)
(2001). In other embodiments, any suitable type of interface can be used
between west WAN physical layer 120 and west RMAC device 122.

West RMAC network elements SPI 132 and west network elements SPI
134 communicate data packets in an SPI interface similar to the interface of SPI
124 and west RMAC SPI 128. Also, SPI 140 and east RMAC SPI 144
communicate data packets in an SPI interface similar to the interface of SPI 124
and west RMAC SPI 128. In addition, east RMAC network elements SPI 148
and east network elements SPI 150 communicate data packets in an SPI interface
similar to the interface of SPI 124 and west RMAC SPI 128. In other embodiments, any suitable type of interface can be used for exchanging data.

West RMAC mate interface 130 is communicatively coupled to east RMAC mate interface 146 via data communications path 152 and status communications path 154. West RMAC mate interface 130 and east RMAC mate interface 146 communicate in asynchronous packet communication and provide full duplex operation. West RMAC mate interface 130 and east RMAC mate interface 146 support data packet transfers between west device 106 and east device 110 via data communications path 152. West RMAC mate interface 130 and east RMAC mate interface 146 support status packet transfers between west device 106 and east device 110 via status communications path 152. In one embodiment, west RMAC mate interface 130 and east RMAC mate interface 146 support data packet transfer rates of 10 Gbps and status packet transfer rates of 1 Gbps. In other embodiments, west RMAC mate interface 130 and east RMAC mate interface 146 support data packet and status packet transfer rates of any suitable bit rate.

In one example operation, west WAN physical layer 120 receives data in channels or timeslots in synchronous TDM communication via clockwise communications path 102. West WAN physical layer 120 assembles the data received for a channel into data packets for the channel. The data packets are transmitted by SPI 124 to west RMAC SPI 128. West RMAC device 122 receives the data packets in asynchronous packet communication via west RMAC SPI 128 from SPI 124. If the data packets are addressed to west network elements 108, west RMAC device 122 transmits the data in the data packets in asynchronous packet communication via west RMAC network elements SPI 132 to west network elements SPI 134 and west network elements 108. West RMAC device 122 also receives data packets for channels in asynchronous packet communication via west RMAC network elements SPI 132 from west network elements SPI 134 and west network elements 108.

West RMAC device 122 segments the received channel data into segments and frames the segments into timeslots in data packets that are transmitted by west RMAC mate interface 130 to east RMAC mate interface 146 via data communications path 152. West RMAC device 122 also frames
segment status information of the segments in the data packets into status packets that are transmitted by west RMAC mate interface 130 to east RMAC mate interface 146 via status communications path 154. In one embodiment, west RMAC device 122 transmits data for a channel in the clockwise communications path 102 at approximately the data rate of the channel in clockwise communications path 102.

East RMAC device 138 receives the data packets and the status packets via east RMAC mate interface 146. East RMAC device 138 assembles the data segments in the data packets into the original data for each of the channels. East RMAC device 138 assembles the data for a channel into data packets for the channel and east RMAC SPI 144 transmits the data packets in asynchronous packet communication to SPI 140. East WAN physical layer 136 assembles the data in the data packets into channels in synchronous TDM communication and transmits the data via clockwise communications path 102. In one embodiment, east RMAC device 138 transmits data for a channel in clockwise communications path 102 at essentially the data rate of the channel in clockwise communications path 102.

In another example operation, the roles of west device 106 and east device 110 are reversed. East WAN physical layer 136 receives data in channels or timeslots in synchronous TDM communication via counter clockwise communications path 104. East WAN physical layer 136 assembles the data received for a channel into data packets for the channel. The data packets are transmitted by SPI 140 to east RMAC SPI 144. East RMAC device 138 receives the data packets in asynchronous packet communication via east RMAC SPI 144 from SPI 140. If the data packets are addressed to east network elements 112, east RMAC device 138 transmits the data in the data packets in asynchronous packet communication via east RMAC network elements SPI 148 to east network elements SPI 150 and east network elements 112. East RMAC device 138 also receives data packets for channels in asynchronous packet communication via east RMAC network elements SPI 148 from east network elements SPI 150 and east network elements 112.

East RMAC device 138 segments the received channel data into segments and frames the segments into timeslots in data packets that are
transmitted by east RMAC mate interface 146 to west RMAC mate interface 130 via data communications path 152. East RMAC device 138 also frames segment status information of the segments in the data packets into status packets that are transmitted by east RMAC mate interface 146 to west RMAC mate interface 130 via status communications path 154. In one embodiment, east RMAC device 138 transmits data for a channel in the counter clockwise communications path 104 at approximately the data rate of the channel in counter clockwise communications path 104.

West RMAC device 122 receives the data packets and the status packets via west RMAC mate interface 130. West RMAC device 122 assembles the data segments in the data packets into the original data for each of the channels. West RMAC device 122 assembles the data for a channel into data packets for the channel and west RMAC SPI 128 transmits the data packets in asynchronous packet communication to SPI 124. West WAN physical layer 120 assembles the data in the data packets into channels in synchronous TDM communication and transmits the data via counter clockwise communications path 104. In one embodiment, west RMAC device 122 transmits data for a channel in counter clockwise communications path 104 at essentially the data rate of the channel in counter clockwise communications path 104.

Figure 3 is a diagram illustrating one embodiment of a west RMAC device 200 and an east RMAC device 202 in a communications system according to the present invention. West RMAC device 200 is similar to west RMAC device 122 (shown in Figure 2) and east RMAC device 202 is similar to east RMAC device 138 (shown in Figure 2). West RMAC device 200 communicates with east RMAC device 202 via data communications path 204 and status communications path 206.

Data communications path 204 includes a west-to-east data communications path 208 and an east-to-west data communications path 210. West-to-east data communications path 208 transfers bits at 10 Gbps and east-to-west data communications path 210 transfers bits at 10 Gbps. In other embodiments, west-to-east data communications path 208 and east-to-west data communications path 210 can transfer bits at any suitable frequency.
Status communications path 206 includes a west-to-east status communications path 212 and an east-to-west status communications path 214. West-to-east status communications path 212 transfers bits at 1 Gbps and east-to-west status communications path 214 transfers bits at 1 Gbps. In other embodiments, west-to-east status communications path 212 and east-to-west status communications path 214 can transfer bits at any suitable frequency.

West RMAC device 200 includes west RMAC mate interface, indicated at 216, transmit (TX) manager 218, receive (RX) manager 220, west RMAC SPI 222, and west RMAC network elements SPI 224. West RMAC SPI 222 is similar to west RMAC SPI 128 (shown in Figure 2) and west RMAC network elements SPI 224 is similar to west RMAC network elements SPI 132 (shown in Figure 2). West RMAC mate interface 216 is similar to west RMAC mate interface 130 (shown in Figure 2). West RMAC mate interface 216 includes a reconciliation sub-layer (RS) 226, a 10 Gbps media independent interface (XGMII) 228, a 10 Gbps attachment unit interface (XAUI) 230, and status interface 232.

West RMAC SPI 222 is coupled with and communicates with TX manager 218 via TX communications path 234 and with RX manager 220 via RX communications path 236. TX manager 218 is coupled with and communicates with RS 226 via data TX path 238 and with status interface 232 via status TX path 240. Also, TX manager 218 is coupled with and communicates with RX manager 220 via status path 242. In addition, TX manager 218 is coupled with and communicates with west RMAC network elements SPI 224 via network elements communications path 243. RX manager 220 is coupled with and communicates with RS 226 via data RX path 244 and with status interface 232 via status RX path 246. Status interface 232 is coupled to west-to-east status communications path 212 and east-to-west status communications path 214. RS 226 is coupled with and communicates with XGMII 228 that is coupled with and communicates with XAUI 230 that is coupled to west-to-east data communications path 208 and east-to-west data communications path 210.

East RMAC device 202 includes east RMAC mate interface, indicated at 248, transmit (TX) manager 250, receive (RX) manager 252, east RMAC SPI
254, and east RMAC network elements SPI 256. East RMAC SPI 254 is similar to east RMAC SPI 144 (shown in Figure 2) and east RMAC network elements SPI 256 is similar to east RMAC network elements SPI 148 (shown in Figure 2). East RMAC mate interface 248 is similar to east RMAC mate interface 146 (shown in Figure 2). East RMAC mate interface 248 includes a reconciliation sub-layer (RS) 258, XGMII 260, XAUI 262, and status interface 264.

East RMAC SPI 254 is coupled with and communicates with TX manager 250 via TX communications path 266 and with RX manager 252 via RX communications path 268. TX manager 250 is coupled with and communicates with RS 258 via data TX path 270 and with status interface 264 via status TX path 272. Also, TX manager 250 is coupled with and communicates with RX manager 252 via status path 274. In addition, TX manager 250 is coupled with and communicates with east RMAC network elements SPI 256 via network elements communications path 275. RX manager 252 is coupled with and communicates with RS 258 via data RX path 276 and with status interface 264 via status RX path 278. Status interface 264 is coupled to west-to-east status communications path 212 and east-to-west status communications path 214. RS 258 is coupled with and communicates with XGMII 260 that is coupled with and communicates with XAUI 262 that is coupled to west-to-east data communications path 208 and east-to-west data communications path 210.

West RMAC SPI 222 is an asynchronous packet interface that receives data in data packets from a physical layer, such as west WAN physical layer 120 (shown in Figure 2). Each of the data packets includes data from a channel in the synchronous TDM network. The data packets are passed to TX manager 218 that assembles the data for each channel into memory locations for that channel. West RMAC SPI 222 also receives data in data packets for channels in the synchronous TDM network from RX manager 220 and transmits the data packets to the physical layer, such as west WAN physical layer 120.

West RMAC network elements SPI 224 is an asynchronous packet interface that receives data in data packets from network elements, such as west network elements 108 (shown in Figure 2). Each of the data packets includes data for a channel in the synchronous TDM network. The data packets are
passed to TX manager 218 that assembles the data for each channel into memory locations for that channel. West RMAC network elements SPI 224 also receives data in data packets for network elements from TX manager 218 and transmits the data packets to the network elements, such as west network elements 108.

TX manager 218 is part of a media access control (MAC) layer in west RMAC device 200. TX manager 218 receives data packets from west RMAC SPI 222 and from west RMAC network elements SPI 224 and assembles the data for each channel into memory locations for that channel. TX manager 218 also regulates the transmission data rate for each channel from west RMAC mate interface 216 to east RMAC mate interface 248. In one embodiment, TX manager 218 transmits data for a channel at approximately the data rate of the channel on the synchronous TDM network.

TX manager 218 divides the data for each channel into data segments and inserts segments from different channels into a segment frame or packet. The segment packet includes timeslots and each timeslot in the segment packet carries a segment from one of the channels. To regulate the transmission data rate for a channel, TX manager 218 inserts and transmits segments from a channel at a suitable data rate.

TX manager 218 compiles segment information about each of the segments in a segment packet. The segment information includes valid data length, segment status, information about the segments destination channel, and the status of the receive buffer of the channel that originated the segment. TX manager 218 inserts the segment information into timeslots in a segment status packet. Each timeslot in the segment status packet carries segment information about one segment in a corresponding segment packet. The sequence of segment information in the segment status packet corresponds to the sequence of segments in the corresponding segment packet.

TX manager 218 transmits the segment packet to RS 226 and the segment status packet to status interface 232. The process of inserting segments into timeslots of a segment packet and inserting segment information into timeslots in a corresponding segment status packet continues for other data. TX manager 218 organizes the segment packets sequentially and presents them to RS 226 for transmission across west-to-east data communications path 208.
Also, TX manager 218 organizes segment status packets sequentially and 
presents them to status interface 232 for transmission across west-to-east status 
communications path 212. The segment packet and corresponding segment 
status packet are transmitted at about the same time to arrive at east RMAC 
device 202 at about the same time.

In addition, TX manager 218 processes acknowledgement frames and 
controls access to RS 226 and status interface 232. TX manager 218 receives the 
status of receive buffers for channels from RX manager 220, which receives the 
receive buffer status of channels from segment status packets transmitted by east 
RMAC device 202. TX manager 218 controls transmitting segments to channels 
based on the status of the destination channel receive buffer.

RX manager 220 is part of the MAC layer in west RMAC device 200. RX 
manager 220 communicates with RS 226 and status interface 232 and 
receives segment packets via RS 226 from east RMAC device 202 and 
corresponding segment status packets via status interface 232 from east RMAC 
device 202. RX manager 220 assembles the data for a channel from the segment 
packets and stores the assembled channel data into memory locations for the 
channel.

RX manager 220 also regulates the transmission data rate of each channel 
from west RMAC SPI 222 to a physical layer, such as west WAN physical layer 
120 (shown in Figure 2). To regulate the transmission data rate for a channel, 
RX manager 220 transmits data in data packets for a channel at a suitable data 
rate. The physical layer receives the data in data packets and transmits the data 
in synchronous TDM communication on the synchronous TDM network. In one 
embodiment, RX manager 220 transmits data for a channel in data packets at 
especially the data rate of the channel on the synchronous TDM network.

RX manager 220 acquires segment information from the segment status 
packet for each segment in the corresponding segment packet. RX manager 220 
uses the valid data length and segment status information to separate data 
segments from the segment packet. RX manager 220 uses the destination 
channel information to assemble the channel data and transmit the data to 
destination channels. RX manager 220 also transmits an acknowledgement 
frame to TX manager 218 that transmits the acknowledgement frame back to
east RMAC device 202. In addition, RX manager 220 acquires the receive
buffer status for the channel that originated the corresponding segment and
transmits the receive buffer status to TX manager 218. If the receive buffer
status is normal, TX manager 218 continues transmitting segments to the
channel. If the receive buffer status is satisfy, TX manager 218 skips one
transmission to the channel to prevent overflowing the receive buffer for that
channel.

RS 226 operates as a command translator between TX manager 218 and
XGMII 228. RS 226 adapts bit serial protocols of TX manager 218 to the
parallel encodings of XGMII 228. Also, RS 226 adapts the parallel encodings of
XGMII 228 to bit serial protocols of RX manager 220.

XGMII 228 provides a standard interconnection that supports 10 Gbps
operations with a 32 bit wide transmit data path and a 32 bit wide receive data
path. XGMII 228 includes 4 transmit control signals and a transmit clock, and 4
receive control signals and a receive clock to provide full duplex operation.
Each direction of data transfer is independent and serviced by the independent
data, control, and clock signals. XGMII 228 passes data and control signals
between RS 226 and XAUI 230. The 32 bit wide transmit data and 4 transmit
control signals are converted to four serial transmit paths in XAUI 230. The 32
bit wide receive data and 4 receive control signals are converted from four serial
receive paths in XAUI 230. The conversions can be done in XGMII 228 and/or
XAUI 230. In one embodiment, a 10 Gbps XGMII extender sub-layer (XGXS)
is inserted between XGMII 228 and XAUI 230 to perform the conversions.

XGMII 228 balances the need for media independence with the need for
a simple and cost effective interface. The bus width and signaling rate are
applicable to short distance integrated circuit chip-to-chip interconnections with
printed circuit board trace lengths electrically limited to about 7 centimeters
(cm). XGMII 228 is described in IEEE Std 802.3ae entitled “Part 3: Carrier
Sense Multiple Access with Collision Detection (CSMA/CD) Access Method
and Physical Layer Specifications, Amendment: Media Access Control (MAC)
Perimeters, Physical Layers, and Management Perimeters for 10 Gb/s
Operation.”
XAUI 230 extends the operational distance of the XGMI2 interface and reduces the number of interface signals. XAUI 230 provides an interconnection between XGMI2 228 and XAUI 262 in east RMAC device 202. XAUI 230 and XAUI 262 extend the operational distance of the interface to 50 cm and can be used as an integrated circuit to integrated circuit interface implemented with traces on a printed circuit board. XAUI 230 transmits data to XAUI 262 via west-to-east data communications path 208 and XAUI 230 receives data from XAUI 262 via east-to-west data communications path 210.

XAUI 230 supports the 10 Gbps data rate of XGMI2 228 through four differential pair transmit paths and four differential pair receive paths. Each of the transmit paths and each of the receive paths is a serial, independent data path that uses low voltage swing differential signaling. Thus, XAUI 230 includes four differential pair transmit paths or eight transmit lines and four differential pair receive paths or eight receive lines.

XGMI2 228 is organized into four transmit lanes with each lane conveying a data octet on each edge of the associated clock, and four receive lanes with each lane conveying a data octet on each edge of the associated clock. Each of the four transmit lanes in XGMI2 228 is transmitted across one of the four differential pair transmit paths in XAUI 230. Also, each of the four receive lanes in XGMI2 228 is transmitted across one of the four differential pair receive paths in XAUI 230. XAUI 230 is further described in IEEE Std 802.3ae, previously referenced herein.

In other embodiments, west RMAC device 200 and east RMAC device 202 do not include XAUI 230 and XAUI 262. Instead, XGMI2 228 is communicatively coupled to XGMI2 260 via data communications path 204. In other embodiments, west RMAC device 200 and east RMAC device 202 include any suitable interface, such as a media independent interface (MII) that operates at less than 10 Gbps or more than 10 Gbps.

Status interface 232 provides an interconnection between TX manager 218 and status interface 264 in east RMAC device 202 and between RX manager 220 and status interface 264 in east RMAC device 202. Status interface 232 and status interface 264 can be used as an integrated circuit to integrated circuit interface implemented with traces on a printed circuit board. Status interface
232 and status interface 264 transmit and receive data in serial bit streams at about 1 Gbps. Status interface 232 transmits data to status interface 264 via west-to-east status communications path 212 and status interface 232 receives data from status interface 264 via east-to-west status communications path 214.

Status interface 232 receives segment status packets from status interface 264 via east-to-west status communications path 214 and passes the received segment status packets to RX manager 220. Status interface 232 receives segment status packets from TX manager 218 and transmits the received segment status packets to status interface 264 via west-to-east status communications path 212. A segment status packet and the corresponding segment packet are transmitted at about the same time to arrive at east RMAC device 202 at about the same time. In one embodiment, each segment status packet can be transmitted slightly ahead of its corresponding segment packet to arrive at east RMAC device 202 slightly ahead of the segment packet.

East RMAC SPI 254 is an asynchronous packet interface that receives data in data packets from a physical layer, such as east WAN physical layer 136 (shown in Figure 2). Each of the data packets includes data from a channel in the synchronous TDM network. The data packets are passed to TX manager 250 that assembles the data for each channel into memory locations for that channel.

East RMAC SPI 254 also receives data in data packets for channels in the synchronous TDM network from RX manager 252 and transmits the data packets to the physical layer, such as east WAN physical layer 136.

East RMAC network elements SPI 256 is an asynchronous packet interface that receives data in data packets from network elements, such as east network elements 112 (shown in Figure 2). Each of the data packets includes data for a channel in the synchronous TDM network. The data packets are passed to TX manager 250 that assembles the data for each channel into memory locations for that channel. East RMAC network elements SPI 256 also receives data in data packets for network elements from TX manager 250 and transmits the data packets to the network elements, such as east network elements 112.

TX manager 250 is part of a media access control (MAC) layer in east RMAC device 202. TX manager 250 receives data packets from east RMAC SPI 254 and from east RMAC network elements SPI 256 and assembles the data
for each channel into memory locations for that channel. TX manager 250 also regulates the transmission data rate for each channel from east RMAC mate interface 248 to west RMAC mate interface 216. In one embodiment, TX manager 250 transmits data for a channel at approximately the data rate of the channel on the synchronous TDM network.

TX manager 250 divides the data for each channel into data segments and inserts segments from different channels into a segment frame or packet. The segment packet includes timeslots and each timeslot in the segment packet carries a segment from one of the channels. To regulate the transmission data rate for a channel, TX manager 250 inserts and transmits segments from a channel at a suitable data rate.

TX manager 250 compiles segment information about each of the segments in a segment packet. The segment information includes valid data length, segment status, information about the segments destination channel, and the status of the receive buffer of the channel that originated the segment. TX manager 250 inserts the segment information into timeslots in a segment status packet. Each timeslot in the segment status packet carries segment information about one segment in a corresponding segment packet. The sequence of segment information in the segment status packet corresponds to the sequence of segments in the corresponding segment packet.

TX manager 250 transmits the segment packet to RS 258 and the segment status packet to status interface 264. The process of inserting segments into timeslots of a segment packet and inserting segment information into timeslots in a corresponding segment status packet continues for other data. TX manager 250 organizes the segment packets sequentially and presents them to RS 258 for transmission across east-to-west data communications path 210. Also, TX manager 250 organizes segment status packets sequentially and presents them to status interface 264 for transmission across east-to-west status communications path 214. The segment packet and corresponding segment status packet are transmitted at about the same time to arrive at west RMAC device 200 at about the same time.

In addition, TX manager 250 processes acknowledgement frames and controls access to RS 258 and status interface 264. TX manager 250 receives the
status of receive buffers for channels via RX manager 252, which receives the receive buffer status of channels from segment status packets transmitted by west RMAC device 200. TX manager 250 controls transmitting segments to channels based on the status of the destination channel receive buffer.

RX manager 252 is part of the MAC layer in east RMAC device 202. RX manager 252 communicates with RS 258 and status interface 264 and receives segment packets via RS 258 from west RMAC device 200 and corresponding segment status packets via status interface 264 from west RMAC device 200. RX manager 252 assembles the data for a channel from the segment packets and stores the assembled channel data into memory locations for the channel.

RX manager 252 also regulates the transmission data rate of each channel from east RMAC SPI 254 to a physical layer, such as east WAN physical layer 136 (shown in Figure 2). To regulate the transmission data rate for a channel, RX manager 252 transmits data in data packets for a channel at a suitable data rate. The physical layer receives the data in data packets and transmits the data in synchronous TDM communication on the synchronous TDM network. In one embodiment, RX manager 252 transmits data for a channel in data packets at essentially the data rate of the channel on the synchronous TDM network.

RX manager 252 acquires segment information from the segment status packet for each segment in the corresponding segment packet. RX manager 252 uses the valid data length and segment status information to separate data segments from the segment packet. RX manager 252 uses the destination channel information to assemble the channel data and transmit the data to destination channels. RX manager 252 also transmits an acknowledgement frame to TX manager 250 that transmits the acknowledgement frame back to west RMAC device 200. In addition, RX manager 252 acquires the receive buffer status for the channel that originated the corresponding segment and transmits the receive buffer status to TX manager 250. If the receive buffer status is normal, TX manager 250 continues transmitting segments to the channel. If the receive buffer status is satisfy, TX manager 250 skips one transmission to the channel to prevent overflowing the receive buffer for that channel.
RS 258 operates as a command translator between TX manager 250 and XGMII 260. RS 258 adapts bit serial protocols of TX manager 250 to the parallel encodings of XGMII 260. Also, RS 258 adapts the parallel encodings of XGMII 260 to bit serial protocols of RX manager 252.

XGMII 260 provides a standard interconnection that supports 10 Gbps operations with a 32 bit wide transmit data path and a 32 bit wide receive data path. XGMII 260 includes 4 transmit control signals and a transmit clock, and 4 receive control signals and a receive clock to provide full duplex operation. Each direction of data transfer is independent and serviced by the independent data, control, and clock signals. XGMII 260 passes data and control signals between RS 258 and XAUI 262. The 32 bit wide transmit data and 4 transmit control signals are converted to four serial transmit paths in XAUI 262. The 32 bit wide receive data and 4 receive control signals are converted from four serial receive paths in XAUI 262. The conversions can be done in XGMII 260 and/or XAUI 262. In one embodiment, a 10 Gbps XGMII extender sub-layer (XGXS) is inserted between XGMII 260 and XAUI 262 to perform the conversions.

XGMII 260 balances the need for media independence with the need for a simple and cost effective interface. The bus width and signaling rate are applicable to short distance integrated circuit chip-to-chip interconnections with printed circuit board trace lengths electrically limited to about 7 centimeters (cm). XGMII 260 is described in IEEE Std 802.3ae, previously referenced herein.

XAUI 262 extends the operational distance of the XGMII interface and reduces the number of interface signals. XAUI 262 provides an interconnection between XGMII 260 and XAUI 230 in west RMAC device 200. XAUI 230 and XAUI 262 extend the operational distance of the interface to 50 cm and can be used as an integrated circuit to integrated circuit interface implemented with traces on a printed circuit board. XAUI 262 transmits data to XAUI 230 via east-to-west data communications path 210 and XAUI 262 receives data from XAUI 230 via west-to-east data communications path 208.

XAUI 262 supports the 10 Gbps data rate of XGMII 260 through four differential pair transmit paths and four differential pair receive paths. Each of the transmit paths and each of the receive paths is a serial, independent data path.
that uses low voltage swing differential signaling. Thus, XAUI 262 includes four differential pair transmit paths or eight transmit lines and four differential pair receive paths or eight receive lines.

XGMII 260 is organized into four transmit lanes with each lane conveying a data octet on each edge of the associated clock, and four receive lanes with each lane conveying a data octet on each edge of the associated clock. Each of the four transmit lanes in XGMII 260 is transmitted across one of the four differential pair transmit paths in XAUI 262. Also, each of the four receive lanes in XGMII 260 is transmitted across one of the four differential pair receive paths in XAUI 262. XAUI 262 is further described in IEEE Std 802.3ae, previously referenced herein.

Status interface 264 provides an interconnection between TX manager 250 and status interface 232 in west RMAC device 200 and between RX manager 252 and status interface 232 in west RMAC device 200. Status interface 232 and status interface 264 can be used as an integrated circuit to integrated circuit interface implemented with traces on a printed circuit board. Status interface 232 and status interface 264 transmit and receive data in serial bit streams at about 1 Gbps. Status interface 264 transmits data to status interface 232 via east-to-west status communications path 214 and status interface 264 receives data from status interface 232 via west-to-east status communications path 212.

Status interface 264 receives segment status packets from status interface 232 via west-to-east status communications path 212 and passes the received segment status packets to RX manager 252. Status interface 264 receives segment status packets from TX manager 250 and transmits the received segment status packets to status interface 232 via east-to-west status communications path 214. A segment status packet and the corresponding segment packet are transmitted at about the same time to arrive at west RMAC device 200 at about the same time. In one embodiment, each segment status packet can be transmitted slightly ahead of its corresponding segment packet to arrive at west RMAC device 200 slightly ahead of the segment packet.

In operation of transmissions from west RMAC device 200 to east RMAC device 202, TX manager 218 receives data packets from west RMAC
SPI 222 and west RMAC network elements SPI 224 and assembles data for each channel into memory locations for that channel. TX manager 218 divides the data for each channel into data segments and checks the receive buffer status for each segments destination channel. TX manager 218 inserts segments from different channels into a segment frame or packet. TX manager 218 only inserts segments with destination channels having a normal receive buffer status.

TX manager 218 compiles segment information about each of the segments in the segment packet and inserts the segment information into timeslots in a segment status packet that corresponds to the segment packet. The sequence of timeslots in the segment status packet corresponds to the sequence of segments in the segment packet. TX manager 218 presents segment packets sequentially to RS 226 for transmission across west-to-east data communications path 208 and TX manager 218 presents segment status packets sequentially to status interface 232 for transmission across west-to-east status communications path 212.

RS 226 maps each of the segment packets received from TX manager 218 to a format compatible with XGMII 228. RS 226 passes each of the segment packets to XGMII 228 that passes each of the segment packets to XAUI 230. XAUI 230 transmits each of the segment packets to XAUI 262 via west-to-east data communications path 208.

Status interface 232 receives each of the segment status packets and transmits each of the segment status packets to status interface 264 via west-to-east status communications path 212. Each segment packet and corresponding segment status packet arrives at east RMAC device 202 at about the same time. The process of inserting segments into timeslots of a segment packet and inserting segment information into timeslots in a corresponding segment status packet continues for other data.

XAUI 262 in east RMAC device 202 receives segment packets from XAUI 230, and status interface 264 in east RMAC device 202 receives the corresponding segment status packets from status interface 232. XAUI 262 passes received segment packets to RX manager 252 through XGMII 260 and RS 258. Status interface 264 passes received segment status packets to RX manager 252.
RX manager 252 divides segment information from the segment status packet and uses the valid data length and segment status to separate segments from the corresponding segment packet. RX manager 252 uses the destination channel information to assemble the data for a channel from the segment packets and store the assembled channel data into memory locations for the channel. RX manager 252 transmits the data to destination channels and an acknowledgement frame to TX manager 250 that transmits the acknowledgement frame back to west RMAC device 200. In addition, RX manager 252 acquires the receive buffer status for the channel that originated the corresponding segment and transmits the receive buffer status to TX manager 250.

RX manager 252 regulates the transmission data rate of each channel from east RMAC SPI 254 to a physical layer, such as east WAN physical layer 136 (shown in Figure 2). To regulate the transmission data rate for a channel, RX manager 252 transmits data in data packets for a channel at a suitable data rate. The physical layer receives the data in data packets and transmits the data in synchronous TDM communication on the synchronous TDM network.

In operation of transmissions from east RMAC device 202 to west RMAC device 200, TX manager 250 receives data packets from east RMAC SPI 254 and east RMAC network elements SPI 256 and assembles data for each channel into memory locations for that channel. TX manager 250 divides the data for each channel into data segments and checks the receive buffer status for each segments destination channel. TX manager 250 inserts segments from different channels into a segment frame or packet. TX manager 250 only inserts segments with destination channels having a normal receive buffer status.

TX manager 250 compiles segment information about each of the segments in the segment packet and inserts the segment information into timeslots in a segment status packet that corresponds to the segment packet. The sequence of timeslots in the segment status packet corresponds to the sequence of segments in the segment packet. TX manager 250 presents segment packets sequentially to RS 258 for transmission across east-to-west data communications path 210 and TX manager 250 presents segment status packets sequentially to status interface 264 for transmission across east-to-west status communications path 214.
RS 258 maps each of the segment packets received from TX manager 250 to a format compatible with XGMII 260. RS 258 passes each of the segment packets to XGMII 260 that passes each of the segment packets to XAUI 262. XAUI 262 transmits each of the segment packets to XAUI 230 via east-to-west data communications path 210.

Status interface 264 receives each of the segment status packets and transmits each of the segment status packets to status interface 232 via east-to-west status communications path 214. Each segment packet and corresponding segment status packet arrives at west RMAC device 200 at about the same time. The process of inserting segments into timeslots of a segment packet and inserting segment information into timeslots in a corresponding segment status packet continues for other data.

XAUI 230 in west RMAC device 200 receives segment packets from XAUI 262, and status interface 232 in west RMAC device 200 receives the corresponding segment status packets from status interface 264. XAUI 230 passes received segment packets to RX manager 220 through XGMII 228 and RS 226. Status interface 232 passes received segment status packets to RX manager 220.

RX manager 220 divides segment information from the segment status packet and uses the valid data length and segment status to separate segments from the corresponding segment packet. RX manager 220 uses the destination channel information to assemble the data for a channel from the segment packets and store the assembled channel data into memory locations for the channel. RX manager 220 transmits the data to destination channels and an acknowledgement frame to TX manager 218 that transmits the acknowledgement frame back to east RMAC device 202. In addition, RX manager 220 acquires the receive buffer status for the channel that originated the corresponding segment and transmits the receive buffer status to TX manager 218.

RX manager 220 regulates the transmission data rate of each channel from west RMAC SPI 222 to a physical layer, such as west WAN physical layer 120 (shown in Figure 2). To regulate the transmission data rate for a channel, RX manager 220 transmits data in data packets for a channel at a suitable data
rate. The physical layer receives the data in data packets and transmits the data in synchronous TDM communication on the synchronous TDM network.

Figure 4 is a diagram illustrating one embodiment of a segment packet 300. Segment packet 300 includes a preamble 302, a status control field 304, a start of frame delimiter (SFD) 306, a payload 308, an end of frame delimiter (EFD) 310, and an inter packet gap (IPG) 312. Segment packet 300 is communicated between west RMAC device 200 and east RMAC device 202 via XAUI 230 and XAUI 262 (shown in Figure 3).

Preamble 302 is a 6 byte field that begins a segment packet transmission. Preamble 302 is initiated by a TX manager, such as TX manager 218 or TX manager 250, and begins as six bytes in the following hexadecimal bit pattern: 0xAA AA AA AA AA AA. A reconciliation sub-layer, such as RS 226 and RS 258, converts the first byte of 0xAA into a one byte start control character that indicates the beginning of a segment packet. During reception, a reconciliation sub-layer converts the start control character back into 0xAA. In one embodiment, preamble 302 is a 7 byte field. In other embodiments, preamble 302 can be any suitable length.

Status control field 304 follows preamble 302 and is used to exchange synchronization messages between devices, such as west RMAC device 200 and east RMAC device 202. Status control field 304 sometimes includes a scan table synchronization byte that indicates the first segment in the first timeslot of payload 308 corresponds to the first channel entry in the indicated scan table. Each scan table includes channel identification entries and each of the paired devices, such as west RMAC device 200 and east RMAC device 202, includes transmit scan tables and receive scan tables. The transmit scan tables in one device correspond to the receive scan tables in the other device, such that corresponding scan tables include channel identification entries in the same sequence.

The transmitting device selects a channel identification entry at a pointer into a transmit scan table and increments the pointer to the next entry. A data segment from the selected channel is transmitted in a segment packet. Also, segment information that includes destination channel information for the data segment is transmitted in the corresponding segment status packet.
The receiving device receives the segment packet and the corresponding segment status packet that includes the destination channel information for the data segment in the segment packet. The destination channel information is a scan table identifier that is passed to the receiving device. A pointer into the identified scan table points to the channel for the data segment and the pointer is incremented. The scan table synchronization byte ensures that pointers into corresponding scan tables in the transmitting and receiving devices are synchronized to ensure that segments are sent to the correct channel. Scan tables and scan table synchronization is described in further detail later herein.

Also, status control field 304 sometimes includes a link capacity adjustment scheme (LCAS) byte that indicates which LCAS to use. In one embodiment, each device includes two LCAS configuration sets, one active and one shadow LCAS. A swap between the active and the shadow LCAS can be done remotely through status control field 304. In one embodiment, status control field 304 is a 1 byte field. In other embodiments, status control field 304 can be any suitable length.

SFD 306 follows status control field 304 and indicates the start of payload 308. In one embodiment, SFD 306 is a 1 byte field. In other embodiments, SFD 306 can be any suitable length. In one embodiment, SFD 306 is a hexadecimal 0xAB. In other embodiments, SFD 306 can be any suitable bit pattern.

Payload 308 follows SFD 306 and includes M+1 timeslots 314a-314n that can include segments from different channels. Each of the timeslots 314a-314n carries a segment from one channel. In one embodiment, M equals 511 and payload 308 includes 512 timeslots. In one embodiment, each of the timeslots 314a-314n is 16 bytes long. In one embodiment, M equals 511 and each of the timeslots 314a-314n is 16 bytes long, such that payload 308 includes 8192 bytes. In other embodiments, M can be any suitable number and each timeslot can be any suitable length, such as 8 bytes or 32 bytes long.

EFD 310 follows payload 308 and indicates the end of payload 308. EFD 310 can be any suitable bit pattern. In one embodiment, EFD 310 is a 1 byte field. In other embodiments, EFD 310 can be any suitable length.
IPG 312 follows EFD 310 and precedes the next segment packet preamble. IPG 312 inserts a delay or time gap between segment packets. This delay provides inter-packet recovery time for OSI layers and the physical medium. In one embodiment, IPG 312 is 11 bytes. In other embodiments, IPG 312 can be any suitable length.

Figure 5 is a diagram illustrating one example of segmenting and framing channel payloads 400, 402, and 404 to 406 into segment packets 408 and 410 in one embodiment of a device, such as west RMAC device 200 or east RMAC device 202 (shown in Figure 3). Each of the channel payloads 400, 402, and 404 to 406 is for a channel in the synchronous TDM network and each of the channel payloads 400, 402, and 404 to 406 is received by the TX manager in the device. In west RMAC device 200, each of the channel payloads 400, 402, and 404 to 406 is received by TX manager 218. In east RMAC device 202, each of the channel payloads 400, 402, and 404 to 406 is received by TX manager 250. In one embodiment, at least two of the channel payloads 400, 402, and 404 to 406 are for different channels. In one embodiment, each of the channel payloads 400, 402, and 404 to 406 is for a different channel.

The TX manager divides channel payloads 400, 402, and 404 to 406 into segments. Channel payload 400 is divided into segments 400a-400n, channel payload 402 is divided into segments 402a-402n, channel payload 404 is divided into segments 404a-404n, and channel payload 406 is divided into segments 406a-406n. Channel payloads between channel payload 404 and channel payload 406 are also divided into segments. In one embodiment, each of the channel payloads 400, 402, and 404 to 406 is divided into segments that are each 16 bytes long. In other embodiments, each of the channel payloads 400, 402, and 404 to 406 is divided into segments that are any suitable length.

Each of the segment packets 408 and 410 is similar to segment packet 300 of Figure 4. Segment packet 408 includes preamble 412, status control field 414, SFD 416, payload 418, EFD 420, and IPG 422. Segment packet 410 includes preamble 424, status control field 426, SFD 428, payload 430, EFD 432, and IPG 434. Each of the components of segment packets 408 and 410 is similar to the corresponding component in segment packet 300 described herein. The TX manager frames segments 400a-400n, 402a-402n, and 404a-404n to
406a-406n into payloads of segment packets, such as payloads 418 and 430 of segment packets 408 and 410.

Payload 418 includes M+1 timeslots 418a-418n and payload 430 includes M+1 timeslots 430a-430n. Each of the timeslots 418a-418n and 430a-430n can carry one of the segments 400a-400n, 402a-402n, and 404a-404n to 406a-406n. In one embodiment, M equals 511 and each of the payloads 418 and 430 includes 512 timeslots. In one embodiment, each of the timeslots 418a-418n and 430a-430n is 16 bytes long. In one embodiment, M equals 511 and each of the timeslots 418a-418n and 430a-430n is 16 bytes long, such that each of the payloads 418 and 430 includes 8192 bytes. In other embodiments, M can be any suitable number and each timeslot can be any suitable length, such as 8 bytes or 32 bytes long.

The TX manager inserts segments 400a-400n, 402a-402n, and 404a-404n to 406a-406n into timeslots such as timeslots 418a-418n and 430a-430n. In one embodiment, the TX manager inserts one of the segments 400a-400n, 402a-402n, and 404a-404n to 406a-406n into one of the timeslots 418a-418n and 430a-430n. In one embodiment, the TX manager can insert more than one of the segments 400a-400n, 402a-402n, and 404a-404n to 406a-406n into one of the timeslots 418a-418n and 430a-430n, where the multiple segments in one timeslot are from one channel.

In one example operation, the TX manager inserts segment 400a into timeslot 0 at 418a, segment 402a into timeslot 1 at 418b, segment 404a into timeslot 2 at 418c, and on, up to segment 406a into timeslot M at 418n. After payload 418 is framed, segment packet 408 can be transmitted to another device.

Next, the TX manager inserts segment 400b into timeslot 0 at 430a, segment 402b into timeslot 1 at 430b, segment 404b into timeslot 2 at 430c, and on, up to segment 406b into timeslot M at 430n. After payload 430 is framed, segment packet 410 can be transmitted to another device. Framing continues until all of the segments 400a-400n, 402a-402n, and 404a-404n to 406a-406n in channel payloads 400, 402, and 404 to 406 have been inserted into segment packets, such as segment packets 408 and 410.

Each of the segment packets 408 and 410 is transmitted to another device and received by an RX manager, such as RX manager 220 or RX manager 252.
The RX manager divides out segment information from the corresponding segment status packet and uses the valid data length and segment status to separate segments 400a-400n, 402a-402n, and 404a-404n to 406a-406n from segment packets, such as segment packets 408 and 410. Segments 400a-400n, 402a-402n, and 404a-404n to 406a-406n are reassembled into channel payloads 400, 402, and 404 to 406 by the RX manager and transmitted to a physical layer and the synchronous TDM network.

Figure 6 is a diagram illustrating one embodiment of a segment status packet 500. The segment status packet 500 includes a frame alignment field 502 and a status payload 504. Each segment status packet, such as segment status packet 500, corresponds to one segment packet, such as segment packet 408 or segment packet 410 (shown in Figure 5). The segment status packet 500 carries segment information about each of the segments in the corresponding segment packet. Segment status packet 500 and the corresponding segment packet are transmitted at about the same time to arrive at the receiving device at about the same time. Segment status packet 500 is communicated between west RMAC device 200 and east RMAC device 202 via status interface 232 and status interface 264 (shown in Figure 3). In one embodiment, segment packets, such as segment packets 408 and 410, are transmitted at 10 Gbps and segment status packets, such as segment status packet 500, are transmitted at 1 Gbps.

Frame alignment field 502 is a 2 byte (16 bit) field that begins a transmission of a segment status packet 500. Frame alignment field 502 is used for synchronization of the segment status packet 500 at the receiving device. The 2 byte hexadecimal bit pattern is 0xFC 03 transmitted from left to right, where the last eight bits are the complement of the first eight bits and mark the start of status payload 504. The hexadecimal sequence 0xFC is repeated continuously where bit stuffing is needed to synchronize segment status packet 500 with its corresponding segment packet and to maintain a designated bit rate, such as 1 Gbps. In other embodiments, frame alignment field 502 can be any suitable length. Also, in other embodiments, frame alignment field 502 can be any suitable bit pattern.

Status payload 504 follows frame alignment field 502 and includes M+1 timeslots 504a-504n. In one embodiment, M equals 511 and status payload 504
includes 512 timeslots. In one embodiment, each of the timeslots 504a-504n is 10 bits long. In another embodiment, each of the timeslots 504a-504n could be greater than 10 bits. In one embodiment, M equals 511 and each of the timeslots 504a-504n is 10 bits long, such that status payload 504 includes 640 bytes (5120 bits). In other embodiments, M can be any suitable number and each timeslot can be any suitable length.

The timeslots 504a-504n correspond one to one with timeslots in the corresponding segment packet. For example, if segment packet 408 (shown in Figure 5) is the corresponding segment packet for segment status packet 500, timeslots 504a-504n correspond one to one with timeslots 418a-418n in segment packet 408. Timeslot 504a carries segment information about the segment in timeslot 418a, timeslot 504b carries segment information about the segment in timeslot 418b, timeslot 504c carries segment information about the segment in timeslot 418c, and so on, up to timeslot 504n that carries segment information about the segment in timeslot 418n. In one embodiment with segment packet 408 being much longer than segment status packet 500, the segment packet 408 is transmitted at 10 Gbps and segment status packet 500 is transmitted at 1 Gbps and each of the timeslots 418a-418n in payload 418 is received at the receiving device at about the same time as each of the corresponding timeslots 504a-504n in status payload 504.

Each of the timeslots 504a-504n includes a receive buffer status field 506, a segment status field 508, an error correction code field 510, a scan table identification field 512, and a valid data length field 514. In one embodiment, each of the timeslots 504a-504n is 10 bits long with a receive buffer status field 506 of 1 bit, segment status field 508 of 2 bits, error correction code field 510 of 1 bit, scan table identification field 512 of 2 bits, and valid data length field 514 of 4 bits.

The receive buffer status field 506 indicates the receive buffer status for the channel that originated the segment in the corresponding segment packet timeslot. The receive buffer status is transmitted from the RX manager to the TX manager and can be either satisfy or normal. If the receive buffer status field 506 indicates that the receive buffer status is satisfy, the TX manager skips a transmission of one or more segments to the channel. If the receive buffer status
field 506 indicates that the receive buffer is normal, the TX manager continues to transmit segments to the channel. In one embodiment, the receive buffer status field is 1 bit. In other embodiments, the receive buffer status field can be any suitable length.

The scan table identification field 512 and the error correction code field 510 indicate the channel of the segment in the segment packet, which corresponds to the timeslot 504a-504n in the status payload 504. The scan table identification field 512 indicates the scan table that contains the channel identification entry for the segment. A pointer into the identified scan table points to the channel that is the destination channel for the segment. Also, the pointer into the identified scan table is incremented. To ensure that corresponding scan tables are kept synchronized, error correction code field 510 provides error correction code data that is used to correct errors in the scan table identification field 512. In one embodiment, the scan table identification field 512 is 2 bits long and the error correction code field is 1 bit long. In other embodiments, the scan table identification field 512 can be any suitable length and the error correction code field 510 can be any suitable length.

The segment status field 508 provides status about the segment in the segment packet, which corresponds to the timeslot 504a-504n in the status payload 504. The segment status field 508 indicates whether the segment is errored, the end of a channel payload, all valid, or all invalid. The segment status field 508 and valid data length field 514 are used to separate valid bytes from invalid bytes and to indicate the beginning of a new channel payload for the channel. In addition, in certain situations, the valid data length field 514 is used for other purposes, such as indicating the status of a transit queue or carrying control commands to the receiving device. In one embodiment, the segment status field 508 is 2 bits. In one embodiment, the valid data length field 514 is 4 bits. In other embodiments, the segment status field 508 can be any suitable length and the valid data length field 514 can be any suitable length.

If the segment status field 508 indicates that the segment is errored, the valid data length field 514 indicates the number of valid bytes in the segment timeslot. The transmitting TX manager inserts a status byte in the segment timeslot after the last valid byte and if the inserted status byte indicates a new
channel payload, the remaining bytes in the corresponding timeslot are valid. If the status byte does not indicate a new channel payload, the remaining bytes are invalid.

If the segment status field 508 indicates the segment is the end of a channel payload, the valid data length field 514 indicates the number of valid bytes. The transmitting TX manager inserts a status byte in the segment timeslot after the last valid byte and if the status byte indicates a new channel payload, the remaining bytes in the segment timeslot are valid. If the status byte does not indicate a new channel payload, the remaining bytes are invalid.

If the segment status field 508 indicates that all bytes in the segment timeslot are valid, the valid data length field 514 can be used for other purposes, such as indicating the status of a transit queue or carrying control commands to the receiving device. Also, if the segment status field 508 indicates that all bytes in the segment timeslot are invalid, the valid data length field 514 can be used for other purposes, such as indicating the status of a transit queue or carrying control commands to the receiving device.

Figure 7 is a diagram illustrating one embodiment of a TX manager 600. TX manager 600 is similar to TX manager 218 (shown in Figure 3) and TX manager 250 (shown in Figure 3). TX manager 600 is part of a MAC layer that receives data packets via asynchronous packet communication and assembles data from a channel into a memory location for that channel. Also, TX manager 600 regulates the data rate for each channel. In one embodiment, TX manager 600 transmits data for a channel at approximately the data rate of the channel on the synchronous TDM network.

TX manager 600 includes a network element queue 602, a transit queue 604, a generated frame queue 606, an idle frame and bandwidth generator 608, and a TX forwarding block 610. The network element queue 602 is communicatively coupled to receive data from network elements via network element communications path 612. Also, network element queue 602 is communicatively coupled to TX forwarding block 610 via queue communications path 614. Transit queue 604 is communicatively coupled to receive data from the physical layer and synchronous TDM network via TX communications path 616. Also, transit queue 604 is communicatively coupled
to TX forwarding block 610 via queue communications path 618. Generated
frame queue 606 is communicatively coupled to TX forwarding block 610 via
queue communications path 620, and idle frame and bandwidth generator 608 is
communicatively coupled to TX forwarding block 610 via queue
communications path 622.

TX manager 600 includes TX scan table 0 at 624, TX scan table 1 at 626,
TX scan table 2 at 628, scan table 0 scheduler 630, scan table 1 scheduler 632,
scan table 2 scheduler 634, a multiplexer 636, and a buffer status processor 638.
Each of the TX scan tables 624, 626, and 628 is communicatively coupled to one
of the scan table schedulers 630, 632, and 634. TX scan table 0 at 624 is
communicatively coupled to scan table 0 scheduler 630, TX scan table 1 at 626
is communicatively coupled to scan table 1 scheduler 632, and TX scan table 2
at 628 is communicatively coupled to scan table 2 scheduler 634. Scan table 0
scheduler 630 is communicatively coupled to multiplexer 636 via scheduler
communications path 640. Scan table 1 scheduler 632 is communicatively
coupled to multiplexer 636 via scheduler communications path 642. Scan table
2 scheduler 634 is communicatively coupled to multiplexer 636 via scheduler
communications path 644. Also, multiplexer 636 is communicatively coupled to
TX forwarding block 610 via multiplexer communications path 646. In
addition, buffer status processor 638 is communicatively coupled to each of the
scan table schedulers 630, 632, and 634 via communications path 648 and to an
RX manager, such as RX manger 220 (shown in Figure 3) and RX manager 252
(shown in Figure 3) via status path 650.

Also, TX manager 600 includes a transmit buffer 652. TX forwarding
block 610 is communicatively coupled to transmit buffer 652 via transmit buffer
communications path 654. Transmit buffer 652 transmits data to a mate
interface, such as mate interface 216 (shown in Figure 3) or mate interface 248
(shown in Figure 3), via TX communications path 656.

Network element queue 602 stores data received in data packets from
network elements via network element communications path 612. TX manager
600 receives the data packets in asynchronous packet communication and stores
the data in network element queue 602. The data is stored in first-in-first-out
(FIFO) memories 658a-658h associated with channel identifications 660a-660h.
Each of the FIFO memories 658a-658h is associated with one of the channel identifications 660a-660h. FIFO memory 658a is associated with channel identification 660a, FIFO memory 658b is associated with channel identification 660b, and so on, up to FIFO memory 658h being associated with channel identification 660h. Each of the FIFO memories 658a-658h stores data for the channel identified by the corresponding one of the channel identifications 660a-660h. Channel identifications 660a-660h identify channels on the synchronous TDM network. In one embodiment, each of the channel identifications 660a-660h is different from other channel identifications 660a-660h.

Transit queue 604 stores data received in data packets from a physical layer via TX communications path 616. The physical layer receives the data via synchronous TDM communication with the synchronous network. TX manager 600 receives the data packets in asynchronous packet communication and stores the data in transit queue 604. The data is stored in first-in-first-out (FIFO) memories 662a-662k associated with channel identifications 664a-664k. Each of the FIFO memories 662a-662k is associated with one of the channel identifications 664a-664k. FIFO memory 662a is associated with channel identification 664a, FIFO memory 662b is associated with channel identification 664b, and so on, up to FIFO memory 662k being associated with channel identification 664k. Each of the FIFO memories 662a-662k stores data for the channel identified by the corresponding one of the channel identifications 664a-664k. Channel identifications 664a-664k identify channels on the synchronous TDM network. In one embodiment, each of the channel identifications 664a-664k is different from other channel identifications 664a-664k.

Generated frame queue 606 stores data generated to control communications on the mate interface and the synchronous TDM network. The generated data includes fairness frames, such as single choke fairness frames, and control frames. The fairness frames are generated to eliminate or reduce congestion on the synchronous TDM network by adjusting channel bandwidth to fit available network bandwidth. Control frames can include acknowledgement frames that acknowledge receipt of data through the mate interface from another device. The generated data is stored in generated frame queue 606 in first-in-first-out (FIFO) memories 666a-666l associated with channel identifications
Each of the FIFO memories 666a-666l is associated with one of the channel identifications 668a-668l. FIFO memory 666a is associated with channel identification 668a, FIFO memory 666b is associated with channel identification 668b, and so on, up to FIFO memory 666l being associated with channel identification 668l. Each of the FIFO memories 666a-666l stores data for the channel identified by the corresponding one of the channel identifications 668a-668l. Channel identifications 668a-668l identify channels on the synchronous TDM network. In one embodiment, each of the channel identifications 664a-664l is different from other channel identifications 664a-664l.

Channel identifications 660a-660h, channel identifications 664a-664k, and channel identifications 668a-668l identify channels on the synchronous TDM network. In one embodiment, the same channel can be identified in channel identifications 660a-660h, channel identifications 664a-664k, and/or channel identifications 668a-668l.

Idle frame and bandwidth generator 608 generates frames and reserves bandwidth on the mate interface and the synchronous TDM network. Idle frames are generated to fill bandwidth on the mate interface and/or the synchronous TDM network. Bandwidth is reserved on the mate interface and the synchronous TDM network for generic framing procedure (GFP) overhead bytes on the synchronous TDM network.

TX forwarding block 610 obtains data from the network element queue 602, transit queue 604, generated frame queue 606, and idle frame and bandwidth generator 608 and provides data segment packets and segment status packets to transmit buffer 652. TX forwarding block 610 receives channel identification signals from multiplexer 636 and in response obtains data for the channel indicated by the received channel identification signals from the network element queue 602, transit queue 604, or generated frame queue 606.

In one embodiment, TX forwarding block 610 obtains the data from the network element queue 602, transit queue 604, or generated frame queue 606 and divides the data for each channel into data segments. TX forwarding block 610 inserts segments from different channels into a segment packet. Also, TX forwarding block 610 compiles segment information about each of the segments.
in the segment packet and inserts the segment information into timeslots in a segment status packet that corresponds to the segment packet. The sequence of timeslots in the segment status packet corresponds to the sequence of segments in the segment packet. TX forwarding block 610 provides segment packets and segment status packets to transmit buffer 652.

TX scan table 0 at 624 is communicatively coupled to scan table 0 scheduler 630 and includes scan table entries 670a-670n and channel identifications 672a-672n. Each of the scan table entries 670a-670n is associated with one of the channel identifications 672a-672n. Scan table entry 670a is associated with channel identification 672a, scan table entry 670b is associated with channel identification 672b, and scan table entry 670n is associated with channel identification 672n. Channel identifications 672a-672n identify channels on the synchronous TDM network. In addition, channel identifications 672a-672n can be no operation entries. A pointer into TX scan table 0 at 624 points to one of the scan table entries 670a-670n and to the corresponding one of the channel identifications 672a-672n.

Scan table 0 scheduler 630 includes a polling mechanism that includes a timer, which periodically indicates the end of a timed period. The polling mechanism operates independently of other polling mechanisms in TX manager 600. At the end of the timed period, scan table 0 scheduler 630 increments the pointer into TX scan table 0 at 624. In one embodiment, the pointer is incremented in a round robin scheme from scan table entry 670a to scan table entry 670b and so on, up to scan table entry 670n, and the pointer is incremented from the last scan table entry 670n to the first scan table entry 670a in TX scan table 0 at 624.

The pointer points to one of the scan table entries 670a-670n and the corresponding one of the channel identifications 672a-672n, which is provided to multiplexer 636 and TX forwarding block 610. The TX forwarding block 610 obtains a data segment for the identified channel and assembles the data segment into a data segment packet that is sent to transmit buffer 652 and transmitted over a mate interface, such as mate interface 216 (shown in Figure 3) or mate interface 248 (shown in Figure 3). TX forwarding block 610 can obtain idle frames or segments if the corresponding one of the channel identifications 672a-
672n is a no operation entry. In one embodiment, the corresponding one of the channel identifications 672a-672n is provided to either the TX forwarding block 610 to transmit data from transmit buffer 652 over a mate interface or to a transmit buffer to transmit data through an SPI interface, such as west RMAC SPI 222 (shown in Figure 3) and east RMAC SPI 254 (shown in Figure 3). In one embodiment, TX scan table 0 at 624 and scan table 0 scheduler 630 operate at the equivalent of the synchronous transport signal 1/virtual concatenation 3 (STS-1/VC-3) data rate.

Buffer status processor 638 obtains receive buffer status signals of normal or satisfy from the RX manager, such as RX manager 220 (shown in Figure 3) and RX manager 252 (shown in Figure 3). Buffer status processor 638 and scan table 0 scheduler 630 flag channels that have a receive buffer status of satisfy and prevent channel identifications 672a-672n of the flagged channels from being sent to multiplexer 636 and TX forwarding block 610.

In one embodiment, TX scan table 0 at 624 and scan table 0 scheduler 630 operate in the 10 Giga bit per second (Gbps) mode. In this mode, n equals 192 and TX scan table 0 at 624 includes 192 scan table entries 670a-670n and 192 corresponding channel identifications 672a-672n. Also, scan table 0 scheduler 630 indicates the end of a timed period every 13.30353 nanoseconds.

A different one of the scan table entries 670a-670n is selected every 13.30353 nanoseconds and each of the scan table entries 670a-670n is selected about once every 2.55 microseconds. If a 16 byte segment for a channel is transmitted every 2.55 microseconds, the data rate for the channel is about 50 mega bits per second (Mbps). Other data rates can be obtained by having multiple entries of the same channel in channel identifications 672a-672n. For example, if TX scan table 0 at 624 includes three entries for one channel in channel identifications 672a-672n, the data rate is about 150 Mbps for the channel.

In one embodiment, TX scan table 0 at 624 and scan table 0 scheduler 630 operate in the 2.5 Gbps mode. In this mode, n equals 48 and TX scan table 0 at 624 includes 48 scan table entries 670a-670n and 48 channel identifications 672a-672n. Scan table 0 scheduler 630 indicates the end of a timed period of 53.21413 nanoseconds. A different one of the scan table entries 670a-670n is selected every 53.21413 nanoseconds and each of the scan table entries 670a-
670n is selected about once every 2.55 microseconds. If a 16 byte segment for a channel is transmitted every 2.55 microseconds, the data rate for the channel is about 50 Mbps. Other data rates can be obtained by having multiple entries of the same channel in channel identifications 672a-672n. For example, if TX scan table 0 at 624 includes three entries for one channel in channel identifications 672a-672n, the data rate is about 150 Mbps for the channel.

TX scan table 1 at 626 is communicatively coupled to scan table 1 scheduler 632 and includes scan table entries 674a-674m and channel identifications 676a-676m. Each of the scan table entries 674a-674m is associated with one of the channel identifications 676a-676m. Scan table entry 674a is associated with channel identification 676a, scan table entry 674b is associated with channel identification 676b, and scan table entry 674m is associated with channel identification 676m. Channel identifications 676a-676m identify channels on the synchronous TDM network. In addition, channel identifications 676a-676m can be no operation entries. A pointer into TX scan table 1 at 626 points to one of the scan table entries 674a-674m and to the corresponding one of the channel identifications 676a-676m.

Scan table 1 scheduler 632 includes a polling mechanism that includes a timer, which periodically indicates the end of a timed period. The polling mechanism operates independently of other polling mechanisms in TX manager 600. At the end of the timed period, scan table 1 scheduler 632 increments the pointer into TX scan table 1 at 626. In one embodiment, the pointer is incremented in a round robin scheme from scan table entry 674a to scan table entry 674b and so on, up to scan table entry 674m, and the pointer is incremented from the last scan table entry 674m to the first scan table entry 674a in TX scan table 1 at 626.

The pointer points to one of the scan table entries 674a-674m and the corresponding one of the channel identifications 676a-676m, which is provided to multiplexer 636 and TX forwarding block 610. The TX forwarding block 610 obtains a data segment for the identified channel and assembles the data segment into a data segment packet that is sent to transmit buffer 652 and transmitted over a mate interface, such as mate interface 216 (shown in Figure 3) or mate interface 248 (shown in Figure 3). TX forwarding block 610 can obtain idle
frames or segments if the corresponding one of the channel identifications 676a-676m is a no operation entry. In one embodiment, the corresponding one of the channel identifications 676a-676m is provided to either the TX forwarding block 610 to transmit data from transmit buffer 652 over a mate interface or to a transmit buffer to transmit data through an SPI interface, such as west RMAC SPI 222 (shown in Figure 3) and east RMAC SPI 254 (shown in Figure 3). In one embodiment, TX scan table 1 at 626 and scan table 1 scheduler 632 operate at the equivalent of the virtual tributary 2/tributary unit 12 (VT2/TU-12) data rate.

Buffer status processor 638 obtains receive buffer status signals of normal or satisfy from the RX manager, such as RX manager 220 (shown in Figure 3) and RX manager 252 (shown in Figure 3). Buffer status processor 638 and scan table 1 scheduler 632 flag channels that have a receive buffer status of satisfy and prevent channel identifications 676a-676m of the flagged channels from being sent to multiplexer 636 and TX forwarding block 610.

In one embodiment, TX scan table 1 at 626 and scan table 1 scheduler 632 operate in the 10 Giga bit per second (Gbps) mode. In this mode, n equals 4032 and TX scan table 1 at 626 includes 4032 scan table entries 674a-674m and 4032 corresponding channel identifications 676a-676m. Also, scan table 1 scheduler 632 indicates the end of a timed period every 13.77865 nanoseconds. A different one of the scan table entries 674a-674m is selected every 13.77865 nanoseconds and each of the scan table entries 674a-674m is selected about once every 55.56 microseconds. If a 16 byte segment for a channel is transmitted every 55.56 microseconds, the data rate for the channel is about 2.304 Mbps.

Other data rates can be obtained by having multiple entries of the same channel in channel identifications 676a-676m. For example, if TX scan table 1 at 626 includes three entries for one channel in channel identifications 676a-676m, the data rate is about 6.91 Mbps for the channel.

In one embodiment, TX scan table 1 at 626 and scan table 1 scheduler 632 operate in the 2.5 Gbps mode. In this mode, n equals 1008 and TX scan table 1 at 626 includes 1008 scan table entries 674a-674m and 1008 channel identifications 676a-676m. Scan table 1 scheduler 632 indicates the end of a timed period of 55.11464 nanoseconds. A different one of the scan table entries
674a-674m is selected every 55.11464 nanoseconds and each of the scan table entries 674a-674m is selected about once every 55.56 microseconds. If a 16 byte segment for a channel is transmitted every 55.56 microseconds, the data rate for the channel is about 2.304 Mbps. Other data rates can be obtained by having multiple entries of the same channel in channel identifications 676a-676m.

TX scan table 2 at 628 is communicatively coupled to scan table 2 scheduler 634 and includes scan table entries 678a-678p and channel identifications 680a-680p. Each of the scan table entries 678a-678p is associated with one of the channel identifications 680a-680p. Scan table entry 678a is associated with channel identification 680a, scan table entry 678b is associated with channel identification 680b, and scan table entry 678p is associated with channel identification 680p. Channel identifications 680a-680p identify channels on the synchronous TDM network. In addition, channel identifications 680a-680p can be no operation entries. A pointer into TX scan table 2 at 628 points to one of the scan table entries 678a-678p and to the corresponding one of the channel identifications 680a-680p.

Scan table 2 scheduler 634 includes a polling mechanism that includes a timer, which periodically indicates the end of a timed period. The polling mechanism operates independently of other polling mechanisms in TX manager 600. At the end of the timed period, scan table 2 scheduler 634 increments the pointer into TX scan table 2 at 628. In one embodiment, the pointer is incremented in a round robin scheme from scan table entry 678a to scan table entry 678b and so on, up to scan table entry 678p, and the pointer is incremented from the last scan table entry 678p to the first scan table entry 678a in TX scan table 2 at 628.

The pointer points to one of the scan table entries 678a-678p and the corresponding one of the channel identifications 680a-680p, which is provided to multiplexer 636 and TX forwarding block 610. The TX forwarding block 610 obtains a data segment for the identified channel and assembles the data segment into a data segment packet that is sent to transmit buffer 652 and transmitted over a mate interface, such as mate interface 216 (shown in Figure 3) or mate interface 248 (shown in Figure 3). TX forwarding block 610 can obtain idle frames or segments if the corresponding one of the channel identifications 680a-
680p is a no operation entry. In one embodiment, the corresponding one of the channel identifications 680a-680p is provided to either the TX forwarding block 610 to transmit data from transmit buffer 652 over a mate interface or to a transmit buffer to transmit data through an SPI interface, such as west RMAC SPI 222 (shown in Figure 3) and east RMAC SPI 254 (shown in Figure 3). In one embodiment, TX scan table 2 at 628 and scan table 2 scheduler 634 operate at the equivalent of the VT1.5/TU-11 data rate.

Buffer status processor 638 obtains receive buffer status signals of normal or satisfy from the RX manager, such as RX manager 220 (shown in Figure 3) and RX manager 252 (shown in Figure 3). Buffer status processor 638 and scan table 2 scheduler 634 flag channels that have a receive buffer status of satisfy and prevent channel identifications 680a-680p of the flagged channels from being sent to multiplexer 636 and TX forwarding block 610.

In one embodiment, TX scan table 2 at 628 and scan table 2 scheduler 634 operate in the 10 Giga bit per second (Gbps) mode. In this mode, n equals 5376 and TX scan table 2 at 628 includes 5376 scan table entries 678a-678p and 5376 corresponding channel identifications 680a-680p. Also, scan table 2 scheduler 634 indicates the end of a timed period every 13.77865 nanoseconds. A different one of the scan table entries 678a-678p is selected every 13.77865 nanoseconds and each of the scan table entries 678a-678p is selected about once every 74.07 microseconds. If a 16 byte segment for a channel is transmitted every 74.07 microseconds, the data rate for the channel is about 1.728 Mbps. Other data rates can be obtained by having multiple entries of the same channel in channel identifications 680a-680p. For example, if TX scan table 2 at 628 includes three entries for one channel in channel identifications 680a-680p, the data rate is about 5.18 Mbps for the channel.

In one embodiment, TX scan table 2 at 628 and scan table 2 scheduler 634 operate in the 2.5 Gbps mode. In this mode, n equals 1344 and TX scan table 2 at 628 includes 1344 scan table entries 678a-678p and 1344 channel identifications 680a-680p. Scan table 2 scheduler 634 indicates the end of a timed period of 55.11464 nanoseconds. A different one of the scan table entries 678a-678p is selected every 55.11464 nanoseconds and each of the scan table entries 678a-678p is selected about once every 74.07 microseconds. If a 16 byte
segment for a channel is transmitted every 74.07 microseconds, the data rate for
the channel is about 1.728 Mbps. Other data rates can be obtained by having
multiple entries of the same channel in channel identifications 680a-680p.

Transmit buffer 652 stores segment status packets 682a-682s and data
segment packets 684a-684s. Each of the segment status packets 682a-682s is
similar to segment status packet 500 of Figure 6 and each of the data segment
packets 684a-684s is similar to segment packet 300 of Figure 4. Also, each of
the segment status packets 682a-682s corresponds to one of the segment packets
684a-684s. Segment status packet 682a corresponds to segment packet 684a,
segment status packet 682b corresponds to segment packet 684b, and on, up to
segment status packet 682s that corresponds to segment packet 684s. Transmit
buffer 652 receives segment status packets 682a-682s and segment packets
684a-684s from TX forwarding block 610 and transmits segment status packets
682a-682s and data segment packets 684a-684s over a mate interface, such as
mate interface 216 (shown in Figure 3) or mate interface 248 (shown in Figure
3). Segment status packets 682a-682s are transmitted to a status interface, such
as status interface 232 (shown in Figure 3) and status interface 264 (shown in
Figure 3). Segment packets 684a-684s are transmitted to an RS, such as RS 226
(shown in Figure 3) and RS 258 (shown in Figure 3).

In operation, TX manager 600 receives data packets in asynchronous
packet communication from network elements and stores the data in network
element queue 602. Also, TX manager 600 receives data packets in
asynchronous packet communication from the physical layer, which receives the
data from the synchronous TDM network, and stores the data in transit queue
604. In addition, generated frame queue 606 stores data generated to control
communications on the mate interface and the synchronous TDM network, and
idle frame and bandwidth generator 608 generates frames and reserves
bandwidth on the mate interface and the synchronous TDM network.

Scan table 0 scheduler 630, scan table 1 scheduler 632, and scan table 2
scheduler 634 operate independently of one another to select channel
identifications from TX scan table 0 at 624, TX scan table 1 at 626, and TX scan
table 2 at 628. Scan table 0 scheduler 630 polls TX scan table 0 at 624 to
increment the pointer into TX scan table 0 at 624 and select one of the scan table
entries 670a-670n and one of the channel identifications 672a-672n. Scan table 1 scheduler 632 polls TX scan table 1 at 626 to increment the pointer into TX scan table 1 at 626 and select one of the scan table entries 674a-674m and one of the channel identifications 676a-676m. Scan table 2 scheduler 634 polls TX scan table 2 at 628 to increment the pointer into TX scan table 2 at 628 and select one of the scan table entries 678a-678p and one of the channel identifications 680a-680p. Each of the scan table schedulers 630, 632, and 634 filter selected channel identifications 672a-672n, 676a-676m, and 680a-680p to remove channel identifications that have been flagged due to a receive buffer status of satisfy. Channel identifications 672a-672n, 676a-676m, and 680a-680p that have not been flagged are provided to multiplexer 636 and TX forwarding block 610.

TX forwarding block 610 selects a data segment from each of the channels identified by the provided channel identifications 672a-672n, 676a-676m, and 680a-680p and assembles the selected data segments into data segment packets that are transmitted to transmit buffer 652. Also, TX forwarding block 610 assembles segment information into segment status packets that are transmitted to transmit buffer 652. The transmit buffer 652 transmits each of the segment status packets 682a-682s and corresponding segment packets 684a-684s to the mate interface and another RMAC device.

In one embodiment, scan table 0 scheduler 630, scan table 1 scheduler 632, and scan table 2 scheduler 634 select channel identifications 672a-672n, 676a-676m, and 680a-680p that identify channels and no operation entries to select each of the channels on the synchronous TDM network at a data rate that corresponds to the data rate of each of the identified channels on the synchronous TDM network. Channel identifications 672a-672n, 676a-676m, and 680a-680p can identify a channel once or more than once to achieve the data rate of the channel on the synchronous TDM network. In one embodiment, scan table 0 scheduler 630, scan table 1 scheduler 632, and scan table 2 scheduler 634 select channel identifications 672a-672n, 676a-676m, and 680a-680p that identify channels on the synchronous TDM network and no operation entries at a combined data rate that corresponds to the synchronous TDM network data rate, such as 2.5 Gbps or 10 Gbps.
Figure 8 is a diagram illustrating one embodiment of an RX manager 700. RX manager 700 is similar to RX manager 220 (shown in Figure 3) and RX manager 252 (shown in Figure 3). RX manager 700 is part of a MAC layer that receives data packets via asynchronous packet communication in a mate interface, such as mate interface 216 (shown in Figure 3) and mate interface 248 (shown in Figure 3). RX manager 700 assembles data for a channel and stores the data in a memory location for the channel. Also, RX manager 700 transmits the data to an SPI interface, such as west RMAC SPI 222 (shown in Figure 3) and east RMAC SPI 254 (shown in Figure 3). In addition, RX manager 700 regulates the data rate for each channel on the synchronous TDM network. In one embodiment, RX manager 700 transmits data for a channel at essentially the data rate of the channel on the synchronous TDM network.

RX manager 700 includes a packet receive buffer 702, an RX forwarding block 704, RX scan table 0 at 706, RX scan table 1 at 708, and RX scan table 2 at 710. The packet receive buffer 702 is communicatively coupled to receive data from the mate interface via RX communications path 712. Also, packet receive buffer 702 is communicatively coupled to RX forwarding block 704 via buffer communications path 714. The RX forwarding block 704 is communicatively coupled to RX scan table 0 at 706 via pointer communications path 716, and to RX scan table 1 at 708 via pointer communications path 718, and to RX scan table 2 at 710 via pointer communications path 720.

Also, RX manager 700 includes a channel data receive buffer 722 and a transmit block 724. RX forwarding block 704 is communicatively coupled to channel data receive buffer 722 via forwarding block communications path 726 and channel data receive buffer 722 is communicatively coupled to transmit block 724 via receive buffer communications path 728. Transmit block 724 transmits the data to an SPI interface via RX communications path 730. The SPI interface transmits the data to a physical layer, which transmits the data to the synchronous TDM network.

Packet receive buffer 702 receives segment status packets 732a-732s and data segment packets 734a-734s via RX communications path 712 from the mate interface. Each of the segment status packets 732a-732s is similar to segment status packet 500 of Figure 6 and each of the data segment packets 734a-734s is
similar to segment packet 300 of Figure 4. Also, each of the segment status packets 732a-732s corresponds to one of the segment packets 734a-734s. Segment status packet 732a corresponds to segment packet 734a, segment status packet 732b corresponds to segment packet 734b, and on, up to segment status packet 732s that corresponds to segment packet 734s. Packet receive buffer 702 transmits segment status packets 732a-732s and segment packets 734a-734s to RX forwarding block 704.

RX forwarding block 704 obtains segment status packets 732a-732s and segment packets 734a-734s from packet receive buffer 702. RX forwarding block 704 reassembles data for a channel from segment packets 734a-734s and provides the reassembled data to channel data receive buffer 722. The reassembled data for a channel is stored in a memory area for the channel in channel data receive buffer 722.

To reassemble data for a channel from segment packets 734a-734s, RX forwarding block 704 obtains the scan table identification from each scan table identification field, such as scan table identification field 512 (shown in Figure 6), in segment status packets 732a-732s. The scan table identification for a segment indicates the receive scan table that includes the channel identification for the segment. RX forwarding block 704 obtains the channel identification for the segment from the indicated scan table and reassembles the data into channel data. The reassembled channel data is provided to channel data receive buffer 722.

RX scan table 0 at 706 includes scan table entries 736a-736n and channel identifications 738a-738n. Each of the scan table entries 736a-736n is associated with one of the channel identifications 738a-738n. Scan table entry 736a is associated with channel identification 738a, scan table entry 736b is associated with channel identification 738b, and scan table entry 736n is associated with channel identification 738n. Channel identifications 738a-738n identify channels for data on the synchronous TDM network. In addition, channel identifications 738a-738n can be no operation entries. A pointer into RX scan table 0 at 706 points to one of the scan table entries 736a-736n and to the corresponding one of the channel identifications 738a-738n.
RX forwarding block 704 obtains the scan table identification for a data segment from one of the segment status packets 732a-732s. If the scan table identification indicates scan table 0 at 706, RX forwarding block 704 increments the pointer into RX scan table 0 at 706 and retrieves the one of the channel identifications 738a-738n pointed to by the pointer. RX forwarding block 704 provides the data segment to channel data receive buffer 722 and the data segment is stored in the memory for the channel identified by the retrieved one of the channel identifications 738a-738n. Nothing is sent to channel data receive buffer 722 if the retrieved one of the channel identifications 738a-738n is a no operation entry. In one embodiment, the pointer is incremented in a round robin scheme from scan table entry 736a to scan table entry 736b and so on, up to scan table entry 736n, and the pointer is incremented from the last scan table entry 736n to the first scan table entry 736a.

In one embodiment, RX scan table 0 at 706 corresponds to TX scan table 0 at 624 (shown in Figure 7). The channels identified in channel identifications 738a-738n are in the same sequence in channel identifications 738a-738n and channel identifications 672a-672n (shown in Figure 7). Channel identifications 672a-672n can also include channel identities that are not in channel identifications 738a-738n. TX forwarding block 610 (shown in Figure 7) assembles data segments for channels identified by channel identifications 672a-672n into segment packets. For data segments having channel identifications in TX scan table 0 at 624 and RX scan table 0 at 706, TX forwarding block 610 includes a scan table 0 identification in the scan table identification field of the segment status packet. RX forwarding block 704 receives the segment packets and segment status packets and for each of the scan table 0 identifications, RX forwarding block 704 increments the pointer into RX scan table 0 at 706 and retrieves the one of the channel identifications 738a-738n pointed to by the pointer. The data segment corresponding to the scan table 0 identification is stored in the memory area in channel data receive buffer 722 for the identified channel.

To synchronize the pointer into RX scan table 0 at 706 with the pointer into TX scan table 0 at 624, TX forwarding block 610 includes a scan table synchronization byte that indicates RX scan table 0 in the status control field of
the segment packet, such as status control field 304 in segment packet 300 of Figure 4. The RX scan table 0 synchronization byte indicates that the first data segment in the first timeslot of the payload in the segment packet corresponds to the first channel identification entry 738a in RX scan table 0 at 706. This first channel identification entry 738a is the first channel identification entry in the sequence of channel identification entries in channel identifications 738a-738n that corresponds to the same sequence of channel identification entries in channel identifications 672a-672n. RX forwarding block 704 receives each of the data segment packets 734a-734s and obtains the scan table synchronization bytes for RX scan table 0 at 706 to synchronize the pointer into RX scan table 0 at 706 with the pointer into TX scan table 0 at 624.

In one embodiment, RX scan table 0 at 706 and TX scan table 0 at 624 operate at the equivalent of the STS-1/VC-3 data rate. In one embodiment, RX scan table 0 at 706 and TX scan table 0 at 624 operate in the 10 Giga bit per second (Gbps) mode previously described. In one embodiment, RX scan table 0 at 706 and TX scan table 0 at 624 operate in the 2.5 Gbps mode previously described.

RX scan table 1 at 708 includes scan table entries 740a-740m and channel identifications 742a-742m. Each of the scan table entries 740a-740m is associated with one of the channel identifications 742a-742m. Scan table entry 740a is associated with channel identification 742a, scan table entry 740b is associated with channel identification 742b, and scan table entry 740m is associated with channel identification 742m. Channel identifications 742a-742m identify channels for data on the synchronous TDM network. In addition, channel identifications 742a-742m can be no operation entries. A pointer into RX scan table 1 at 708 points to one of the scan table entries 740a-740m and to the corresponding one of the channel identifications 742a-742m.

RX forwarding block 704 obtains the scan table identification for a data segment from one of the segment status packets 732a-732s. If the scan table identification indicates scan table 1 at 708, RX forwarding block 704 increments the pointer into RX scan table 1 at 708 and retrieves the one of the channel identifications 742a-742m pointed to by the pointer. RX forwarding block 704 provides the data segment to channel data receive buffer 722 and the data.
segment is stored in the memory for the channel identified by the retrieved one of the channel identifications 742a-742m. Nothing is sent to channel data receive buffer 722 if the retrieved one of the channel identifications 742a-742m is a no operation entry. In one embodiment, the pointer is incremented in a round robin scheme from scan table entry 740a to scan table entry 740b and so on, up to scan table entry 740m, and the pointer is incremented from the last scan table entry 740m to the first scan table entry 740a.

In one embodiment, RX scan table 1 at 708 corresponds to TX scan table 1 at 626 (shown in Figure 7). The channels identified in channel identifications 742a-742m are in the same sequence in channel identifications 742a-742m and channel identifications 676a-676m (shown in Figure 7). Channel identifications 676a-676m can also include channel identities that are not in channel identifications 742a-742m. TX forwarding block 610 (shown in Figure 7) assembles data segments for channels identified by channel identifications 676a-676m into segment packets. For data segments having channel identifications in TX scan table 1 at 626 and RX scan table 1 at 708, TX forwarding block 610 includes a scan table 1 identification in the scan table identification field of the segment status packet. RX forwarding block 704 receives the segment packets and segment status packets and for each of the scan table 1 identifications, RX forwarding block 704 increments the pointer into RX scan table 1 at 708 and retrieves the one of the channel identifications 742a-742m pointed to by the pointer. The data segment corresponding to the scan table 1 identification is stored in the memory area in channel data receive buffer 722 for the identified channel.

To synchronize the pointer into RX scan table 1 at 708 with the pointer into TX scan table 1 at 626, TX forwarding block 610 includes a scan table synchronization byte that indicates RX scan table 1 in the status control field of the segment packet, such as status control field 304 in segment packet 300 of Figure 4. The RX scan table 1 synchronization byte indicates that the first data segment in the first timeslot of the payload in the segment packet corresponds to the first channel identification entry 742a in RX scan table 1 at 708. This first channel identification entry 742a is the first channel identification entry in the sequence of channel identification entries in channel identifications 742a-742m.
that corresponds to the same sequence of channel identification entries in
channel identifications 676a-676m. RX forwarding block 704 receives each of
the data segment packets 734a-734s and obtains the scan table synchronization
bytes for RX scan table 1 at 708 to synchronize the pointer into RX scan table 1
at 708 with the pointer into TX scan table 1 at 626.

In one embodiment, RX scan table 1 at 708 and TX scan table 1 at 626
operate at the equivalent of the STS-1/VC-3 data rate. In one embodiment, RX
scan table 1 at 708 and TX scan table 1 at 626 operate in the 10 Giga bit per
second (Gbps) mode previously described. In one embodiment, RX scan table 1
at 708 and TX scan table 1 at 626 operate in the 2.5 Gbps mode previously
described.

RX scan table 2 at 710 includes scan table entries 744a-744p and channel
identifications 746a-746p. Each of the scan table entries 744a-744p is associated
with one of the channel identifications 746a-746p. Scan table entry 744a is
associated with channel identification 746a, scan table entry 744b is associated
with channel identification 746b, and scan table entry 744p is associated with
channel identification 746p. Channel identifications 746a-746p identify
channels for data on the synchronous TDM network. In addition, channel
identifications 746a-746p can be no operation entries. A pointer into RX scan
table 2 at 710 points to one of the scan table entries 744a-744p and to the
the corresponding one of the channel identifications 746a-746p.

RX forwarding block 704 obtains the scan table identification for a data
segment from one of the segment status packets 732a-732s. If the scan table
identification indicates scan table 2 at 710, RX forwarding block 704 increments
the pointer into RX scan table 2 at 710 and retrieves the one of the channel
identifications 746a-746p pointed to by the pointer. RX forwarding block 704
provides the data segment to channel data receive buffer 722 and the data
segment is stored in the memory for the channel identified by the retrieved one
of the channel identifications 746a-746p. Nothing is sent to channel data receive
buffer 722 if the retrieved one of the channel identifications 746a-746p is a no
operation entry. In one embodiment, the pointer is incremented in a round robin
scheme from scan table entry 744a to scan table entry 744b and so on, up to scan
table entry 744p, and the pointer is incremented from the last scan table entry 744p to the first scan table entry 744a.

In one embodiment, RX scan table 2 at 710 corresponds to TX scan table 2 at 628 (shown in Figure 7). The channels identified in channel identifications 746a-746p are in the same sequence in channel identifications 746a-746p and channel identifications 680a-680p (shown in Figure 7). Channel identifications 680a-680p can also include channel identities that are not in channel identifications 746a-746p. TX forwarding block 610 (shown in Figure 7) assembles data segments for channels identified by channel identifications 680a-680p into segment packets. For data segments having channel identifications in TX scan table 2 at 628 and RX scan table 2 at 710, TX forwarding block 610 includes a scan table 2 identification in the scan table identification field of the segment status packet. RX forwarding block 704 receives the segment packets and segment status packets and for each of the scan table 2 identifications, RX forwarding block 704 increments the pointer into RX scan table 2 at 710 and retrieves the one of the channel identifications 746a-746p pointed to by the pointer. The data segment corresponding to the scan table 2 identification is stored in the memory area in channel data receive buffer 722 for the identified channel.

To synchronize the pointer into RX scan table 2 at 710 with the pointer into TX scan table 2 at 628, TX forwarding block 610 includes a scan table synchronization byte that indicates RX scan table 2 in the status control field of the segment packet, such as status control field 304 in segment packet 300 of Figure 4. The RX scan table 2 synchronization byte indicates that the first data segment in the first timeslot of the payload in the segment packet corresponds to the first channel identification entry 746a in RX scan table 2 at 710. This first channel identification entry 746a is the first channel identification entry in the sequence of channel identification entries in channel identifications 746a-746p that corresponds to the same sequence of channel identification entries in channel identifications 680a-680p. RX forwarding block 704 receives each of the data segment packets 734a-734s and obtains the scan table synchronization bytes for RX scan table 2 at 710 to synchronize the pointer into RX scan table 2 at 710 with the pointer into TX scan table 2 at 628.
In one embodiment, RX scan table 2 at 710 and TX scan table 2 at 628 operate at the equivalent of the STS-1/VC-3 data rate. In one embodiment, RX scan table 2 at 710 and TX scan table 2 at 628 operate in the 10 Giga bit per second (Gbps) mode previously described. In one embodiment, RX scan table 2 at 710 and TX scan table 2 at 628 operate in the 2.5 Gbps mode previously described.

Channel data receive buffer 722 includes FIFO memories 748a-748s associated with channel identifications 750a-750s. Each of the FIFO memories 748a-748s is associated with one of the channel identifications 750a-750s. FIFO memory 748a is associated with channel identification 750a, FIFO memory 748b is associated with channel identification 750b, FIFO memory 748c is associated with channel identification 750c, and so on, up to FIFO memory 748s being associated with channel identification 750s. Each of the FIFO memories 748a-748s stores data for the channel identified by the corresponding one of the channel identifications 750a-750s. Channel identifications 750a-750s identify channels on the synchronous TDM network. Channel data receive buffer 722 provides the data in channel identifications 750a-750s to transmit block 724.

Transmit block 724 obtains data from channel data receive buffer 722 and transmits the data to a SPI, such as west RMAC SPI 222 (shown in Figure 3) and east RMAC SPI 254 (shown in Figure 3). The SPI provides the data to a physical layer that transmits the data on the synchronous TDM network. In one embodiment, transmit block 724 transmits data from a channel in channel identifications 750a-750s to the SPI and synchronous TDM network at the data rate of the identified channel on the synchronous TDM network.

In operation, packet receive buffer 702 receives segment status packets 732a-732s and data segment packets 734a-734s via RX communications path 712 from a mate interface. RX forwarding block 704 receives the segment status packets 732a-732s and data segment packets 734a-734s from packet receive buffer 702 and reassembles the data for a channel from segment packets 734a-734s. To reassemble the data for a channel, RX forwarding block 704 obtains each of the scan table identifications for each of the segments in one of the segment packets 734a-734s from the corresponding one of the segment status packets 732a-732s. TX forwarding block 704 increments the pointer into one of
the RX scan tables 706, 708, or 710 based on the scan table identification for a
data segment and obtains the channel identification for the data segment. RX
forwarding block 704 forwards the data segment to the channel receive buffer
722 and stores the data segment in the one of the FIFO’s 748a-748s that
corresponds to the one of the channel identifications 750a-750s for the identified
channel. Transmit block 724 obtains data from channel data receive buffer 722
and transmits the data to an SPI and the synchronous TDM network.

Figures 9A and 9B are diagrams illustrating pointer operation in a TX
scan table 760 and a corresponding RX scan table 762. TX scan table 760
includes sequence numbers 764a-764n and channel identifications 766a-766n.
RX scan table 762 includes sequence numbers 768a-768m and channel
identifications 770a-770m.

To begin, a pointer 772 in TX scan table 760 points to channel
identification 766a and channel 0 is transmitted to a TX forwarding block, such
as TX forwarding block 610 (shown in Figure 7). The TX forwarding block
inserts a data segment for channel 0 into the first timeslot of a segment packet.
Also, channel 0 is the first channel identification in the sequence of channel
identifications 770a-770n in RX scan table 762 and TX forwarding block 610
includes a scan table identification indicating RX scan table 762 in the status
control byte of the segment packet. The scan table identification indicating RX
scan table 762 synchronizes a pointer 774 into RX scan table 762 with pointer
772. An RX forwarding block, such as RX forwarding block 704, sets pointer
772 to point at channel identification 770a.

Next, pointer 772 points to channel identification 766b and channel 4 is
transmitted to the TX forwarding block, which inserts a data segment for channel
4 into the segment packet. Also, channel 4 is the second channel identification
in the sequence of channel identifications 770a-770n and TX forwarding block
inserts a scan table identification indicating RX scan table 762 in the scan table
identification field for the channel 4 data segment of the segment status packet.
The RX forwarding block obtains the scan table identification indicating RX
scan table 762 in the scan table identification field and increments pointer 774 to
channel identification 770b to identify channel 4 as the channel for the channel 4
data segment.
Next, pointer 772 points to channel identification 766c and channel 9 is transmitted to the TX forwarding block. In this example, data for channel 9 is transmitted through the SPI and not the mate interface. Channel 9 is not part of the sequence of channel identifications 770a-770n and pointer 774 is not incremented.

Next, pointer 772 points to channel identification 766d and channel 7 is transmitted to the TX forwarding block, which inserts a data segment for channel 7 into a segment packet. Also, channel 7 is the third channel identification in the sequence of channel identifications 770a-770n and TX forwarding block inserts a scan table identification indicating RX scan table 762 in the scan table identification field for the channel 7 data segment of the segment status packet. The RX forwarding block obtains the scan table identification indicating RX scan table 762 in the scan table identification field and increments pointer 774 to channel identification 770c to identify channel 7 as the channel for the channel 7 data segment.

Next, pointer 772 points to channel identification 766e and channel 25 is transmitted to the TX forwarding block. In this example, data for channel 25 is transmitted through the SPI and not the mate interface. Channel 25 is not part of the sequence of channel identifications 770a-770n and pointer 774 is not incremented.

Next, pointer 772 points to channel identification 766f and channel 17 is transmitted to the TX forwarding block, which inserts a data segment for channel 17 into a segment packet. Also, channel 17 is the fourth channel identification in the sequence of channel identifications 770a-770n and TX forwarding block inserts a scan table identification indicating RX scan table 762 in the scan table identification field for the channel 17 data segment of the segment status packet. The RX forwarding block obtains the scan table identification indicating RX scan table 762 in the scan table identification field and increments pointer 774 to channel identification 770d to identify channel 17 as the channel for the channel 17 data segment.

Next, pointer 772 points to channel identification 766g and channel 3 is transmitted to the TX forwarding block. In this example, data for channel 3 is transmitted through the SPI and not the mate interface. Channel 3 is not part of
the sequence of channel identifications 770a-770n and pointer 774 is not incremented.

This process continues with pointer 772 incrementing from one of the channel identifications 766a-766n to the next and pointer 774 incrementing from one of the channel identifications 770a-770n to the next. If pointer 772 points to channel identification 766n, pointer 772 increments to channel identification 766a in a round robin sequence. Also, if pointer 774 points to channel identification 770n, pointer 774 increments to channel identification 770a in a round robin sequence.

Figures 10A and 10B are diagrams illustrating transmit add/drop register 780 and receive add/drop register 782. TX manager, such as TX manager 600 of Figure 7, includes transmit add/drop register 780 to add or drop scan table channel identification entries in a transmit scan table. RX manager, such as RX manager 700 of Figure 8, includes receive add/drop register 782 to add or drop scan table channel identification entries in a receive scan table. Transmit add/drop register 780 and receive add/drop register 782 are written to add/drop scan table channel identification entries in corresponding transmit and receive scan tables.

Transmit add/drop register 780 includes channel ID 780a, scan table entry number 780b, scan table identification 780c, interface identification 780d, and add/drop decision 780e. Channel ID 780a indicates the identity of the channel to be added or dropped from the identified scan table. Scan table entry number 780b indicates the identity of the scan table entry or sequence number to be changed. Scan table identification 780c indicates the identity of the scan table. Interface identification 780d indicates whether the new scan table entry is transmitted over the SPI or mate interface, and add/drop decision 780e indicates whether the channel ID is added or dropped.

Receive add/drop register 782 includes channel ID 782a, scan table entry number 782b, scan table identification 782c, and add/drop decision 782d. Channel ID 782a indicates the identity of the channel to be added or dropped from the identified scan table. Scan table entry number 782b indicates the identity of the scan table entry or sequence number to be changed. Scan table
identification 782c indicates the identity of the scan table, and add/drop decision
782d indicates whether the channel ID is added or dropped.

In operation, transmit add/drop register 780 and receive add/drop register
782 are written to add/drop a channel identification into corresponding transmit
and receive scan tables. An add/drop command is transmitted to the TX
manager that determines the right time to add/drop the channel identification
indicated in transmit add/drop register 780. After completing the add/drop
command on the transmit side, the TX manager clears transmit add/drop register
780 and transmits an add/drop command to the receive side via the mate
interface. Also, the TX manager generates an interrupt to indicate that the
channel identification has been added or dropped and an add/drop command was
sent to the receive side.

The RX manager receives the add/drop command and completes the
add/drop command on the receive side. After completing the add/drop
command on the receive side, the RX manager clears receive add/drop register
782 and generates an interrupt to indicate that the add/drop command was
received and the channel identification has been added or dropped.

Although specific embodiments have been illustrated and described
herein, it will be appreciated by those of ordinary skill in the art that a variety of
alternate and/or equivalent implementations may be substituted for the specific
embodiments shown and described without departing from the scope of the
present invention. This application is intended to cover any adaptations or
variations of the specific embodiments discussed herein. Therefore, it is
intended that this invention be limited only by the claims and the equivalents
thereof.
WHAT IS CLAIMED IS:

1. A communications system, comprising:
   a first device including a first scan table; and
   a second device including a second scan table, wherein the first device is configured to select the first scan table and transmit scan table identification and the second device is configured to receive the scan table identification from the first device and select the second scan table based on the received scan table identification.

2. The communications system of claim 1, wherein the second device is configured to increment a pointer into the selected second scan table and to select a channel identification entry from the selected second scan table at the pointer based on the received scan table identification.

3. The communications system of claim 1, wherein the first scan table includes first channel identification entries and the second scan table includes second channel identification entries and the first device is configured to select one of the first channel identification entries and transmit data and status based on the selected first channel identification entry and the second device is configured to receive the data and status from the first device and to select one of the second channel identification entries based on the received scan table identification.

4. The communications system of claim 3, wherein the selected first channel identification entry indicates a channel for the received data and the selected second channel identification entry indicates the same channel for the received data.

5. The communications system of claim 3, wherein the second channel identification entries are in a first sequence in the selected second scan table and the first channel identification entries in the selected first scan table include first channel identification entries that correspond to the second channel identification
entries and the first channel identification entries that correspond to the second channel identification entries are in the first sequence in the selected first scan table.

6. The communications system of claim 1, wherein the second device is configured to receive a status control value from the first device and to synchronize a first pointer into the selected second scan table with a second pointer into the selected first scan table based on the status control value.

7. The communications system of claim 1, wherein the second device is configured to receive a status control value that indicates the selected second scan table to synchronize a first pointer into the selected second scan table with a second pointer into the selected first scan table.

8. The communications system of claim 1, wherein the first device is configured to add a first channel identification entry to the first scan table and transmit a first add/drop command to the second device and the second device is configured to add a second channel identification entry to the second scan table in response to the first add/drop command.

9. The communications system of claim 8, wherein the first device includes a first register that is programmed to add the first channel identification entry and the second device includes a second register that is programmed to add the second channel identification entry.

10. The communications system of claim 8, wherein the first device is configured to receive a second add/drop command and to add the first channel identification entry and transmit the first add/drop command to the second device in response to the second add/drop command.

11. The communications system of claim 1, wherein the first device is configured to drop a first channel identification entry from the first scan table and transmit a first add/drop command to the second device and the second
device is configured to drop a second channel identification entry from the second scan table in response to the first add/drop command.

12. The communications system of claim 11, wherein the first device includes a first register that is programmed to drop the first channel identification entry and the second device includes a second register that is programmed to drop the second channel identification entry.

13. The communications system of claim 11, wherein the first device is configured to receive a second add/drop command and to drop the first channel identification entry and transmit the first add/drop command to the second device in response to the second add/drop command.

14. The communications system of claim 1, wherein the first device includes a timer that periodically indicates the expiration of a timed period and the first device is configured to select the first scan table and a first channel identification entry and to transmit data and status based on the selected first channel identification entry in response to the indication from the timer.

15. A communications system, comprising:
   a first system including transmit tables; and
   a second system including receive tables, wherein the first system is configured to select one of the transmit tables and transmit data and status based on the selected transmit table and the second system is configured to receive the data and status and select one of the receive tables based on the received data and status.

16. The communications system of claim 15, wherein the first system comprises timers and each of the timers is associated with one of the transmit tables and configured to periodically initiate a transmit table entry selection from the associated transmit table in response to the expiration of a timed period.
17. The communications system of claim 16, wherein each of the timers is configured to have a different timed period and to initiate the transmit table entry selection from the associated transmit table independently of the other timers.

18. The communications system of claim 15, wherein at least one of the transmit tables is configured to include multiple channel identification entries identifying one channel.

19. The communications system of claim 15, wherein one of the transmit tables is configured to include a first channel identification entry that identifies one channel and another one of the transmit tables is configured to include a second channel identification entry that identifies the one channel.

20. A communications system, comprising:
   means for selecting a first scan table entry from a first scan table;
   means for transmitting data based on the selected first scan table entry;
   means for transmitting scan table identification;
   means for receiving the data and the scan table identification; and
   means for selecting a second scan table based on the received scan table identification.

21. The communications system of claim 20, wherein the means for selecting a first scan table entry comprises:
   means for periodically selecting the first scan table.

22. The communications system of claim 20, wherein the means for selecting a first scan table entry comprises:
   means for periodically initiating a scan table entry selection; and
   means for managing channel buffer status feedback.

23. The communications system of claim 20, comprising:
   means for selecting a second scan table entry from the selected second scan table.
24. The communications system of claim 20, comprising:
   means for synchronizing a first pointer into the selected second scan table
   and a second pointer into the first scan table;
   means for adding first scan table entries to the first scan table and second
   scan table entries to the second scan table; and
   means for dropping first scan table entries from the first scan table and
   second scan table entries from the second scan table.

25. A method for communicating channel information in a communications
    system, comprising:
    selecting a transmit table;
    transmitting a receive table identification based on the selected transmit
    table;
    receiving the receive table identification; and
    selecting a receive table based on the received receive table
    identification.

26. The method of claim 25, comprising:
    selecting a receive table channel identification entry from the selected
    receive table; and
    incrementing a receive pointer into the selected receive table.

27. The method of claim 26, comprising:
    synchronizing the receive pointer into the selected receive table and a
    transmit pointer into the selected transmit table.

28. The method of claim 25, comprising:
    selecting a transmit channel identification entry from the selected
    transmit table; and
    transmitting data based on the selected transmit channel identification
    entry.
29. The method of claim 25, wherein selecting a transmit table comprises:
initiating a transmit table selection periodically with a timer.

30. The method of claim 25, comprising:
managing channel buffer status feedback.
### Fig. 9A

<table>
<thead>
<tr>
<th>SEQ</th>
<th>CHANNEL ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
</tr>
</tbody>
</table>

### Fig. 9B

<table>
<thead>
<tr>
<th>RX SCAN TABLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEQ</td>
</tr>
<tr>
<td>-----</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>...</td>
</tr>
</tbody>
</table>

### Fig. 9C

<table>
<thead>
<tr>
<th>TX SCAN TABLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEQ</td>
</tr>
<tr>
<td>-----</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>...</td>
</tr>
</tbody>
</table>