Abstract:
Techniques for use in Flexible Ethernet data communication include communicating data received on multiple Ethernet inputs having multiple interface rates to an optical network, include executing at least one media access control (MAC) protocol stack instance, constructing an elastic/transparent container operating on data at a MAC layer, configuring an output interface to carry data in time slots according to a transmission schedule, processing data packets received from the multiple Ethernet interfaces through a network protocol stack implemented on the network apparatus to generate multiple processed multiple Ethernet data streams, allocating data packets from the processed multiple Ethernet data streams to time slots of an output data stream transmitted out of the output interface according to the transmission schedule, and using a routing policy by which multiple optical data units of the optical network are to receive the output data stream.

Title: CHANNELIZATION FOR FLEXIBLE ETHERNET

FIG. 26
Published:
— with international search report (Art. 21(3))
CHANNELIZATION FOR FLEXIBLE ETHERNET

TECHNICAL FIELD

[0001] This document relates to a channelization method based on flexible Ethernet, and in particular to flexible Ethernet technology in the Optical Interworking Forum (OIF) field.

BACKGROUND

[0002] To satisfy the ever-increasing demand for data bandwidth, and continued pressure to keep capital and operational expenses low, technology vendors and network operators are looking for ways by which to use available network capacity as efficiently as possible. Optical Interworking Forum (OIF) organization in early 2015 set up a flexible Ethernet (Flex Ethernet or FlexE) project group to address some problems encountered in current data transmission networks. Flex Ethernet is expected to support varying payload bandwidth per wavelength due to optimization of the modulation format.

SUMMARY


[0004] In one example aspect, a method is provided for transferring data from one or more input data streams to a flexible Ethernet (FlexE) shim layer for transmission and includes operating flexible containers to receive the one or more input data streams to generate, from each flexible container, one or more output streams; processing one or more output streams from each flexible container in one or more reconciliation sublayers to generate data signals; and providing, over a modified industry-standard interface, the data signals to a FlexE shim layer in which the data signals are encoded to generate logical serial data streams of encoded blocks representing FlexE clients which are mapped to allocated timing slots for transmission. Each flexible container and the processing in the one or more reconciliation sublayers enable the modified industry-standard interface to include one or more interfaces having respective rates to accommodate for transmission via the FlexE shim layer.

[0005] In another example aspect, a method and an apparatus for implementing a technique of transferring data from one or more input data streams to a FlexE shim layer for transmission
are disclosed. The technique includes operating flexible containers to receive the one or more input data streams to generate, from each flexible container, one or more output streams, processing one or more output streams from each flexible container in one or more reconciliation sublayers to generate 64-bit data signals and providing, over a modified industry-standard interface, the 64-bit data signals to a FlexE shim layer in which the 64-bit data signals are each encoded using a 64B/66B encoding to generate logical serial data streams of 64B/66B blocks representing FlexE clients which are mapped to allocated timing slots for transmission. The modified industry-standard interface comprises one or more industry-standard interfaces having a same standard rate, or one or more Mil interface having different standard rates, or one or more industry-standard interfaces having a non-standard rate. In the technique, each flexible container corresponds to a FlexE client and a processing rate of each flexible container matches a rate of a logical data stream for a corresponding FlexE client.

[0006] In another example aspect, a method and an apparatus for implementing a technique of transferring data from a flexible Ethernet (FlexE) shim layer to one or more output data streams during reception of the data is disclosed. The technique includes providing, over a modified industry standard interface, 64-bit data signals from a FlexE shim layer that decodes, using a 64B/66B decoding, logical serial data streams of 64B/66B blocks representing FlexE clients which are mapped to allocated timing slots in the data, processing the 64-bit data signals through one or more reconciliation sublayers to generate one or more data stream inputs to flexible containers and operating the flexible containers to output the one or more output data streams from the received one or more input streams. The modified industry-standard interface comprises one or more industry-standard interfaces having a same standard rate, or one or more Mil interface having different standard rates, or one or more industry-standard interfaces having a non-standard rate. Using the technique, each flexible container corresponds to a FlexE client and processing rate of each flexible container matches a rate of a logical data stream for a corresponding FlexE client.

[0007] In another example aspect, a method and an apparatus for a technique of communicating data from multiple Ethernet inputs having multiple interface rates to an optical network are disclosed. The technique includes constructing one or more elastic/flexible container operating on data at a media access control (MAC) layer, wherein the elastic/flexible container comprises a variable length data structure, configuring an output interface having an
interface rate equal to a total of the multiple interface rates, wherein the output interface is configured to carry data in time slots according to a transmission schedule, processing data packets received from the multiple Ethernet interfaces through a network protocol stack implemented on the network apparatus to generate multiple processed multiple Ethernet data streams, allocating data packets from the processed multiple Ethernet data streams to time slots of an output data stream transmitted out of the output interface according to the transmission schedule, and communicating, to an optical network, a routing policy by which multiple optical data units of the optical network are to receive the output data stream.

[0008] In another example aspect, a method and apparatus for receiving data from an optical network and transmitting over multiple Ethernet interfaces are disclosed. The disclosed method and apparatus implement a technique that includes executing at least one MAC protocol stack instance on the network apparatus, constructing an elastic/flexible container operating on data at a MAC layer, wherein the elastic/flexible container comprises a variable length data structure, configuring an input interface having an interface rate equal to a total of the multiple interface rates, wherein the input interface is configured to carry data in time slots according to a transmission schedule, receiving, a routing policy by which multiple optical data units of the optical network are transmitting data on to the input interface, and selecting data packets from the input interface according to the transmission schedule for processing through a network protocol stack implemented on the network apparatus for transmission to the multiple Ethernet outputs.

[0009] In yet another example aspect, a technique for transmitting and receiving data using a FlexE shim includes an apparatus for transferring data from a first group of one or more input data streams to a flexible Ethernet (FlexE) shim layer for transmission, and transferring, from the FlexE shim layer to a first group of one or more output data streams for reception. The apparatus includes a number of flexible containers for receiving the first group one or more input data streams to generate, from each flexible container, a second group of one or more output streams, one or more reconciliation sublayer modules for processing the second group of one or more output streams from each flexible container to generate a first group of 64-bit data signals, and a modified industry-standard interface for providing the first group of 64-bit data signals to a FlexE shim layer in which the first group of 64-bit data signals are each encoded using a 64B/66B encoding to generate a first group of logical serial data streams of 64B/66B blocks.
representing FlexE clients which are mapped to a first group allocated timing slots for transmission. The one or more reconciliation sublayers further process the second group of 64-bit data signals to generate a second group of one or more data stream inputs to flexible containers. The flexible containers further outputs the first group of one or more output data streams from the received second group of one or more input streams. The modified industry-standard interface comprises one or more industry-standard interfaces having a same standard rate, or one or more Mil interface having different standard rates, or one or more industry-standard interfaces having a non-standard rate. In the technique, each flexible container corresponds to a FlexE client and a processing rate of each flexible container matches a rate of a logical data stream for a corresponding FlexE client.

[0010] These and other aspects, and their implementations and variations are set forth in the drawings, the description and the claims.

BRIEF DESCRIPTION OF THE DRAWINGS

[0011] FIG. 1 shows a block diagram of an example of a 100G line card with multiple Traffic Management (TM) chips.

[0012] FIG. 2 illustrates an example of a bottleneck of fixed-rate Ethernet with respect to the current flexible IP and optical transmission synergetic networks.

[0013] FIG. 3 is a block diagram example depicting the location of FlexE SHEVI layer within the IEEE802.3 stack.

[0014] FIG. 4 shows an example of data channelization for a FlexE client data stream channelized to form a calendar.

[0015] FIG. 5 shows another example of data channelization for a FlexE client data stream channelized to form a calendar.

[0016] FIG. 6 illustrates an example of flexible Ethernet networking by a router and an OTN device connected via a port.

[0017] FIG. 7 illustrates an example of flexible Ethernet networking by a router and an OTN device connected with four PHY bonding together.

[0018] FIG. 8 shows an example of an optical communication network.

[0019] FIG. 9 illustrates an example of mapping of a media independent interface (Mil) corresponding to an elastic/flexible container.
FIG. 10 illustrates an example of a mapping of an Mil corresponding to a 150G elastic/flexible container.

FIG. 11 illustrates an example of a mapping of an Mil corresponding to a 135G elastic/flexible container.

FIG. 12 illustrates an example extended 10 Gigabit Mil (XGMII) interface.

FIG. 13 illustrates an example of a ten Gigabit attachment unit interface (XAUI).

FIG. 14 illustrates an example of three FlexE client flows set in a router.

FIG. 15 is a distribution diagram showing an example of a FlexE calendar.

FIG. 16 illustrates an example cache structure of a chip.

FIG. 17 illustrates an example of a free cache organized using a linked queue.

FIG. 18 illustrates an example of allocation and freeing of a cache space.

FIG. 19 illustrates an example of a data-send operation of a data transmission apparatus.

FIG. 20 illustrates an example of a data-receive operation of a data transmission apparatus.

FIG. 21 illustrates an example flowchart for a method of data transmission.

FIG. 22 illustrates an example of a data transmission apparatus.

FIG. 23 illustrates an example flowchart for a method of receiving data transmissions.

FIG. 24 illustrates an example of a data reception apparatus.

FIG. 25 illustrates an example of a data structure.

FIG. 26 shows an example flowchart for a method of data communication.

FIG. 27 shows an example flowchart for another method of data communication.

FIG. 28 shows an example of a FlexE mux structure.

FIG. 29 shows an example of a FlexE de-mux structure.

FIG. 30 shows an example transport network that is unaware of FlexE carrying FlexE data.

FIG. 31 shows an example of a transport network that is aware of Flex E carrying FlexE data.
DETAILED DESCRIPTION

[0042] Ethernet is a universally used data connection interface. Currently deployed Ethernet products are often named after the connection rate achieved by the physical layer, e.g., 10 Mbit/s, 1 Gbps, and so on. The current Ethernet interfaces are fixed in bandwidth, and thus are not able to use the flexibility of bandwidth that can be achieved by a packet switching equipment, because a packet switching equipment only provides a single stream with fixed bandwidth externally, e.g., because Ethernet is available only at a few fixed rates. Thus, present day technologies fail to make use of the transmission bandwidth flexibility that can be achieved by aggregating, or combining, traffic from multiple Ethernet devices with different connection rates.

[0043] The Flex Ethernet project managed by the OIF is led by Cisco, and co-initiated by Juniper, Finisar, and Xilinx and other manufacturers. Flex Ethernet defines channelization, binding and sub-rate functions by standardizing a flexible Ethernet (FlexE) MAC interface. FlexE enables a standard Ethernet physical media dependent layer (PMD) to connect one or more Ethernet MACs, and provides efficient channel binding standard rate and non-standard rate, so that Ethernet switches and routers can be configured with different bandwidth on demand, and thus increases flexibility of bandwidth configuration of data center network. Unless otherwise noted, the terms used in the present document are consistent with their meaning in the FlexE Implementation Agreement Draft 1.1., Release date July 2015 (IA OIF-FLEXE-01.0), which is incorporated herein in its entirety.

[0044] When using a sub-rate function, the transmission pipeline rate may be lower than the PMD rate of Ethernet, and a matching of the rate of router port to the transfer rate may be performed to support non-standard Ethernet rate. In this context, binding may refer to a bound to an Ethernet physical layer. For example, a 200G MAC can be supported on two bound 100GBASE-R physical layers. For example, a large Ethernet rate may be divided into multiple (e.g., three or four) sub-rates, and resources corresponding to each sub-rate may be bound to that sub-rate. This mapping between a high bandwidth connection and multiple low bandwidth connections is also sometimes called channelization.

[0045] The Flex Ethernet initiative is also expected to extend the multi-link gearbox (MLG) specification, which is a protocol specification for data transmission, to be physical medium dependent (PMD), and will provide efficient channel for binding standard and non-standard rates.
For optical networks, with the development of ODUflex (Optical channel Data Unit flexible), streams of flexible grid, bandwidth variable transponders (BVT), flexible reconfigurable optical add-drop multiplexers (ROADM) and flexible OTN, and flexible bandwidth can be carried via optical network. This means that the fiber-optic network has the infrastructure capability that can be used for traffic management with flexible bandwidths. Similarly, data packet forwarding devices, such as Ethernet routers or switches, can handle substantially flexible bandwidth stream.

FIG. 1 is a block diagram illustration of an example of a line card 100 that communicatively couples Ethernet data connections with optical transmission equipment. In some embodiments, the technology disclosed in the present document can be implemented on a line card that offers data connectivity between Ethernet electrical signals and optical signals carrying data traffic in an optical network.

FIG. 2 shows an example block diagram to illustrate bottleneck of the fixed-rate Ethernet to the current flexible IP and optical coordinated transport networks. During processing of data packets in some devices, the packets flow through the network processor (NP) units and traffic management (TM) units 202 (both these units can control the flow of bandwidth) e.g., as depicted by flexible rate output streams 210. Using these two units, an Ethernet router/switch can easily generate different streams with different rates and bandwidth on demand at the Ethernet interface 204. However, for Ethernet router/switch, bandwidth of Ethernet interface defined by IEEE 802.3 is a fixed rate, such as 10G, 40G, 100G and 400Gbps, e.g., as depicted by 208. Due to limitations of fixed-rate Ethernet interfaces, the flexibility of an Ethernet router/switch is not fully utilized for feeding data to flexible optical equipment 206 that includes such optical transport devices as a flexible optical module, an ODUflex, a flex ROADM, etc.

The potential of flexibility, that grouping devices already have, can be used to build a flexible network, but a new type of flexible Ethernet technology is useful in enabling a flexible network. The FlexE framework proposed in the OIF provides a common mechanism to support multiple Ethernet MAC layer rates, it may or may not correspond to any existing Ethernet PHY rate. Because of having a more flexible and universal channel bonding characteristics, the followed channel flexibility, sub-rate flexibility, as well as an important feature of no need to modify the PMD, will enable flexible Ethernet to leverage the future market application, and usher in an emerging Ethernet and optical transport market.
FIG. 3 shows an example of the positioning of the FlexE SHEVI layer 302 in the IEEE802.3 stack 300 that implements the flexible Ethernet technology.

After data streams having certain bandwidths reach the MAC layer, e.g., through the Mil interface, they form parallel data streams (e.g., XGMII Interface 32-bit parallel data stream). These data streams are combined into 64-bit data signal TXD <63: 0>. Then, FlexE SHIM layer 302 performs 64B/66B encoding on the data from Mil interface, and generates a 66-bit block. The 66-bit block is formed of two parts, one part is the 2-bit synchronization header, and the other part is a 64-bit payload, the logical serial stream of 64B/66B block is called FlexE client (FlexE client) in FlexE technology.

As seen in the example from FIG. 3, the flexible Ethernet technology adds a FlexE SHIM layer 302 on the original Scramble (scramble) as defined in conventional 802.3 stack, and bypasses the original 64B/66B encoding and decoding process, while locating the 64B/66B encoding and decoding process to the top of the SHEVI layer.

A FlexE group may be composed of 1 to n 100GBASE-R Ethernet physical layer devices (PHY). Each physical layer devices uses most PCS functions described in section 82 of the IEEE draft standard 802.3-2015, including the PCS channel distribution, channel tag insertion, alignment and correction. All PHY in a FlexE group may use the same physical layer clock. FlexE payload carried on each PHY on the FlexE group has a format of logic serial stream of valid 64B/66B block, except for the marks occupied by aligned mark of PCS channel (cannot carry FlexE payload).

Each FlexE client may represent a 64B/66B block logical serial stream of an Ethernet MAC layer. In the MAC layer, an elastic/flexible container may be configured to operate at a variable speed to match the rate of the MAC stream. According to the OIF standard, the MAC layer of a FlexE client can run at a rate of 10, 40 and mx25Gb/s, where m is an integer. The 64B/66B encoding is based on IEEE standard 802.3-2015 Figure 82-4.

The FlexE mechanism functions by using a calendar, and allocates 66B block location on each PHY of FlexE group to each FlexE client. A calendar may use a basic unit of a time slot. In some embodiments, the time slot may correspond to 5Gbps size and a length of 20 time slots for every 100G capacity of FlexE group.

FIG. 28 shows an example of functions performed in a FlexE mux in the transmit direction. Data from FlexE clients is passed through a 64B/66B encoding module and an idle
insert/delete module before being distributed and inserted into a master calendar. What is presented for insertion into the slots of the FlexE master calendar is a stream of 64B/66B encoded blocks encoded per IEEE Std 802.3-2015 Table 82-4 which has been rate-matched to other clients of the same FlexE shim. This stream of 66B blocks might be created directly at the required rate using back-pressure from a network processor. It might come from a single-lane Ethernet PHY such as 10G or 25G, where the process of rate-matching involves both idle insertion/deletion, plus converting the rate-aligned stream from the 4-byte alignment of IEEE Std 802.3-2015 clause 49 to the 8-byte alignment of IEEE Std 802.3-2015 clause 82. The stream of blocks may come from a multi-lane Ethernet PHY, where the lanes need to be deskewed and re-interleaved with alignment markers removed prior to performing idle insertion/deletion to rate match with other clients of the same FlexE shim. Or the stream may have come from another FlexE shim, for example, connected across an OTN network, where all that is required is to perform idle insertion/deletion to rate match with other clients of the same FlexE shim.

With reference to FIG. 15, for a FlexE group comprising n bound 100GBASE-R PHY, the logical length of the master calendar is 20n (1502). According to FIG. 15, each master calendar has the allocated block, is allocated to n sub calendar with a length of 20 on each PHY of FlexE group (1504). The allocation order of once every 20 blocks are selected in the simply cyclic allocation on the 66B block, to facilitate the PHY to add to FlexE server without the need to change existing calendar slot allocated to FlexE client.

FIG. 29 shows example functions of the FlexE demux (the FlexE shim in the receive direction). The 100GBASE-R Lower Layers, PMA, lane deskew, interleave, AM removal, descramble, which represent the layers of each 100GBASE_R PHYs below the PCS, are used exactly as specified in IEEE Std 802.3-2012. The PCS lanes are recovered, deskewed, reinterleaved, and the alignment markers are removed. The aggregate stream is descrambled.

Calendar Interleaving and Overhead Extraction: The calendar slots of the each PHY are logically interleaved in a specified order. The FlexE overhead is recovered from each PHY. LF Generator: In the case that any PHY of the FlexE group has failed (PCS_Status=FALSE) or overhead lock or calendar lock has not been achieved on the overhead of any of the PHYs, LF is generated to be demapped from the master calendar for each FlexE PHY.
66B Block Extraction from Master Calendar: The 66B blocks are extracted from the master calendar positions assigned to each FlexE client in a pre-specified order.

Idle Insertion/Deletion, 66B Decoding: Where these functions are performed and whether they are inside or outside the FlexE implementation is embodiment-specific. The 66B blocks could be delivered directly to a network processor. If delivered to a single-lane PHY, idle insertion/deletion may be used to increase the rate to the PHY rate, realigning to 4-byte boundaries in the process (for 10G or 25G) and recoding 64B/66B according to clause 49. For a multi-lane PHY, idle insertion/deletion is used to increase the rate to the PHY rate less the space needed for alignment markers, the blocks are distributed to PCS lanes with AM insertion. For a FlexE client mapped over OTN, idle insertion/deletion may be used to adjust the rate as required for the OTN mapping.

A SHIM layer of flexible Ethernet realizes the flexible Ethernet basic functions by mapping Flex client data stream to the corresponding calendar time slot set.

However, the current flexible Ethernet technology only considers splitting the standard rate of port into different sub-rate according to need, without mentioning the issue for how to distribute traffic to channels with various sub-rates, and there is no clear method on how the data stream is assigned to each sub-channel. Currently, data stream transmitted from the MAC layer passes through the MIL. For 400G connection rate, this is called CDMII, where the name of Mil corresponding to other rate will change accordingly. For example, the interface name for 10Gb/s is the XAUI interface, or XGMMI interfaces, media-independent interface name for 100Gb/s is CGMII interface, which is a 64-bit parallel interface (the data distribution is based on 64-bit units) interface, and then passed to the physical coding sublayer to process. Due to this, the current Mil interface rate is fixed at the standard rate, it cannot match the rate of FlexE client, which makes the potential of open architecture of the flexible Ethernet not being fully exploited.

Due to the current limitations of flexible Ethernet technology, multiple FlexE client data streams from the router can only share a logical path, which is named logic PHY. If these FlexE client data stream will be destined to different OTN nodes, after these MAC data streams are transmitted to the OTN equipment, appropriate treatments are carried out, otherwise an error shown in FIG. 8 will occur.

In the current proposal of flexible Ethernet standard, there is no clear explanation on how each FlexE client is mapped to time slots in the calendar. The techniques disclosed in the
present document can be used to, among other aspects, allocate a FlexE client data stream to channels with different sub-rate.

[0067] Brief overview

[0068] The FlexE technology currently provides no methods for allocating FlexE client data streams to different sub-rate channels, as well as for traffic shaping and rate matching. The present document provides, among other things, a method of stream distribution to different sub-rate channels.

[0069] In some embodiments, an apparatus to distribute traffic to various sub-rate channels includes one or more optical modules to supports FlexE technology and various stack methods that support FlexE.

[0070] In some embodiments, in order to structure flexible channel bandwidths from MAC layer to a FlexE SHEVI layer, and finally to the Physical layer, and to match rates of MAC layer, FlexE SHEVI and Physical layer with each other, flexibility and changeability solutions can be implemented at one or more of a MAC layer, a FlexE SHIM layer and a Physical layers.

[0071] In some embodiments, an elastic/flexible container structure can be constructed in the MAC layer in order to carry the MAC stream with the changed rate. A flexible Mil interface in the SHIM layer may be used to carry data streams with a different rate from an upper layer. The modified Mil interface can be adapted to MAC layer data stream with changed bandwidth.

[0072] After a data stream having a certain bandwidth reaches the MAC layer, travels through the reconciliation sublayer (RS), and Mil interfaces of different types configured to form parallel data streams with a variety of different rates, the data streams are combined into a data signal TXD 64 bits <63: 0>. In some embodiments, the reconciliation sublayers map the logical MAC physical layer service primitives to and from standard electrical interfaces used by the PHYs. In some embodiments, the RS maps signals from the upper layer (MAC) to physical signaling primitives understood by the sublayer below the RS. A FlexE modules from SHIM Mil interface encodes the data into 64B/66B encoding. The resulting 64B/66B encoding is mapped to the corresponding slot on demand in accordance with the general principles and flexible Ethernet.

[0073] In some embodiments, the technique may include constructing in the MAC layer an elastic/flexible container that matches target FlexE client's various rates. The components of the container include, but are not limited to, a cache, a FIFO (buffer), logic resources, related circuit resources, etc., for loading FlexE client data streams. The elastic/flexible container
corresponding to each data stream for different FlexE clients may be logically separated from each other.

[0074] In some embodiments, the technique may include constructing an Mil interface type set that matches commonly used FlexE client rates. In some embodiments, the set members include:

[0075] Already-existing various Mil interfaces, Mil interface with the rate of 5Gbps, new logic Mil interface formed by combining various Mil interfaces, and a new logic Mil interface formed by combining n same-type interfaces of each Mil interface, and n is a natural number.

[0076] In some embodiments, various Mil interface still correspond to their matching RSs, and the new logic Mil interfaces formed by various combinations still correspond to their matching logic RSs.

[0077] In some embodiments, a new Mil interface, unlike current rate of standard Ethernet, has a rate of 5Gbps, which is constructed by reducing the clock rate of XGMII interface with 10Gbps to half of the original one. In some embodiments, a clock rate change of N/M may be used, where N and M are both integers.

[0078] In some embodiments, the new Mil interface, unlike current rate of standard Ethernet, has a rate of 150Gbps, which is constructed by increasing the clock rate of XGMII interface with 100Gbps to 1.5 times of the original one.

[0079] In some embodiments, the new Mil interface, unlike current rate of standard Ethernet, has a rate of 300Gbps, which is constructed by reducing the clock rate of CDMII interfaces with 400Gbps to 0.75 times of the original one.

[0080] In some embodiments, the new Mil interface, unlike current rate of standard Ethernet, has a rate of 5Gbps, which is constructed by increasing the clock rate of XGMII interface with 10Gbps to four times of the original one.

[0081] In some embodiments, the above various new non-standard-rate Mil interfaces still correspond to their matching RS sublayers.

[0082] Each FlexE client data stream will be corresponded to a member Mil interface or logic Mil interface in the Mil interface type set through RS, so as to generate 64B/66B data to be mapped into a corresponding ODUflex, and the ODUflex is corresponding to a time slot assigned to the MAC client in a 20n Master Calender.
[0083] Optical module that supports FlexE technology does not need to modify PMD, and methods that support general principles of FlexE can refer to oif2015.127.02 document.

[0084] FIG. 4 depicts a system 400 in which each single instance of a FlexE client provides data being coupled to a single instance of ODUflex. In comparison, in FIG. 5, a system 500 is depicted, in which a single FlexE client is supporting data communication over traffic split into two Mil flows, each having its own 64 / 66 bit conversion stage and a their own reconciliation stage, which individually may have a standard Mil rate, that are being fed to the ODUflex stage.

[0085] The FlexE channelization can be accomplished by implementing the following processing operations.

10 [0086] Preparatory operation:

[0087] - Construct, in the MAC layer, an elastic/flexible container that matches target FlexE client's various rates. The components of the container include, but are not limited to, cache, FIFO, logic resources, the related circuit resources, etc., for loading FlexE client data streams. The elastic/flexible container corresponding to each data stream for different FlexE clients is separated from each other.

[0088] - Construct an Mil interface type set that matches commonly used FlexE client rates. The set members comprise already-existing various Mil interfaces, Mil interface with the rate of 5Gbps, new type interface formed by combining various Mil interfaces, and a new type interface formed by combining n same-type interfaces of each Mil interface, and n is a natural number.

20 [0089] According to demanded FlexE client rate, a logical Mil interface is constructed that matches such rate. Various Mil interface still correspond to their matching RSs, and the new logic Mil interfaces formed by various combinations still correspond to their matching logic RSs.

[0090] Next, in a first operation, the MAC layer of a FlexE client may operate at a rate of 10, 40 or m x 25 Gbps. In some embodiments, the MAC layer of a FlexE client may also operate at a multiple of the following rates: multiples of 5 Gb/s, 10 Gb/s, 25 Gb/s, 40 Gb/s, 50 Gb/s, 100 Gb/s, 200 Gb/s, 300 Gb/s, 400 Gb/s, 1T Gb/s. Alternatively or additionally, multiples, or linear combinations of sub-multiples of the rates may be used (e.g., combination using weights that are rational numbers - where numerator and denominator of the multiple are integers).

[0091] In a second operation, each MAC data stream passes through RS sublayer and Mil interfaces in an one-to-one correspondence (or 1:n mapping relationship, where n is a natural number). The 64B/66B data generated (belong to FlexE SHEVI layer from here) are mapped to a
corresponding ODUflex. In some embodiments, the Mil interface can also be replaced with XAUI, XGMII interfaces, etc., as appropriate.

[0092] In a third operation, the ODUflex is corresponding to time slots assigned to the MAC client in the 20n Master Calendar. The data inside ODUflex are put into the time slot set to form a multiframe.

[0093] The multiframe is formed by each FlexE data block separated by 20 x 1024 and a corresponding overhead block. The structure of the overhead block can be referred to FIG. 25, and the time slot structure in the multi-frame can refer to FIG. 15.

[0094] In the fourth operation, the router/switch setting policies enable multiple data streams (for example, data streams with a specific destination MAC address, a specific destination IP address, or data stream with a specific protocol, and the setting mode is not limited) to be sent to a destination output port that supports FlexE technology in the local device;

[0095] This operation can be achieved via the command lines, network management software, or SDN controllers.

[0096] After the data stream reaches the MAC layer, through the Mil interface, through the processing of FlexE SHIM to form a logic serial stream of 64B/66B block, that is FlexE client (FlexE client).

[0097] In the fifth operation, FlexE client generated is processed in the order of above first step to the fourth step.

[0098] In one advantageous aspect, the method described here achieves the effect of channelized allocating traffic to each of different sub-rate channels, with each FlexE client data flowing into the channel of their own time slot which will not be mixed together. This technique thus gives business applications a very flexible means to control the pipeline. Parameters of sub-channel can be flexibly controlled through either the command line, or SDN network controller.

[0099] Example Embodiments

[00100] Preparatory operation:

[00101] - Construct in the MAC layer an elastic/flexible container that matches target FlexE client's various rates. The components of the container include, but are not limited to, cache, FIFO, logic resources, the related circuit resources, etc., for loading FlexE client data streams.

The elastic/flexible container corresponding to each data stream for different FlexE clients is separated from each other.
[00102] Construct a Mil interface type set that matches commonly used FlexE client rates. The set members comprise already-existing various Mil interfaces. Mil interface with the rate of 5Gbps, new type interface formed by combining various Mil interfaces, and a new type interface formed by combining n same-type interfaces of each Mil interface, and n is a natural number.

[00103] According to demanded FlexE client rate, constructing a logical Mil interface that matches such rate.

[00104] Various Mil interface still correspond to their matching RSs, and the new logic Mil interfaces formed by various combinations still correspond to their matching logic RSs. In some embodiments, the new Mil interface, unlike current rate of standard Ethernet, has a rate of 5Gbps, which is constructed by reducing the clock rate of XGMII interface with 10Gbps to half of the original one. In some embodiments, the new Mil interface, unlike current rate of standard Ethernet, has a rate of 150Gbps, which is constructed by increasing the clock rate of XGMII interface with 100Gbps to 1.5 times of the original one.

[00105] In some embodiments, the new Mil interface, unlike current rate of standard Ethernet, has a rate of 300Gbps, which is constructed by reducing the clock rate of CDMII interfaces with 400Gbps to 0.75 times of the original one.

[00106] In some embodiments, the new Mil interface, unlike current rate of standard Ethernet, has a rate of 5Gbps, which is constructed by increasing the clock rate of XGMII interface with 100Gbps to four times of the original one.

[00107] The above various new non-standard-rate Mil interfaces still correspond to their matching RS sublayers.

[00108] In a first operation, a MAC layer of a FlexE client may operate at a rate of 10, 40, or m x 25 Gb/s. The MAC layer rate FlexE client may also be a multiple of the following rates: 5 Gb/s, 10 Gb/s, 25 Gb/s, 40 Gb/s, 50 Gb/s, 100 Gb/s, 200 Gb/s, 300 Gb/s, 400 Gb/s, 1T Gb/s.

[00109] In a second operation, each MAC data stream passes through RS sublayer and Mil interfaces in an one-to-one correspondence (or 1:n mapping relationship, where n is a natural number). The 64B/66B data generated (belong to FlexE SHEVI layer from here) are mapped to a corresponding ODUflex; said Mil interface can also be replaced with XAUI, XGMII interfaces, etc.
In a third operation, the ODUflex is corresponding to time slots assigned to the MAC client in the 20n Master Calendar. The data inside ODUflex are put into the time slot set to form a multiframe.

In a next operation, the router/switch setting policies are set to enable multiple data streams (for example, data streams with a specific destination MAC address, a specific destination IP address, or data stream with a specific protocol, and the setting mode is not limited) to be sent to a destination output port that supports FlexE technology in the local device.

This operation can be achieved via the command lines, network management software, or SDN controllers.

After the data stream reaches the MAC layer, a logic serial stream of 64B/66B block forms, the logic serial stream is called FlexE client (FlexE client) in FlexE technology;

In a next operation, the FlexE client generated is processed in the order of above operations.

With the above processing, FlexE channelization is realized. Some example embodiments are discussed herein.

Embodiment 1

This embodiment illustrates an example of how to construct an elastic/flexible container, and how to correspond sub-streams with different rates, elastic/flexible container that the sub-streams belongs to, and a cached rate.

In designing an Ethernet switch and a control chip, elastic/flexible containers that match various target rate of FlexE client may be constructed. Taking into account to maximize the use of cache resources, to meet the design conditions, resources that form the elastic/flexible container may be established as a shared structure, for example, shared cache structure, and shared TM traffic management, etc.

Elastic/flexible container planning is considered in both in and out direction, and therefore the corresponding resources should be divided into two sets as sending and receiving, which are shown in FIG. 19 and FIG. 20.

The transmission direction uses three modules, including a stream distributor, an elastic/flexible container and a channel distributor. The receiving direction uses three modules, including a streaming aggregator, an elastic/flexible container and a channel aggregator.
Referring to FIG. 19, the stream distributor module is a switching module that is responsible for the formation of sub-streams, its input is from the input port of a router/switch, and the output of stream divider comprises n independent sub-streams that are sent to flexible containers. The stream distributor recognizes the input streams, and assigns each of them to one of n output sub streams. The input of channel distributor is from a sub-stream from the elastic/flexible container. A few designated physical channels (e.g., time slot set) are used for loading sub-streams based on the configuration. Channel allocation is responsible for loading a sub-stream to the specified physical channel. The channel aggregator functions the opposite of the channel distributor. It receives the stream from the specified channel, restores the data packets, and then sent to the elastic/flexible container. In the receiving direction, the data streams from the channel are grouped, and are aggregated on the channel aggregator, the packets are restored, and are written into a flexible container. Then, the packet is read out by the stream aggregation from all the flexible containers, and a complete stream from the receiving direction is combined and recovered.

Because bandwidth for each sub-stream is not necessarily the same, elastic/flexible container resources may also be changed. FIG. 16 shows the internal cache structure of a switch chip. Assuming a capacity of 180 KB on-chip data packet cache space is divided into 720 units, each called a page, and each page with a capacity of 256 Byte, for storing the received data packet. Cache space is allocated in units of pages for each data packet, and larger packets can occupy more than one pages of cache space. In addition to packet cache region, the data cache space also stores several groups of queues for the transmitting descriptor. Each queue is aimed at each transmission port in the chip, which stores the transmitting descriptor for the port. Each transmitting descriptor packet contains the index number for address of data packet cache space and associated control information. The internal chip of router/switch device organizes free cache space by linked queue. A queue represented by linked list can be short as linked queue. A linked queue requires two pointers indicating the head of the queue and the tail of the queue (referred to the head pointer and tail pointers) to be uniquely determined. It is a single linked table with both a head pointer and a tail pointer, and its operating characteristic is FIFO. Addresses for all free cache spaces in switch and control chip are managed using a linked table. The linked table is a free address queue, wherein the free list represents free address queue,
which is formed by the free address head pointer Free_Hptr and tail pointer Free_Tptr. All operations on the free cache management are focused on free list for free address queues.

[00123] FIG. 17 shows the a list of states for cache space block 0, 3, 5, 1, 7, 719, 6 that are free, as well as contents in CRAM and Free_Hptr, Free_Tptr register. Free block list uses an initialization process before use. After the initialization, content in Free_Hptr register is 0, content in word 0 of CRAM 0 is 1. Content in word 1 is 2, ..., Content in word 718 is 719. Content in Free_Tptr register is 719.

[00124] The allocation of free cache space is to take free cache unit from the head of list, while the release of cache space is to add the already-processed cache unit to the end of the list. FIG. 18 depicts the allocation and release of cache space. When the list has only one item, namely Free_Hptr register has the same contents as the Free_Tptr register, after allocation operation, value in the Free_Tptr register should be modified accordingly. Defining both Free_Hptr and Free_Tptr register as all Is indicates an invalid state. When the list is empty, contents in both Free_Hptr and Free_Tptr are all Is, at this time there is no cache space to be allocated. For the operation to release free unit, the Free_Hptr and Free_Tptr register only need to be set to the newly released free unit number. While the embodiment described herein assumes that the on-chip buffer is 180 KB in size as an example, it may be possible to use larger on-chip buffers for processing that is consistent to the above description, including using on-chip buffers of storage capacities of 3-4 orders of magnitude greater than 180 KB or even larger buffers.

[00125] In the above example for the cache design, based on the expected size of the elastic/flexible container, setting the number of pages managed by a linked queue corresponding to the elastic/flexible container is actually corresponding to the rate of the FlexE client. The ratio of the numbers of pages allocated by FlexE clients with different rates may be equal to the ratio of the respective rates. In this way, the aim of dynamic and flexible adjustment of cache resources for each sub-streams under the condition of a shared cache is achieved. Depending on the expected size of an elastic container and the size or number of the page(s) managed by corresponding linked list of the elastic container, a dynamic adjustment can be made in allocating the resources to render a ratio of different rates of sub-streams to the number of assigned pages to be equal to the ratio of the individual rates.

[00126] Embodiment 2
[00127] FIG. 9 is a diagram that is corresponding to the embodiment. After the elastic/flexible container is constructed, Mil interface may be constructed next.

[00128] Suppose three FlexE clients have rates of 400G, 25G and 150G respectively, as shown in the example illustration of FIG. 9, three elastic/flexible containers are constructed in the MAC layer to match the target FlexE client rates, for loading the three FlexE clients' data streams. The elastic/flexible containers corresponding to different FlexE clients' data streams are isolated from each other.

[00129] The Mil interface combinations matching commonly used FlexE client's rate are constructed. If a FlexE client has a rate of 400G, the corresponding Mil interface is constructed, which comprises: four 100G Mil interfaces, or eight 50G Mil interfaces, or eighty 5G Mil interfaces.

[00130] **Embodiment 3**

[00131] FIG. 10 is a diagram that is corresponding to the following embodiment. As previously described, a combination of different types of Mil interfaces can be constructed. For example, if a FlexE client has a rate of 150G, the corresponding Mil interface can be constructed, which comprises: one 100G Mil interface and one 50G Mil interface. Similarly, different types and numbers of Mil interface combinations can also be constructed, for example, if a FlexE client has a rate of 135G, the corresponding Mil interface can be constructed, which comprises: two 50G Mil interfaces, one 20G of Mil interface, and three 5G Mil interfaces, as depicted in FIG. 11.

[00132] **Embodiment 4**

[00133] The previous embodiments described how to plan and build a system for transferring FlexE packet traffic. In some embodiments, the system can be pre-set.

[00134] In the networking environment for routers and OTN devices connectivity, as shown in FIG. 6, the Ethernet port on the router has a rate of 400G. Mil interface group (four 100G Mil interfaces) corresponding to 400G rate constructed in accordance with embodiment 2, is called logical Mil interface corresponding to the FlexE client's rate. FlexE flexible Ethernet is enabled on the port, without performing flexible Ethernet binding to PHY of other ports. The port of OTN equipment also has a speed of 400G, and Mil interface group (comprising four 100G Mil interfaces) corresponding to 400G rate can also be constructed the same in accordance with Embodiment 2.
3 FlexE client data flows FlexE client 1, FlexE client 2 and FlexE client 3 are set in the router, which are able to pass the flexible Ethernet ports, with a respective rate of 10Gb/s, 25Gb/s, and 20Gb/s, and the setting method is shown in FIG. 14.

Destination MAC address of data stream from Port1 is set to MAC1, which enters the elastic/flexible container having a capacity of 10G.

Destination MAC address of data streams from port2, port3, port4, port5, port6, port7, port8, port9 is set to MAC2, which enters the elastic/flexible container having a capacity of 25G.

Destination MAC address of data streams from port9, port10 is set to MAC3, which enters the elastic/flexible container having a capacity of 20G. In some embodiments, the above configurations and assignment of port IDs to MACs in the router are done by the network management software.

Each set FlexE client data stream corresponds to a Mil media independent interface or a logical Mil interface in a one-to-one correspondence via the RS sublayer. Each data within the MAC data stream forms parallel data streams via the Mil interface, relying on 64B/66B encoding mapping and coding techniques to generate a 66-bit block for use by FlexE flexible Ethernet SHIM layer. As described with reference to FIG. 3, under the existing FlexE specification, the FlexE SHIM layer is generally understood to include the 64B/66B conversion or encoding module.

Next, the generated 64B/66B data are mapped into a corresponding ODUflex. The ODUflex corresponds to a time slot set assigned to the MAC client in the 20n Master Calendar, and the data inside the ODUflex are put into these time slot set.

Due to the above processing, the MAC data streams into corresponding ODUflex correspond to time slot sets, or groups, assigned to the MAC client in the 20n Master Calendar. These time slot set, ODUflex constitute logic PHY which belongs to the FlexE client data stream, and are channelized. The MAC data stream can therefore be used as a whole, and be transmitted to the destination node intact.

Without the above-described processing flow, in accordance with the existing flexible Ethernet technology, three FlexE client data streams may not be distinguishable from each other, and the sent-to logical PHY will carry three FlexE clients' data streams, and the logical PHY will send the data to one destination node. If the above three FlexE clients' data streams are destined to different nodes, then it will cause some FlexE client's data stream to be sent to the
wrong destination OTN node. In order to avoid occurrence of such situation, logical PHY data should be unloaded on the local OTN node, and layer 2 processing chip will perform switch operation on the Ethernet data as payload. After switch, the data are re-packaged and then sent to different destinations.

[00143] With the processing mechanism in some described embodiments, different FlexE client data streams are channelized before entering the shared physical pipe, which eliminates the need for a following Layer 2 switch in OTN equipment, and truly realizes sub-rate function which flexible Ethernet pursues.

[00144] Embodiment 5

[00145] In this embodiments, port PHY of a router are composed together. The networking environment of connected routers and OTN devices are shown in FIG. 7, wherein four Ethernet ports with rates of 100Gb/s are bound to a flexible Ethernet port of a rate of 400Gb/s.

[00146] The port rate of the OTN equipment is also 400G, wherein similarly four Ethernet ports with rates of 100Gb/s are bound to a flexible Ethernet port of a rate of 400Gb/s.

Corresponding detection and configuration are done by network management.

[00147] 3 FlexE client data flows FlexE client 1, FlexE client 2 and FlexE client 3 are set in the router, which are able to pass the flexible Ethernet ports, with a respective rate of 10Gb/s, 25Gb/s, and 20Gb/s. Each FlexE client data stream corresponds to a Mil media independent interface (or logical Mil interface) in a one-to-one correspondence via the RS sublayer. Each data within FlexE client data stream (the MAC data stream) forms parallel data streams via the Mil interface, relying on 64B/66B encoding mapping and coding techniques to generate 66-bit block, for use by FlexE flexible Ethernet SHIM layer. According to specification defined by a flexible Ethernet technology, the location start from where 66 bits are formed belongs to the category of FlexE SHIM layer.

[00148] Next, the generated 64B/66B data are mapped into a corresponding ODUflex. The ODUflex corresponds to a time slot set assigned to the MAC client in the 20n Master Calendar, and the data inside the ODUflex are put into these time slot sets; other processes are same as the conventional FlexE flexible Ethernet technology.

[00149] Due to the above processing, the MAC data streams into corresponding ODUflex correspond to time slot sets assigned to the MAC client in the 20n Master Calendar. These time slot set, ODUflex constitute logic PHY which belongs to the FlexE client data stream, and are
channelized. The FlexE client data stream can therefore be used as a whole, and be transmitted to the destination node intact.

[00150] Without the above-described processing flow, in accordance with the existing flexible Ethernet technology, three FlexE client data streams will not be distinguished, and the sent-to logical PHY will carry three FlexE clients' data streams, and the logical PHY will send the data to one destination node. If the above three FlexE clients’ data streams are respectively destined to different nodes, then it will cause some FlexE client's data stream to be sent to the wrong destination OTN node. In order to avoid occurrence of such situation, logical PHY data should be unloaded on the local OTN node, and layer 2 processing chip will perform switch operation on the Ethernet data as payload. After switch, the data are re-packaged and then sent to different destinations.

[00151] With the described processing mechanism, different FlexE client data streams are channelized before entering the shared physical pipe, which eliminates the need for a following Layer 2 switch in OTN equipment, and truly realizes sub-rate function which flexible Ethernet pursues.

[00152] Embodiment 6

[00153] This is another composition of port PHY. In the networking environment of connected routers and OTN devices, as shown in FIG. 7, four Ethernet ports with rates of 100Gb/s are bound to a flexible Ethernet port of a rate of 400Gb/s. The three FlexE client data flows FlexE client 1, FlexE client 2 and FlexE client 3 are set in the router, which are able to pass the flexible Ethernet ports, with a respective rate of 10Gb/s, 150Gb/s, and 20Gb/s. Each FlexE client data stream corresponds to a RS sublayer and a Mil media independent interface in the following manner:

[00154] FlexE client 1 corresponds to one Mil interface.

[00155] FlexE client 2 corresponds to a Mil ground formed by a combination of three 50GMII interfaces (logical Mil interface).

[00156] FlexE client 3 corresponds to a Mil ground formed by a combination of two 10GMII interfaces (logical Mil interface).

[00157] Data within FlexE client 1 data stream form parallel data streams via the Mil interface, relying on 64B/66B encoding mapping and coding techniques to generate 66-bit block, and mapped to two 5G time slots in the first Master Calendar; data within FlexE client 2 data
stream form three groups of parallel data streams via three 50G Mil interfaces, relying on 64B/66B encoding mapping and coding techniques to generate three 66-bit blocks, and mapped to 30 5G time slots in the second and third Master Calendars; data within FlexE client 3 data stream form two groups of parallel data streams via two 10G Mil interfaces, relying on 64B/66B encoding mapping and coding techniques to generate two 66-bit blocks, and mapped to two 5G time slots in the fourth Master Calendar. After the mapping, the data can be sent by FlexE SHIM layer.

[00158] According to the FlexE protocol, the location of start from where 66 bits are formed belongs to the category of FlexE SHEVI layer. Next, the generated 64B/66B data are mapped into a corresponding ODUflex. The ODUflex corresponds to a time slot set assigned to the MAC client in the 20n Master Calendar, and the data inside the ODUflex are put into these time slot sets. The remaining processes are same as the conventional FlexE flexible Ethernet technology.

[00159] Due to the above processing, the FlexE client data streams into corresponding ODUflex correspond to time slot sets assigned to the MAC client in the 20n Master Calendar. These time slot set, ODUflex constitute logic PHY which belongs to the FlexE client data stream, and are channelized. The FlexE client data stream can therefore be used as a whole, and be transmitted to the destination node intact.

[00160] Without the above-described processing flow, in accordance with the existing flexible Ethernet technology, three FlexE client data streams will not be distinguished, and the sent-to logical PHY will carry three FlexE clients' data streams, and the logical PHY will send the data to one destination node. If the above three FlexE clients' data streams are respectively destined to different nodes, then it will cause some FlexE client's data stream to be sent to the wrong destination OTN node. In order to avoid occurrence of such situation, logical PHY data should be unloaded on the local OTN node, and layer 2 processing chip will perform switch operation on the Ethernet data as payload. After switch, the data are re-packaged and then sent to different destinations.

[00161] With the processing mechanism described herein, different FlexE client data streams are channelized before entering the shared physical pipe, which eliminates the need for a following Layer 2 switch in OTN equipment, and truly realizes sub-rate function which flexible Ethernet pursues.

[00162] Embodiment 7
[00163] In embodiment 7, an existing Mil interface is used to construct Mil interfaces with different rates.

[00164] Some embodiments may considers reusing existing interfaces. The new Mil interface, unlike current rate of standard Ethernet, has a rate of 5Gbps, which is constructed by reducing the transmitting and receiving clocks from a clock rate of XGMII interface with 10Gbps to half of the original one.

[00165] The new Mil interface, unlike current rate of standard Ethernet, has a rate of 150Gbps, which is constructed by increasing the clock rate of the transmitting and receiving clocks from a clock rate of XGMII interface with 10Gbps to 1.5 times of the original one;

[00166] The new Mil interface, unlike current rate of standard Ethernet, has a rate of 300Gbps, which is constructed by reducing the clock rate of the transmitting and receiving clocks from a clock rate of CDMII interfaces with 400Gbps to 0.75 times of the original one;

[00167] The new Mil interface, unlike current rate of standard Ethernet, has a rate of 800Gbps, which is constructed by increasing the clock rate of the transmitting and receiving clocks from a clock rate of XGMII interface with 400Gbps to twice of the original one;

[00168] The above various new non-standard-rate Mil interfaces still correspond to their matching RS sublayers;

[00169] From FIG. 12:

[00170] TXD [31:0]: data transmission channel, 32-bit parallel data.

[00171] RXD [31:0]: data receiving channel, 32-bit parallel data.

[00172] TXC [3:0]: transmit channel control signal, when TXC = 0, it indicates the signals transmitted on TXD are data; TXC = 1, it indicates that the signals transmitted on TXD are control characters. TXC [3:0] respectively correspond to TXD [31:24], TXD [23:16], TXD [15:8], and TXD [7:0].

[00173] RXC [3:0]: receiving channel control signal, when RXC = 0, it indicates that the signals transmitted on RXD are data. When RXC = 1, it indicates that the signals transmitted on RXD are control character. RXC [3:0] respectively correspond to RXD [31:24], RXD [23:16], RXD [15:8], RXD [7:0].

[00174] TX_CLK: reference clock for TXD and TXC. The clock frequency is (1/2) * 156.25MHz. Both rising and falling edges of the clock signal sample data. (1/2) * 156.25MHz * 2 * 32 = 5Gbps.
RX_CLK: reference clock for RXD and RXC. The clock frequency is \((1/2) \times 156.25MHz\), Both rising and falling edges of the clock signal sample data.

From FIG. 13:

Due to the impact of electrical characteristics, maximum transmission distance of PCB traces of XGMII interface is only 7cm, and XGMII interfaces have too many line connections, which is inconvenient for practical applications. Therefore, in practical applications, XGMII interfaces are usually replaced with XAUI interfaces. XAUI, namely 10 Gigabit attachment unit interface, 10G attachment unit Interface. XAUI, based on XGMII, realizes expansion of physical distance of XGMII interface, and increases the transmission distance of PCB traces to 50cm, and enables trace on the back side.

The source XGMII divide the transceiving 32 bit width data stream into four separate lane channels, each lane channel corresponding to a byte. After completing 8B/10B encoding by XGXS (XGMII Extender Sublayer), the four lanes correspond to 4 independent channels of XAUI, with a XAUI port rate: \((1/2) \times 2.5Gbps \times 1.25 \times 4 = 6.25Gbps\), which can transmit 5Gbps signals.

With reference to FIG. 30 and FIG. 31, an operational problem associated with transfer of FlexE data over a transport network is described.

FIG. 30 shows an example transport network that is unaware of FlexE carrying FlexE data. Starting from the left side of FIG. 30 to the right side, data traffic from a transmitting FlexE is shown to be sent over four fiber connections through the transport network to the receiving FlexE shim. Data packets from a single Flex client may thus be carried over multiple different physical connections, e.g., optical fibers in FIG. 30. In practice, the physical distance between the transmitting shim and the receiving shim may be hundreds or even thousands of kilometers. As a result, the data packets for the same transmitting FlexE client may experience large differences in data propagation times during transmission from the transmitting shim to the receiving shim. While Ethernet allows for some clock adjustments to remove differential delays or skews, the differential delays experienced in long haul networks may be too large to overcome by the currently prescribed FlexE technologies.

FIG. 31 shows an example of a transport network that is aware of FlexE carrying FlexE data. A similar clock skew problem may exist in when data packets from the same FlexE client at FlexE shim 3102 travel through two different paths (the 150G path, or the 25G/50G path).
to the receiving Flex shim 3104. This clock skew may be due to the differential propagation delay and also due to processing delay through the protocol stack implementation at the shim 3106.

[00182] In order to overcome such clock skew problems, in some embodiments, a timing skew correction mechanism similar to the Precision Time Protocol mechanism (PTP) specified in IEEE 1558, which is incorporated by reference herein, may be utilized. While PTP was specified for improving time synchronization in a local area network with smaller physical reach, e.g., few hundred meters, its basic principles can be applied for achieving timing synchronization in FlexE data transport through long haul networks. The equipment running one of the FlexE shims may be selected or chosen as the timing server, and may provide accurate timing information to the other equipment. The clock synchronization may be performed at a protocol sublayer that is between the FlexE shim and the physical layer, e.g., in the PCS.

[00183] Alternatively, or additionally, in some embodiments, a mechanism similar to synchronous Ethernet (SyncE) may be used for correcting clock skews. In this embodiment, clock information may be passed between phy layers of all receiving and transmitting devices and may be used for correcting clock skews. In some embodiments, a very high precision clock source, e.g., a master clock, may be made available in the network. The precision of this clock source may be of the order of $10^{-11}$ clock inaccuracy.

[00184] In some embodiments, the clock synchronization is achieved by using an operations administration and maintenance (OAM) protocol data unit (PDU) that is identified by a specific Ethernet frame header. The synchronization may be achieved at the FlexE physical layer by processing OAM PDUs after the 64B/66B encoding is performed.

[00185] FIG. 21 illustrates a flowchart for an example method 2100 for communicating data from multiple Ethernet inputs having multiple interface rates to an optical network are disclosed.

The method 2100 includes constructing (2104) one or more elastic/flexible containers operating on data at a media access control (MAC) layer. The elastic/flexible container comprises a variable length data structure and variable logical resources. The method 2100 includes configuring (2106) an output interface having an interface rate equal to a total of the multiple Ethernet interface rates, to carry data in time slots according to a transmission schedule corresponding to the constructed one or more elastic/flexible containers in the MAC layer. The method 2100 includes processing (2108) data packets received from the multiple Ethernet
interfaces through the one or more elastic/flexible containers to generate multiple processed
multiple Ethernet data streams. The method 2100 includes allocating (2110) data packets from
the processed multiple Ethernet data streams according to time slots of the transmission schedule
to generate an output data stream. The method 2100 includes communicating (2112), to an
optical network, a routing policy by which multiple optical data units of the optical network are
to receive the output data stream.

[00186] FIG. 22 illustrates a block diagram for an example apparatus 2200 for communicating
data from multiple Ethernet inputs having multiple interface rates to an optical network are
disclosed. The apparatus 2200 includes a module for constructing (2204) one or more elastic/
flexible containers operating on data at a media access control (MAC) layer, wherein each
elastic/flexible container comprises a variable length data structure and variable logic resources,
a module for configuring (2206) an output interface having an interface rate equal to a total of
the multiple Ethernet interface rates to carry data in time slots according to a transmission
schedule corresponding to the constructed one or more elastic/flexible containers in the MAC
layer, a module for processing (2208) data packets received from the multiple Ethernet interfaces
through the one or more elastic/flexible containers to generate processed multiple Ethernet data
streams, a module for allocating (2210) data packets from the processed multiple Ethernet data
streams according to time slots of the transmission schedule to generate an output data stream,
and a module for communicating (2212), to an optical network, a routing policy by which
multiple optical data units of the optical network are to receive the output data stream.

[00187] FIG. 23 illustrates a flowchart for an example method 2300 for receiving data from an
optical network and transmitting over multiple Ethernet interfaces are disclosed. The method
2300 includes constructing (2304) one or more elastic/flexible container operating on data at a
media access control (MAC) layer, wherein each elastic/flexible container comprises a variable
length data structure and variable logic resources, configuring (2306) an input interface having
an interface rate equal to a total of the multiple Ethernet interface rates, wherein the input
interface is configured to carry data in time slots according to a transmission schedule
corresponding to the constructed one or more elastic/flexible containers in the MAC layer,
receiving (2308), , a routing policy by which multiple optical data units of the optical network
are transmitting data on to the input interface, and selecting (2310) data packets from the input
interface according to the transmission schedule for processing through a network protocol stack
implemented on the network apparatus for transmission to the multiple Ethernet outputs. The method 2300 may be implemented to transfer data from an ingress point of a network to an egress point of the transmission network using the FlexE technology and the flexible/elastic container described herein.

FIG. 24 illustrates a block diagram of an example apparatus 2400 for receiving data from an optical network and transmitting over multiple Ethernet interfaces are disclosed. The apparatus 2400 includes a module for constructing (2404) one or more elastic/flexible container operating on data at a media access control (MAC) layer, wherein each elastic/flexible container comprises a variable length data structure and variable logic resources, a module for configuring (2406) an input interface having an interface rate equal to a total of the multiple Ethernet interface rates, wherein the input interface is configured to carry data in time slots according to a transmission schedule corresponding to the constructed one or more elastic/flexible containers in the MAC layer, a module for receiving (2408), a routing policy by which multiple optical data units of the optical network are transmitting data on to the input interface, and a module for selecting (2410) data packets from the input interface according to the transmission schedule for processing through a network protocol stack implemented on the network apparatus for transmission to the multiple Ethernet outputs.

With reference to methods 2100, 2300 and apparatus 2200, 2400, in some embodiments, the output interface is configured in form of one or multiple ODUflexes. In some embodiments, each elastic/flexible container further comprises at least some of a data cache, a first in first out (FIFO) structure and a logic circuit. In some embodiments, the method 2100 or 2300 may include constructing the one or more elastic/flexible containers such that there is one elastic/flexible container corresponding to each of the multiple Ethernet inputs. In some embodiments, the output interface rate is a linear combination of submultiples of at least two standard media independent interface rates.

In some embodiments, the standard media independent interface rates include 5 Gbps, 10 Gbps, 25 Gbps, 40 Gbps, 50 Gbps, 100 Gbps, 200 Gbps, 300 Gbps, 400 Gbps and IT Gbps. In some embodiments, in method 2100, achieving a submultiple rate for a given standard media independent interface rate by dividing a clock for the given standard media independent interface rate by an integer factor. In some embodiments, the linear combination may be obtained by increasing clock rate of at least one standard media independent interface by an integer factor.
FIG. 26 shows a flowchart example of a method 2600 of transferring data from one or more input data streams to a flexible Ethernet (FlexE) shim layer for transmission. The method may be implemented in a network processor or another network apparatus such as a router or a switch that is used to transmit data to a transport network at an ingress point.

The method 2600 includes, at 2602, operating flexible containers to receive the one or more input data streams to generate, from each flexible container, one or more output streams.

The method 2600 includes, at 2604, processing one or more output streams from each flexible container in one or more reconciliation sublayers to generate 64-bit data signals.

The method 2600 includes, at 2606, providing, over a modified industry-standard interface, the 64-bit data signals to a FlexE shim layer in which the 64-bit data signals are each encoded using a 64B/66B encoding to generate logical serial data streams of 64B/66B blocks representing FlexE clients which are mapped to allocated timing slots for transmission. In some embodiments, the master calendar mechanism, e.g., as described in the FlexE specification document, may be used for the timing slot allocation.

The modified industry-standard interface may include one or more industry-standard interfaces having a same standard rate, or one or more Mil interface having different standard rates, or one or more industry-standard interfaces having a non-standard rate;

During implementation of method 2600, each flexible container may correspond to a FlexE client and the processing rate of each flexible container may match a rate of a logical data stream for a corresponding FlexE client.

In some embodiments, an apparatus for transferring data from one or more input data streams to a flexible Ethernet (FlexE) shim layer for transmission includes a number of flexible containers for receiving the one or more input data streams to generate, from each flexible container, one or more output streams, one or more reconciliation sublayer modules for processing the one or more output streams from each flexible container to generate 64-bit data signals, and a modified industry-standard interface for providing the 64-bit data signals to a FlexE shim layer in which the 64-bit data signals are each encoded using a 64B/66B encoding to generate logical serial data streams of 64B/66B blocks representing FlexE clients which are mapped to allocated timing slots for transmission. The modified industry-standard interface comprises one or more industry-standard interfaces having a same standard rate, or one or more Mil interface having different standard rates, or one or more industry-standard interfaces having
a non-standard rate. Each flexible container corresponds to a FlexE client and the processing rate of each flexible container matches a rate of a logical data stream for a corresponding FlexE client. In some embodiments, the method 2600 further includes performing, by the FlexE shim layer, idle insert/delete processing on the 64B/66B blocks according to IEEE 802.3 standard. In some embodiments, the 2600 may also include inserting, prior to the transmission, timing information for clock synchronization. The timing information is inserted according to the Precision Time Protocol of IEEE-1558 or the Synchronous Ethernet protocol. In some embodiments, the clock synchronization information is inserted in an operations administration and maintenance (OAM) protocol data unit (PDU).

FIG. 27 shows a flowchart example of a method 2700 transferring data from a FlexE shim layer to one or more output data streams during reception of the data. The input data streams may a transport network ingress point, while the output data stream may be at a transport network egress point.

The method 2700 includes, at 2702, providing, over a modified industry standard interface, 64-bit data signals from a FlexE shim layer that decodes, using a 64B/66B decoding, logical serial data streams of 64B/66B blocks representing FlexE clients which are mapped to allocated timing slots in the data. In some embodiments, the data may have been allocated to timing slots using the master calendar mechanism, e.g., as described in the FlexE specification document.

The method 2700 includes, at 2704, processing the 64-bit data signals through one or more reconciliation sublayers to generate one or more data stream inputs to flexible containers.

The method 2700 includes, at 2706, operating the flexible containers to output the one or more output data streams from the received one or more input streams.

In some embodiments, an apparatus for transferring data from a flexible Ethernet (FlexE) shim layer to one or more output data streams during reception of the data includes a modified industry standard interface that provides 64-bit data signals from a FlexE shim layer that decodes, using a 64B/66B decoding, logical serial data streams of 64B/66B blocks representing FlexE clients which are mapped to allocated timing slots in the data, one or more reconciliation sublayers to process the 64-bit data signals to generate one or more data stream inputs to flexible containers and flexible containers to output the one or more output data streams from the received one or more input streams. In some embodiments, the method 2700 includes
performing clock synchronization using the above-described techniques for achieving clock
synchronization.

[00203] With respect to the methods 2600, 2700 and apparatus for transferring data described
above, the non-standard rate may be constructed by reducing a rate of a standard Mil.

Alternatively, or in addition, the non-standard rate may be constructed by increasing a rate of a
standard MIL. The non-standard rate may be constructed by changing clock rate of a standard
Mil or by changing data transfer rate of a SerDes interface or by using a 4-pulse amplitude
modulation (PAM-4) data signal modulation technique, e.g., in place of the NRZ modulation. In
some embodiments, the non-standard rate is constructed by combining two or more standard Mil
rates to form a logic Mil interface. In some embodiments, the non-standard rate is constructed by
combining two or more different Mil interfaces wherein one of the different Mil interface rates
is a rate that is adjusted from a standard rate.

[00204] In various embodiments, the standard rate corresponds to a rate of a 100 Gigabit Mil
interface (CGMII), a 10 Gigabit Mil (XGMII), a 10 Gigabit attachment unit interface (XAUI) or
a 400 Gigabit Mil interface (CDMII), or another rate, e.g., a reduced media independent
interface rate (RMII), an RGMII, a quad serial GMII, a serial Mil (SMII), and so on.

[00205] Furthermore with reference to the apparatus for transferring and methods 2600 and
2700 described above, the assignment of which input and output streams are to be processed by
which flexible container may be performed by a stream distributor module. In some
embodiments, the resources allocated to each flexible container, e.g., hardware resources, may be
proportional to the bandwidth of the corresponding input data stream processed by the flexible
container. The hardware resources may, e.g., be a cache space or a buffer. In some embodiments,
e.g., as shown in FIGs. 9, 10 and 11, some flexible containers may receive from, or provide to,
the data over multiple data streams and/or Mil streams.

[00206] In some embodiments, the rate of operation of the flexible/elastic container matches
the data rate of the targeted FlexE client. In some embodiments, method 2600 further includes
encoding the data into a logical series stream as 64B/66B blocks and performing idle
insert/delete processing on the logical series stream according to IEEE 802.3 standard. An
example of this processing is described with reference to FIG. 28.

[00207] In another example aspect, a technique for transmitting and receiving data using a
FlexE shim includes an apparatus for transferring data from a first group of one or more input
data streams to a flexible Ethernet (FlexE) shim layer for transmission, and transferring, from the FlexE shim layer to a first group of one or more output data streams for reception. The apparatus includes a number of flexible containers for receiving the first group one or more input data streams to generate, from each flexible container, a second group of one or more output streams, one or more reconciliation sublayer modules for processing the second group of one or more output streams from each flexible container to generate a first group of 64-bit data signals, and a modified industry-standard interface for providing the first group of 64-bit data signals to a FlexE shim layer in which the first group of 64-bit data signals are each encoded using a 64B/66B encoding to generate a first group of logical serial data streams of 64B/66B blocks representing FlexE clients which are mapped to a first group allocated timing slots for transmission. The one or more reconciliation sublayers further process the second group of 64-bit data signals to generate a second group of one or more data stream inputs to flexible containers. The flexible containers further outputs the first group of one or more output data streams from the received second group of one or more input streams. The modified industry-standard interface comprises one or more industry-standard interfaces having a same standard rate, or one or more Mil interface having different standard rates, or one or more industry-standard interfaces having a non-standard rate. In the technique, each flexible container corresponds to a FlexE client and a processing rate of each flexible container matches a rate of a logical data stream for a corresponding FlexE client. Examples of transmitting embodiments are shown and described with respect to FIGs. 9, 10, 11 and 19, and examples of receive side operations are described with respect to FIG. 20.

[00208] It will be appreciated that several techniques have been disclosed for facilitating transfer of flexible Ethernet traffic rates to and from optical networks in which flexible data rates can be used for data communication. It will further be appreciated that the disclosed techniques may be able to be combined with an industry standard FlexE shim layer implementation to allow the use of multiple FlexE clients and mapping/de-mapping of data generated to and from the clients for facilitating transportation and reception of FlexE data streams at transmission network ingress points or egress points.

[00209] The disclosed and other embodiments and the functional operations and modules described in this document can be implemented in digital electronic circuitry, or in computer software, firmware, or hardware, including the structures disclosed in this document and their
structural equivalents, or in combinations of one or more of them. The disclosed and other embodiments can be implemented as one or more computer program products, i.e., one or more modules of computer program instructions encoded on a computer readable medium for execution by, or to control the operation of, data processing apparatus. The computer readable medium can be a machine-readable storage device, a machine-readable storage substrate, a memory device, a composition of matter effecting a machine-readable propagated signal, or a combination of one or more them. The term "data processing apparatus" encompasses all apparatus, devices, and machines for processing data, including by way of example a programmable processor, a computer, or multiple processors or computers. The apparatus can include, in addition to hardware, code that creates an execution environment for the computer program in question, e.g., code that constitutes processor firmware, a protocol stack, a database management system, an operating system, or a combination of one or more of them. A propagated signal is an artificially generated signal, e.g., a machine-generated electrical, optical, or electromagnetic signal, that is generated to encode information for transmission to suitable receiver apparatus.

[00210] A computer program (also known as a program, software, software application, script, or code) can be written in any form of programming language, including compiled or interpreted languages, and it can be deployed in any form, including as a stand alone program or as a module, component, subroutine, or other unit suitable for use in a computing environment. A computer program does not necessarily correspond to a file in a file system. A program can be stored in a portion of a file that holds other programs or data (e.g., one or more scripts stored in a markup language document), in a single file dedicated to the program in question, or in multiple coordinated files (e.g., files that store one or more modules, sub programs, or portions of code). A computer program can be deployed to be executed on one computer or on multiple computers that are located at one site or distributed across multiple sites and interconnected by a communication network.

[00211] The processes and logic flows described in this document can be performed by one or more programmable processors executing one or more computer programs to perform functions by operating on input data and generating output. The processes and logic flows can also be performed by, and apparatus can also be implemented as, special purpose logic circuitry, e.g., an FPGA (field programmable gate array) or an ASIC (application specific integrated circuit).
Processors suitable for the execution of a computer program include, by way of example, both general and special purpose microprocessors, and any one or more processors of any kind of digital computer. Generally, a processor will receive instructions and data from a read only memory or a random access memory or both. The essential elements of a computer are a processor for performing instructions and one or more memory devices for storing instructions and data. Generally, a computer will also include, or be operatively coupled to receive data from or transfer data to, or both, one or more mass storage devices for storing data, e.g., magnetic, magneto optical disks, or optical disks. However, a computer need not have such devices. Computer readable media suitable for storing computer program instructions and data include all forms of non volatile memory, media and memory devices, including by way of example semiconductor memory devices, e.g., EPROM, EEPROM, and flash memory devices; magnetic disks, e.g., internal hard disks or removable disks; magneto optical disks; and CD ROM and DVD-ROM disks. The processor and the memory can be supplemented by, or incorporated in, special purpose logic circuitry.

While this document contains many specifics, these should not be construed as limitations on the scope of an invention that is claimed or of what may be claimed, but rather as descriptions of features specific to particular embodiments. Certain features that are described in this document in the context of separate embodiments can also be implemented in combination in a single embodiment. Conversely, various features that are described in the context of a single embodiment can also be implemented in multiple embodiments separately or in any suitable sub-combination. Moreover, although features may be described above as acting in certain combinations and even initially claimed as such, one or more features from a claimed combination can in some cases be excised from the combination, and the claimed combination may be directed to a sub-combination or a variation of a sub-combination. Similarly, while operations are depicted in the drawings in a particular order, this should not be understood as requiring that such operations be performed in the particular order shown or in sequential order, or that all illustrated operations be performed, to achieve desirable results.

Only a few examples and implementations are disclosed. Variations, modifications, and enhancements to the described examples and implementations and other implementations can be made based on what is disclosed.
What is claimed is:

1. A method of transferring data from one or more input data streams to a flexible Ethernet (FlexE) shim layer for transmission, comprising:
   operating flexible containers to receive the one or more input data streams to generate, from each flexible container, one or more output streams;
   processing one or more output streams from each flexible container in one or more reconciliation sublayers to generate data signals; and
   providing, over a modified industry-standard interface, the data signals to a FlexE shim layer in which the data signals are encoded to generate logical serial data streams of encoded blocks representing FlexE clients which are mapped to allocated timing slots for transmission;
   wherein each flexible container and the processing in the one or more reconciliation sublayers enable the modified industry-standard interface to include one or more interfaces having respective rates to accommodate for transmission via the FlexE shim layer.

2. The method of claim 1, wherein the one or more reconciliation sublayers generate 64-bit data signals; and
   the 64-bit data signals are each encoded using a 64B/66B encoding to generate logical serial data streams of 64B/66B blocks representing FlexE clients which are mapped to allocated timing slots for transmission.

3. The method of claim 2, wherein the modified industry-standard interface comprises one or more industry-standard interfaces having a same standard rate, or one or more Mil interface having different standard rates, or one or more industry-standard interfaces having a non-standard rate.

4. The method of claim 1 through 3, wherein each flexible container corresponds to a FlexE client.
5. The method of claim 2 or 3, wherein a processing rate of each flexible container matches a rate of a logical data stream for a corresponding FlexE client before the 64B/66B encoding.

6. The method of claim 3, wherein the non-standard rate is constructed by reducing a rate of a standard media independent interface (Mil).

7. The method of claim 3, wherein the non-standard rate is constructed by increasing a rate of a standard media independent interface (Mil).

8. The method of claim 3, wherein the non-standard rate is constructed by changing clock rate of a standard media independent interface (Mil) or by changing data transfer rate of a SerDes interface or by using a 4-pulse amplitude modulation (PAM-4) data signal modulation technique.

9. The method of claim 3, wherein the non-standard rate is constructed by combining two or more standard media independent interface (Mil) rates to form a logic Mil interface.

10. The method of claim 3, wherein the non-standard rate is constructed by combining two or more different media independent interface (Mil) interfaces wherein one of the different Mil interface rates is a rate that is adjusted from a standard rate.

11. The method of claim 10, wherein the standard rate corresponds to a rate of a 100 Gigabit Mil interface (CGMII), a 10 Gigabit Mil (XGMII), a 10 Gigabit attachment unit interface (XAUI) or a 400 Gigabit Mil interface (CDMII), or a different rate.

12. The method of claim 1 through 3, wherein the operating the flexible containers to receive the one or more input data stream and generate corresponding output streams, includes assigning an input data stream to a flexible container using a stream distributer module.

13. The method of claim 1, further including:
allocating resources to each flexible container proportional to a bandwidth of a
 corresponding input data stream processed by that flexible container.

14. The method of claim 13, wherein the resources include a cache space.

15. The method of claim 1, further including:
operating at least one of the flexible containers to produce two or more output streams.

16. The method of claim 1, further including:
performing, by the FlexE shim layer, idle insert/delete processing on the 64B/66B blocks
according to IEEE 802.3 standard.

17. The method of claim 1, further including:
inserting, prior to the transmission, timing information for clock synchronization.

18. The method of claim 17, wherein the inserting the timing information includes inserting
the timing information in an operations administration and maintenance (OAM) protocol data
unit (PDU).

19. The method of claim 17, wherein the timing information is inserted according to the
Precision Time Protocol of IEEE-1558 or the Synchronous Ethernet protocol.

20. An apparatus for transferring data from one or more input data streams to a flexible
Ethernet (FlexE) shim layer for transmission, comprising:
a number of flexible containers for receiving the one or more input data streams to
generate, from each flexible container, one or more output streams;
one or more reconciliation sublayer modules for processing the one or more output steams
from each flexible container to generate 64-bit data signals; and
a modified industry-standard interface for providing the 64-bit data signals to a FlexE shim
layer in which the 64-bit data signals are each encoded using a 64B/66B encoding to generate
logical serial data streams of 64B/66B blocks representing FlexE clients which are mapped to allocated timing slots for transmission;

wherein the modified industry-standard interface comprises one or more industry-standard interfaces having a same standard rate, or one or more Mil interface having different standard rates, or one or more industry-standard interfaces having a non-standard rate;

wherein each flexible container corresponds to a FlexE client; and

wherein a processing rate of each flexible container matches a rate of a logical data stream for a corresponding FlexE client before the 64B/66B encoding.

21. The apparatus of claim 20, wherein the non-standard rate is constructed by reducing a rate of a standard media independent interface (Mil).

22. The apparatus of claim 20, wherein the non-standard rate is constructed by increasing a rate of a standard media independent interface (Mil).

23. The apparatus of claim 20, wherein the non-standard rate is constructed by changing clock rate of a standard media independent interface (Mil) or by changing data transfer rate of a SerDes interface or by using a 4-pulse amplitude modulation (PAM-4) data signal modulation technique.

24. The apparatus of claim 20, wherein the non-standard rate is constructed by combining two or more standard media independent interface (Mil) rates to form a logic Mil interface.

25. The apparatus of claim 20, wherein the non-standard rate is constructed by combining two or more different media independent interface (Mil) interfaces wherein one of the different Mil interface rates is a rate that is adjusted from a standard rate.

26. The apparatus of claim 25, wherein the standard rate corresponds to a rate of a 100 Gigabit Mil interface (CGMII), a 10 Gigabit Mil (XGMII), a 10 Gigabit attachment unit interface (XAUI) or a 400 Gigabit Mil interface (CDMII), or another rate.
27. The apparatus of claim 20, further including:
   a stream distributor module that assigns an input data stream to the flexible containers.

28. The apparatus of claim 20, wherein each flexible container includes:
   hardware resources proportional to a bandwidth of a corresponding input data stream processed by the flexible container.

29. The apparatus of claim 28, wherein the hardware resources include a cache space.

30. The apparatus of claim 20, wherein at least one of the flexible containers produces two or more output streams.

31. The apparatus of claim 20, wherein the FlexE shim layer performs idle insert/delete processing on the 64B/66B blocks according to IEEE 802.3 standard.

32. The apparatus of claim 20, further including:
   a module for inserting, prior to the transmission, timing information for clock synchronization.

33. The apparatus of claim 32, wherein the timing information is inserted according to the Precision Time Protocol of IEEE-1558 or the Synchronous Ethernet protocol.

34. The apparatus of claim 32, wherein the timing information includes the timing information in an operations administration and maintenance (OAM) protocol data unit (PDU).

35. A method of transferring data from a flexible Ethernet (FlexE) shim layer to one or more output data streams during reception of the data, comprising:
   providing, over a modified industry standard interface, 64-bit data signals from a FlexE shim layer that decodes, using a 64B/66B decoding, logical serial data streams of 64B/66B blocks representing FlexE clients which are mapped to allocated timing slots in the data;
processing the 64-bit data signals through one or more reconciliation sublayers to generate one or more data stream inputs to flexible containers; and operating the flexible containers to output the one or more output data streams from the received one or more input streams;

wherein the modified industry-standard interface comprises one or more industry-standard interfaces having a same standard rate, or one or more Mil interface having different standard rates, or one or more industry-standard interfaces having a non-standard rate;

wherein each flexible container corresponds to a FlexE client; and wherein a processing rate of each flexible container matches a rate of a logical data stream for a corresponding FlexE client.

36. The method of claim 35, wherein the non-standard rate is constructed by reducing a rate of a standard media independent interface (Mil).

37. The method of claim 35, wherein the non-standard rate is constructed by increasing a rate of a standard media independent interface (Mil).

38. The method of claim 35, wherein the non-standard rate is constructed by changing clock rate of a standard media independent interface (Mil) or by changing data transfer rate of a SerDes interface or by using a 4-pulse amplitude modulation (PAM-4) data signal modulation technique.

39. The method of claim 35, wherein the non-standard rate is constructed by combining two or more standard media independent interface (Mil) rates to form a logic Mil interface.

40. The method of claim 35, wherein the non-standard rate is constructed by combining two or more different media independent interface (Mil) interfaces wherein one of the different Mil interface rates is a rate that is adjusted from a standard rate.
41. The method of claim 40, wherein the standard rate corresponds to a rate of a 100 Gigabit
Mil interface (CGMII), a 10 Gigabit Mil (XGMII), a 10 Gigabit attachment unit interface
(XAUI) or a 400 Gigabit Mil interface (CDMII), or a different rate.

42. The method of claim 35, wherein the operating the flexible containers to receive the one
or more input data stream and generate corresponding output streams, includes assigning an
input data stream to a flexible container using a stream distributor module.

43. The method of claim 35, further including:
allocating resources to each flexible container proportional to a bandwidth of a
 corresponding input data stream processed by that flexible container.

44. The method of claim 43, wherein the resources include a cache space.

45. The method of claim 35, further including:
operating at least one of the flexible containers to receive two or more input streams.

46. The method of claim 35, further including:
performing, by the FlexE shim layer, idle insert/delete processing on the 64B/66B blocks
according to IEEE 802.3 standard.

47. The method of claim 35, further including:
receiving, prior to the transmission, timing information and performing clock
synchronization using the timing information.

48. The method of claim 47, wherein the timing information conforms to the Precision Time
Protocol of IEEE-1558 or the Synchronous Ethernet protocol.

49. The method of claim 47, wherein the timing information includes timing information in
an operations administration and maintenance (OAM) protocol data unit (PDU).
50. An apparatus for transferring data from a flexible Ethernet (FlexE) shim layer to one or more output data streams during reception of the data, comprising:
   a modified industry standard interface that provides 64-bit data signals from a FlexE shim layer that decodes, using a 64B/66B decoding, logical serial data streams of 64B/66B blocks representing FlexE clients which are mapped to allocated timing slots in the data;
   one or more reconciliation sublayers to process the 64-bit data signals to generate one or more data stream inputs to flexible containers; and
   flexible containers to output the one or more output data streams from the received one or more input streams;
   wherein the modified industry-standard interface comprises one or more industry-standard interfaces having a same standard rate, or one or more Mil interface having different standard rates, or one or more industry-standard interfaces having a non-standard rate;
   wherein each flexible container corresponds to a FlexE client; and
   wherein a processing rate of each flexible container matches a rate of a logical data stream for a corresponding FlexE client.

51. The apparatus of claim 50, wherein the non-standard rate is constructed by reducing a rate of a standard media independent interface (Mil).

52. The apparatus of claim 50, wherein the non-standard rate is constructed by increasing a rate of a standard media independent interface (Mil).

53. The apparatus of claim 50, wherein the non-standard rate is constructed by changing clock rate of a standard media independent interface (Mil) or by changing data transfer rate of a SerDes interface or by using a 4-pulse amplitude modulation (PAM-4) data signal modulation technique.

54. The apparatus of claim 50, wherein the non-standard rate is constructed by combining two or more standard media independent interface (Mil) rates to form a logic Mil interface.
55. The apparatus of claim 50, wherein the non-standard rate is constructed by combining two or more different media independent interface (Mil) interfaces wherein one of the different Mil interface rates is a rate that is adjusted from a standard rate.

56. The apparatus of claim 55, wherein the standard rate corresponds to a rate of a 100 Gigabit Mil interface (CGMII), a 10 Gigabit Mil (XGMII), a 10 Gigabit attachment unit interface (XAUI) or a 400 Gigabit Mil interface (CDMII), or a different rate.

57. The apparatus of claim 50, further including a stream distributor that assigns an input data stream to a flexible container.

58. The apparatus of claim 50, wherein each flexible container includes hardware resources proportional to a bandwidth of a corresponding input data stream processed by that flexible container.

59. The apparatus of claim 58, wherein the hardware resources include a cache space.

60. The apparatus of claim 50, wherein at least one of the flexible containers receives two or more input streams.

61. The apparatus of claim 50, wherein the FlexE shim layer further performs idle insert/delete processing on the 64B/66B blocks according to IEEE 802.3 standard.

62. The apparatus of claim 50, further including:

25 a module for receiving, prior to the transmission, timing information and performing clock synchronization using the timing information.

63. The apparatus of claim 62, wherein the timing information conforms to the Precision Time Protocol of IEEE-1558 or the Synchronous Ethernet protocol.
64. An apparatus for transferring data from a first group of one or more input data streams to a flexible Ethernet (FlexE) shim layer for transmission, and transferring, from the FlexE shim layer to a first group of one or more output data streams, comprising:
	a number of flexible containers for receiving the first group one or more input data streams to generate, from each flexible container, a second group of one or more output streams;
	one or more reconciliation sublayer modules for processing the second group of one or more output streams from each flexible container to generate a first group of 64-bit data signals; and
	a modified industry-standard interface for providing the first group of 64-bit data signals to a FlexE shim layer in which the first group of 64-bit data signals are each encoded using a 64B/66B encoding to generate a first group of logical serial data streams of 64B/66B blocks representing FlexE clients which are mapped to a first group allocated timing slots for transmission;
	the modified industry standard interface further providing a second group of 64-bit data signals from the FlexE shim layer that decodes, using a 64B/66B decoding, a second group of logical serial data streams of 64B/66B blocks representing FlexE clients which are mapped to a second group of allocated timing slots in the data;
	the one or more reconciliation sublayers further processing the second group of 64-bit data signals to generate a second group of one or more data stream inputs to flexible containers; and
	the flexible containers further outputting the first group of one or more output data streams from the received second group of one or more input streams;
	wherein the modified industry-standard interface comprises one or more industry-standard interfaces having a same standard rate, or one or more Mil interface having different standard rates, or one or more industry-standard interfaces having a non-standard rate;
	wherein each flexible container corresponds to a FlexE client; and
	wherein a processing rate of each flexible container matches a rate of a logical data stream for a corresponding FlexE client.

65. The apparatus of claim 64, wherein the non-standard rate is constructed by reducing a rate of a standard media independent interface (Mil).
66. The apparatus of claim 64, wherein the non-standard rate is constructed by increasing a rate of a standard media independent interface (Mil).

67. The apparatus of claim 64, wherein the non-standard rate is constructed by changing clock rate of a standard media independent interface (Mil) or by changing data transfer rate of a SerDes interface or by using a 4-pulse amplitude modulation (PAM-4) data signal modulation technique.

68. The apparatus of claim 64, wherein the non-standard rate is constructed by combining two or more standard media independent interface (Mil) rates to form a logic Mil interface.

69. The apparatus of claim 64, wherein the non-standard rate is constructed by combining two or more different media independent interface (Mil) interfaces wherein one of the different Mil interface rates is a rate that is adjusted from a standard rate.

70. A method of communicating data from multiple Ethernet inputs having multiple Ethernet interfaces at respective interface rates to an optical network, implemented at a network apparatus, comprising:

   constructing one or more elastic/flexible containers operating on data at a media access control (MAC) layer, wherein each elastic/flexible container comprises a variable length data structure and variable logic resources;

   configuring an output interface having an interface rate equal to a total of the multiple Ethernet interface rates to carry data in time slots according to a transmission schedule corresponding to the constructed one or more elastic/flexible containers in the MAC layer;

   processing data packets received from the multiple Ethernet interfaces through the one or more elastic/flexible containers to generate processed multiple Ethernet data streams;

   allocating data packets from the processed multiple Ethernet data streams according to time slots of the transmission schedule to generate an output data stream.
71. The method of claim 70, wherein the output interface comprises one or multiple flexible optical data units (ODUflexes).

72. The method of claim 70, wherein each elastic/flexible container further comprises at least some of a data cache, a first in first out (FIFO) structure and a logic circuit.

73. The method of claim 70, further including:
   constructing the one or more elastic/flexible containers such that there is one elastic/flexible container corresponding to each of the multiple Ethernet inputs.

74. The method of claim 70, wherein the output interface rate is a combination of submultiples of at least two standard media independent interface rates.

75. The method of claim 74, further comprising:
   achieving a submultiple rate for a given standard media independent interface rate by dividing a clock for the given standard media independent interface rate by an integer factor.

76. The method of claim 74, wherein the linear combination is obtained by:
   increasing clock rate of at least one standard media independent interface by an integer factor.
FIG. 1

FIG. 2
802.3 Architecture

Clause 4
- MAC
- RS Layer

Clause 8.1
- Same function (can be variable rate)

Clause 8.2
- PCS
- 64b/66b
- Scramble
- Lane Distribution
- AM

Clause 9.1
- FEC

Clause 8.3
- PMD

Various Clauses

Insert FlexE

- MAC
- RS Layer

- Flex E SHIM

- 64b/66b
- TDM Framing
- Distribution
- Frame Header

- 64b/66b
- Scramble
- Lane Distribution
- AM

- FEC

- PMD

FIG. 3
FIG. 9
FIG. 10
FIG. 11
<table>
<thead>
<tr>
<th>CRAM</th>
<th>PRAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Page 0 Free</td>
</tr>
<tr>
<td>1</td>
<td>Page 1 Free</td>
</tr>
<tr>
<td>3</td>
<td>Page 2 Free</td>
</tr>
<tr>
<td>3</td>
<td>Page 3 Free</td>
</tr>
<tr>
<td>4</td>
<td>Page 4 Free</td>
</tr>
<tr>
<td>5</td>
<td>Page 5 Free</td>
</tr>
<tr>
<td>6</td>
<td>Page 6 Free</td>
</tr>
<tr>
<td>7</td>
<td>Page 7 Free</td>
</tr>
<tr>
<td>719</td>
<td>Page 719 Free</td>
</tr>
</tbody>
</table>

**FIG. 17**
<table>
<thead>
<tr>
<th>CRAM</th>
<th>PRAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Page 0 Free</td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Page 1 Free</td>
</tr>
<tr>
<td>XXX</td>
<td>Page 2 Free</td>
</tr>
<tr>
<td>5</td>
<td>Page 3 Free</td>
</tr>
<tr>
<td>XXX</td>
<td>Page 4 Free</td>
</tr>
<tr>
<td>1</td>
<td>Page 5 Free</td>
</tr>
<tr>
<td>XXX</td>
<td>Page 6 Free</td>
</tr>
<tr>
<td>719</td>
<td>Page 7 Free</td>
</tr>
<tr>
<td>719</td>
<td>Page 719 Free</td>
</tr>
<tr>
<td>6</td>
<td></td>
</tr>
</tbody>
</table>

**FIG. 18**
FIG. 19
FIG. 20

Constructing an elastic/flexible container 2104
Configuring an output interface 2106
Processing data packets 2108
Allocating data packets to time slots of an output data stream 2110
Communicating a routing policy 2112

FIG. 21

Module for constructing an elastic/flexible container 2204
Module for configuring an output interface 2206
Module for processing data packets 2208
Module for allocating data packets to time slots of an output data stream 2210
Module for communicating a routing policy 2212

FIG. 22
FIG. 23

By Section

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Calendar AClear Slot 0</td>
<td>Calendar AClear Slot 1</td>
<td>Calendar AClear Slot 2</td>
</tr>
<tr>
<td>1</td>
<td>Calendar BClear Slot 0</td>
<td>Calendar BClear Slot 1</td>
<td>Calendar BClear Slot 2</td>
</tr>
<tr>
<td>2</td>
<td>Calendar CClear Slot 0</td>
<td>Calendar CClear Slot 1</td>
<td>Calendar CClear Slot 2</td>
</tr>
<tr>
<td>3</td>
<td>Calendar DClear Slot 0</td>
<td>Calendar DClear Slot 1</td>
<td>Calendar DClear Slot 2</td>
</tr>
</tbody>
</table>

Management Channel (8 octets)

FIG. 24

By Section

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>PFM MAP Request 0-12</td>
<td>PFM MAP Request 13-25</td>
<td>PFM MAP Request 26-39</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>PFM MAP Request 40-53</td>
<td>PFM MAP Request 54-67</td>
<td>PFM MAP Request 68-81</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>PFM MAP Request 82-95</td>
<td>PFM MAP Request 96-109</td>
<td>PFM MAP Request 110-123</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>PFM MAP Request 124-137</td>
<td>PFM MAP Request 138-151</td>
<td>PFM MAP Request 152-165</td>
<td></td>
</tr>
</tbody>
</table>

48 Overhead repetitions

FIG. 25
operating flexible containers to receive the one or more input data streams to generate, from each flexible container, one or more output streams

processing one or more output streams from each flexible container in one or more reconciliation sublayers to generate 64-bit data signals

providing, over a modified industry-standard interface, the 64-bit data signals to a FlexE shim layer in which the 64-bit data signals are each encoded using a 64B/66B encoding to generate logical serial data streams of 64B/66B blocks representing FlexE clients which are mapped to allocated timing slots for transmission

FIG. 26
providing, over a modified industry standard interface, 64-bit data signals from a FlexE shim layer that decodes, using a 64B/66B decoding, logical serial data streams of 64B/66B blocks representing FlexE clients which are mapped to allocated timing slots in the data

processing the 64-bit data signals through one or more reconciliation sublayers to generate one or more data stream inputs to flexible containers

operating the flexible containers to output the one or more output data streams from the received one or more input streams

FIG. 27
All PHYs of the FlexE group are carried independently, but over the same fiber route, over the transport network. Deskew across the transport network is performed in the FlexE shim.

Since the FlexE server is terminated before crossing the transport network, the total skew is comparable to the Ethernet PCS lane skew.

FIG. 31
INTERNATIONAL SEARCH REPORT

A. CLASSIFICATION OF SUBJECT MATTER
   H04L 12/24(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
   H04L; H04B

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
   WPI, EPDOC, CNPAT, CNKI: flexible, ethernet, flexe, container, buffer, shim, reconciliation, Mil, interface, RS, rate, speed, client, encode, 64B, time, slot, IEEE802.3

C. DOCUMENTS CONSIDERED TO BE RELEVANT

<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>CN 104426651 A (HUAWEI TECHNOLOGIES CO., LTD.) 18 March 2015 (2015-03-18) the whole document</td>
<td>1-76</td>
</tr>
<tr>
<td>A</td>
<td>WO 2015027126 A1 (NEC LABORATORIES AMERICA, INC.) 26 February 2015 (2015-02-26) the whole document</td>
<td>1-76</td>
</tr>
<tr>
<td>A</td>
<td>EP 1124355 A2 (NORTEL NETWORKS CORPORATION) 16 August 2001 (2001-08-16) the whole document</td>
<td>1-76</td>
</tr>
<tr>
<td>A</td>
<td>WO 2014153784 A1 (HUAWEI TECHNOLOGIES CO., LTD.) 02 October 2014 (2014-10-02) the whole document</td>
<td>1-76</td>
</tr>
</tbody>
</table>

Further documents are listed in the continuation of Box C. See patent family annex.

* Special categories of cited documents:
   "A" document defining the general state of the art which is not considered to be of particular relevance
   "E" earlier application or patent but published on or after the international filing date
   "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
   "O" document referring to an oral disclosure, use, exhibition or other means
   "P" document published prior to the international filing date but later than the priority date claimed
   "Q" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
   "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
   "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
   "&" document member of the same patent family

Date of the actual completion of the international search 30 June 2016

Date of mailing of the international search report 22 July 2016

Name and mailing address of the ISA/CN

STATE INTELLECTUAL PROPERTY OFFICE OF THE P.R.CHINA
6, Xitucheng Rd., Jimen Bridge, Haidian District, Beijing 100088, China

Facsimile No. (86-10)62019451

Authorized officer BAI Fangfang

Telephone No. (86-10)62413406

Form PCT/ISA/210 (second sheet) (July 2009)
### C. DOCUMENTS CONSIDERED TO BE RELEVANT

<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>US 9148382 B2 (CIENA CORPORATION) 29 September 2015 (2015-09-29) the whole document</td>
<td>1-76</td>
</tr>
<tr>
<td>Patent document cited in search report</td>
<td>Publication date (day/month/year)</td>
<td>Patent family member(s)</td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>----------------------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DE 60133330 T2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>JP 2001274824 A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EP 1124355 A3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>JP 4741090 B2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DE 60133330 D1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>US 2001014104 A1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>US 6944163 B2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>US 2016013922 A1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EP 2950496 A4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CN 104185976 A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>US 9148382 B2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>US 2013208595 A1</td>
</tr>
</tbody>
</table>

Form PCT/ISA/210 (patent family annex) (July 2009)