Interlaken Look-Aside Protocol Definition

10 December 2008

Revision 1.1

Terms and Conditions

This document has been developed with input from a variety of companies, including members of the Interlaken Alliance, all of which have released their respective rights to information contained in this document for the sole and express purpose of implementation of this document and to encourage others to adopt this interface as an industry standard.

Any company or individual wishing to use all or a part of this document may do so only if they relinquish their proprietary rights to information contained or referenced herein. You agree not to enforce those intellectual property rights against any other party implementing or advocating the implementation of this document.

YOU ACKNOWLEDGE THAT THIS DOCUMENT IS PROVIDED “AS IS” AND WITHOUT WARRANTY OF ANY KIND. FOR THE SAKE OF CLARITY, ALL PARTIES USING OR ADVOCATING THE USE OF THIS DOCUMENT DISCLAIM ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, INCLUDING WITHOUT LIMITATION ANY AND ALL WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT.

IN NO EVENT WILL ANY PARTY THAT UTILIZES OR ADVOCATES THE USE OF THIS DOCUMENT BE LIABLE TO ANY OTHER PARTY FOR ANY INDIRECT, INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH THIS DOCUMENT, AND IN NO EVENT WILL THE CUMULATIVE LIABILITY OF ANY SINGLE PARTY FOR ANY AND ALL CLAIMS IN CONNECTION WITH THIS DOCUMENT EXCEED, IN THE AGGREGATE, ONE HUNDRED DOLLARS ($100.00). YOU AGREE THAT THIS PARAGRAPH IS AN ESSENTIAL BASIS OF THE USE OF THIS DOCUMENT.

Contents

Revision History

Revision 1.1 **10 December 2008**
• Added legal disclaimer page
Revision 1.0 **16 May 2008**
• Initial release.

1 Introduction

To facilitate interoperability between a datapath device and a look-aside co-processor, the Interlaken Alliance has defined Interlaken Look-Aside, a protocol suitable for short, transaction-related transfers. Although based on the Interlaken protocol, Interlaken Look-Aside is not directly compatible with Interlaken and can be considered a different operational mode.

A look-aside device does not reside in-line on the main datapath of a switch, router, or other networking device. Instead, it is connected “to the side” of the datapath. Such a device generally examines only a very small portion of a packet, or performs a comparatively limited operation under control of a datapath device. Since messages to or from a look-aside device are generally short and have different requirements than the datapath, it is desirable to optimize a protocol for these characteristics.

Some examples of look-aside devices are:

- Search engines, which receive small portions of a packet header
- Policing engines, which receive small portions of a packet header, or a simple command set
- Value-add memories, which may perform mathematical operations or linked-list traversals in addition to reads and writes
- Queuing and scheduling engines which dictate the packet transmission order to a packet buffer device

This is not intended to be an exhaustive list.

2 Look-Aside Operation

Interlaken Look-Aside is based in Interlaken, but it is not directly compatible with Interlaken. The Interlaken Protocol Specification [1], and Interlaken Interoperability Recommendations [2] must be consulted to garner a complete picture of the Interlaken Look-Aside protocol. The Interlaken specifications define Interlaken Look-Aside, except where superseded by this document.

The Interlaken Look-Aside protocol modifications comprise five main areas:

- Modifications to the control word, including fewer channels and including a protocol type field (section 2.1)
- Mandating the use of packet mode rather than segment mode (section 2.2)
- Mandating the use of single-burst packets for short packets (section 2.2)
- Reducing BurstShort from 32 bytes to as low as 8 bytes (section 2.3); and
- Mandating the use of a common reference clock and only one skip word (section 2.4).

Furthermore, Interlaken Look-Aside does not mandate any particular number of lanes for an implementation since the wide variety of look-aside applications each come with their own specific bandwidth requirements.

2.1 Interlaken Look-Aside Control Word

The Interlaken Look-Aside control word is largely the same as the Interlaken control word, but it contains a larger number of application-specific bits. This allows applications to place user data in the control word, providing high throughput for the short packets common to look-aside interfaces.

The Interlaken Look-Aside control word is identical to the Interlaken control word in bits [66:57] and [23:0], which contain the framing information and CRC, respectively. The Interlaken Look-Aside control word differs from the Interlaken Control Word in the following ways:

- The most significant channel number bit now carries a protocol identifier field.
- The number of channels is reduced to 2, and the otherwise unused channel bits now carry arbitrary user data.
- The number of flow control bits is reduced to two, and the reset calendar bit is eliminated. These otherwise unused bits now carry arbitrary user data.

The Interlaken Look-Aside control word is defined in the table below. Shaded areas represent differences between Interlaken and Interlaken Look-Aside.

Field	**Bit Pos**	**Function**
Inversion	66	Used to indicate whether bits [63:0] have been inverted to limit the running disparity; 1 = inverted, 0 = not inverted.
Framing	65:64	64B/67B mechanism to distinguish control and data words; a ‘01’ indicates data, and a ‘10’ indicates control.

Field	**Bit Pos**	**Function**
Control	63	If set to '1', this is an Idle or Burst Control Word; if '0', this is a Framing Layer Control Word.
Type	62	If set to '1', the channel number and SOP fields are valid and a data burst follows this control word (a 'Burst Control Word'); if set to a '0', the channel number field and SOP fields are invalid and no data follows this control word (an 'Idle Control Word').
SOP	61	Start of Packet. If set to a '1', the data burst following this control word represents the start of a data packet; if set to a '0', a data burst that follows this control word is either the middle or end of a packet.
EOP_Format	60:57	This field refers to the data burst preceding this control word. It is encoded as follows: '1xxx' – End-of-Packet, with bits [59:57] defining the number of valid bytes in the last 8-byte word in the burst. Bits [59:57] are encoded such that '000' means 8 bytes valid, '001' means 1 byte valid, etc., with '111' meaning 7 bytes valid; the valid bytes start with bit position [63:56]. '0000' – no End-of-Packet, no ERR '0001' – Error and End-of-Packet All other combinations are left undefined.
Application Specific 0	56:42	Application specific data associated with the newly started packet. This field is valid only in Start-of-Packet control word. It is recommended that packet processors and co-processors use this field to carry data to reduce the required number of data words.
Channel 1 XON	41	Flow control indication for channel 1; 1 = XON, 0 = XOFF.
Channel 0 XON	40	Flow control indication for channel 0; 1 = XON, 0 = XOFF.
Protocol Type	39	Optionally indicates how to decode the control word. 0 = Interlaken Control Word; 1 = Interlaken Look-Aside Control Word. Since this bit is defined as the MS channel bit in Interlaken, only implementations that choose to multiplex Interlaken and Interlaken-LA traffic need to use this bit. However, *all* Interlaken-LA implementations *must* set this bit to 1.
Application Specific 1	38:33	Application specific data associated with the newly started packet. This field is valid only in Start-of-Packet control word. It is recommended that packet processors and co-processors use this field to carry data to reduce the required number of data words.
Channel Number	32	Used to indicate one of two channel numbers associated with the data burst following the control word.

Field	**Bit Pos**	**Function**
Application Specific 2	31:24	Application specific data associated with the newly started packet. This field is valid only in Start-of-Packet control word. It is recommended that packet processors and co-processors use this field to carry data to reduce the required number of data words.
CRC24	23:0	A CRC error check that covers the previous data burst, if any, and this control word.

2.1.1 Number of Channels

In most cases, a single channel is sufficient for look-aside operations. In some cases, it is useful to have a second channel (e.g., to allow management commands and data path commands to be throttled independently).

Interlaken Look-Aside supports two logical channels. The channel number is signaled in bit [32] of the control word. Because there are only two channels, a flow control calendar is not required. Instead, two explicit flow bits in the control word, [41:40], indicate the flow control state of each channel.

2.1.2 Protocol Type

Some applications may wish to carry both packet data and co-processor requests and responses on a single Interlaken interface. In order for an implementation to correctly parse the Control Word, bit [39] is used to indicate whether the Control Word is for Interlaken or Interlaken Look-Aside. Since this bit is allocated as the most-significant channel number bit in the Interlaken Control Word, some existing Interlaken implementations may not be suitable for protocol multiplexing. Only implementations aware of Interlaken Look-Aside have any reason to interpret this bit as a protocol selector, however all Interlaken Look-Aside transmitters *must* set the Protocol Type to '1'.

For an application that carries both packet data and co-processor requests and responses on a single Interlaken interface, bits [39:32] of the control word take one of the forms below:

- {0, CHAN_NUM[6:0]} – an Interlaken frame for one of 128 different channels
- {1xxxxxx0} – an Interlaken Look-Aside frame for channel 0
- {1xxxxxx1} – an Interlaken Look-Aside frame for channel 1

Should a standard Interlaken implementation require both the use of the Protocol Type field and more than 128 channels, it expands the channel number width by allocating sufficient bits from Interlaken's Multiple Use field. Specifically, Interlaken control word bit [24] is interpreted as CHAN_NUM[7]. If more than 256 channels are required, Interlaken control word bit [25] is interpreted as CHAN_NUM[8]. Etc.

2.1.3 Application Specific Data

The Interlaken Look-Aside control word logically contains one field 29-bit application specific field, physically composed of control word bits {[56:42], [38:33], [31:24]}. This allows an application to signal almost 4 bytes of data along with the Data Words that comprise the remainder of the co-processor message. An Interlaken Look-Aside implementation *must* provide the application layer with the ability to send and receive these 29 bits, and the application layer uses them for whatever it deems necessary (e.g., to carry request identifiers, command codes, etc). Combined with a reduced Burst Short parameter (see section 2.3), this greatly improves the throughput of short messages.

To simplify implementations and to ensure interoperability, the application specific data is defined only during a start-of-packet condition, and applies to the newly started packet. Any use of this field in idle control words or end-of-packet-plus-idle control words is outside the scope of this specification.

2.2 Packet Transfer Mode and Considerations

Implementations designed expressly for Interlaken Look-Aside should be of low complexity. Hence, several requirements are placed on how packets are transferred across the interface.

2.2.1 Packet Transfer Mode

Look-aside interfaces support only packet mode operation. Because transactions are typically small in size (less than 1024 bits), interleaving segments from multiple channels has little benefit but adds significant design complexity to the interface.

Multiplexing data packet traffic and co-processor traffic on a single interface can be accomplished using the Protocol Type field (section 2.1.2). This can lead to long co-processor latencies unless a transmitter and receiver allow segment mode transfers for Interlaken while respecting Interlaken Look-Aside's requirement for packet mode transfers. This is entirely optional. A simplifying factor, however, is that Interlaken Look-Aside packets will fit within a single burst for most applications.

2.2.2 BurstMax Parameter

Interlaken recommends a BurstMax of 256 bytes. Interlaken Look-Aside *requires* that transmitters are configurable for a BurstMax of 256 bytes, and are encouraged to support higher values.

There is no requirement to support smaller values of BurstMax.

2.2.3 Maximum Sized Bursts Required

To simplify receiver implementations, if a packet's size is less than or equal to BurstMax, the transmitter *must* send the packet in a single segment. This means control words (e.g., idles) are not allowed to interrupt the packet before the EOP control word terminates the packet. The only exception to this is a metaframe boundary is permitted to interrupt a packet transfer.

Applications that do not require packets larger than BurstMax can simplify their implementations by supporting only single-burst packets. Since the required minimum BurstMax is 256 bytes, applications requiring packets less than 257 bytes never need to segment or reassemble packets. The co-processor messages for which Interlaken Look-Aside was designed are historically smaller than 256 bytes, and are not expected to grow appreciably. Larger values of BurstMax can also be supported.

Interlaken Look-Aside optionally supports transmitting and receiving packets longer than BurstMax. In this case, a packet must be broken up into more than one segment. However, all non-last segments for the packet must BurstMax bytes long and the last segment must be at least BurstShort bytes long. An implication of this is BurstMin must be equal to BurstShort for Interlaken Look-Aside packets. This is not a serious drawback since the value of BurstShort for Interlaken Look-Aside significantly reduces or even eliminates the need for the Interlaken-defined scheduling enhancements that employ the BurstMin parameter.

See section 2.3 for more information on the BurstShort parameter. See the Interlaken Specification for more information on scheduling enhancements and the BurstMin parameter.

2.3 Minimum Burst Size (“BurstShort” Parameter)

For short message sizes, the transaction rate over the Look-Aside interface is improved by reducing the value of Burst Short to 16 bytes (from 32). Further improvements can be made by reducing BurstShort.

2.3.1 Interlaken Look-Aside Requirements

On very wide interfaces, supporting increasingly smaller values for BurstShort can be difficult for receivers since many more packets can complete in a single word time. For transmitters, smaller values are not as challenging. Hence, the requirements for devices compliant to Interlaken Look-Aside are:

- Interlaken Look-Aside receivers *must* be able to receive messages sent with BurstShort = 16 bytes.
- Interlaken Look-Aside receivers can be constructed to receive messages with BurstShort = 8 bytes, if required by the application.
- Interlaken Look-Aside transmitters *must* be able to transmit messages with BurstShort = 16 bytes. This *must* be the default condition.
- Interlaken Look-Aside transmitters *should* implement a user-selectable operational mode whereby they can send messages with BurstShort = 8 bytes.

This particular set of requirements ensures there is always one mode in which a receiver and a transmitter implemented by two different parties can interoperate. I.e., Interlaken Look-Aside transmitters and receivers can always interoperate with a BurstShort of 16 bytes. Better performance can be achieved for some applications if both the transmitter and receiver use BurstShort = 8 bytes. In cases where the BurstShort parameter of the receiver does not match that of the transmitter:

- Receivers designed for BurstShort = 8 are inherently capable of interpreting a data stream transmitted with BurstShort = 16 bytes. Indeed, this is no different from simply receiving a longer packet.
- Receivers designed for BurstShort = 16 are not capable of interpreting a data stream transmitted with BurstShort = 8 bytes. However, since transmitters must also support BurstShort = 16 bytes, interoperability is ensured. The application must configure the transmitter to use BurstShort = 16 bytes to communicate with a receiver designed for BurstShort = 16 bytes.

2.3.2 Dual-Mode Requirements

For implementations that support both Interlaken and Interlaken Look-Aside dynamically on a single interface by interpreting the Protocol Type field in the control word:

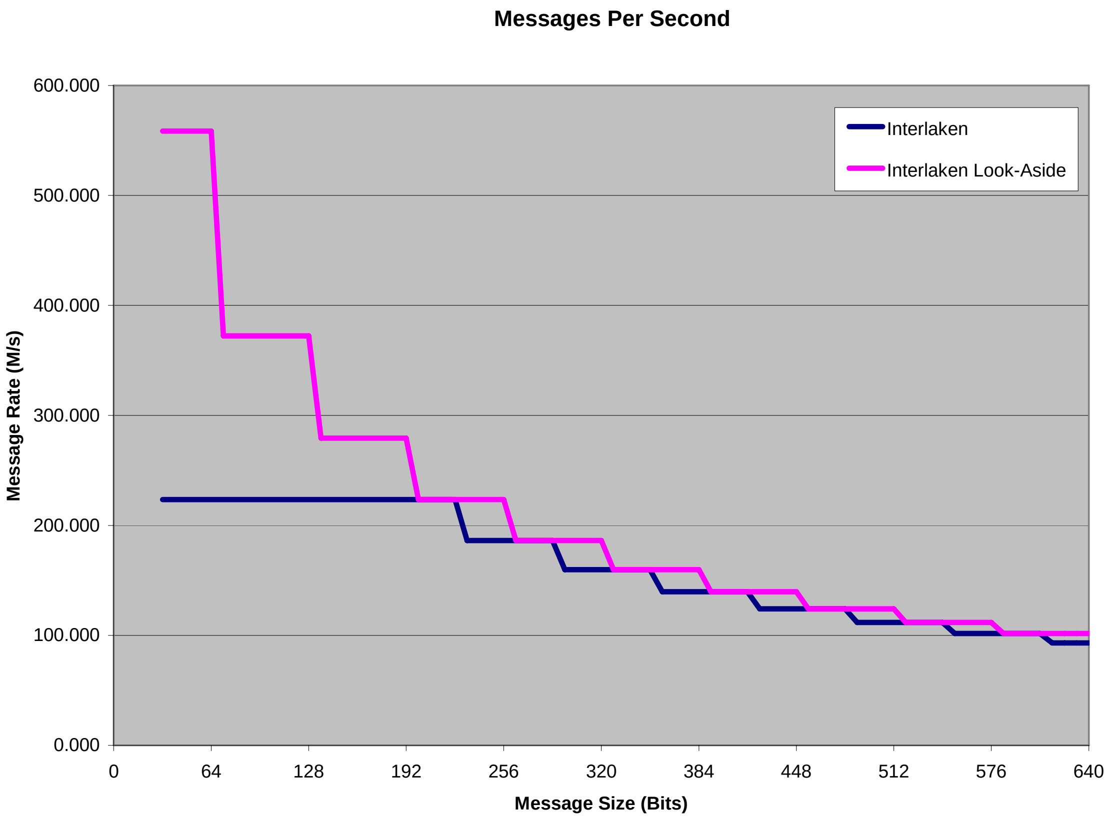
- Receivers must support the smallest expected BurstShort. Since Interlaken specifies a minimum BurstShort 32 bytes, the demands for Interlaken Look-Aside are more stringent, and the receiver must accept a BurstShort of 16 or 8 bytes, as described in section 2.3.1.

- Transmitters must use a BurstShort of 16 or 8 bytes for Interlaken Look-Aside frames, as described in section 2.3.1. Transmitters must use a BurstShort of at least 32 bytes for standard Interlaken frames. This ensures any intervening standard-Interlaken-only switches, or any final-destination standard-Interlaken-only devices are able to interpret the bursts.

2.3.3 Performance Graphs

Figure 2.3-1 and Figure 2.3-2 illustrate the impact of the Interlaken Look-Aside protocol on small packets.

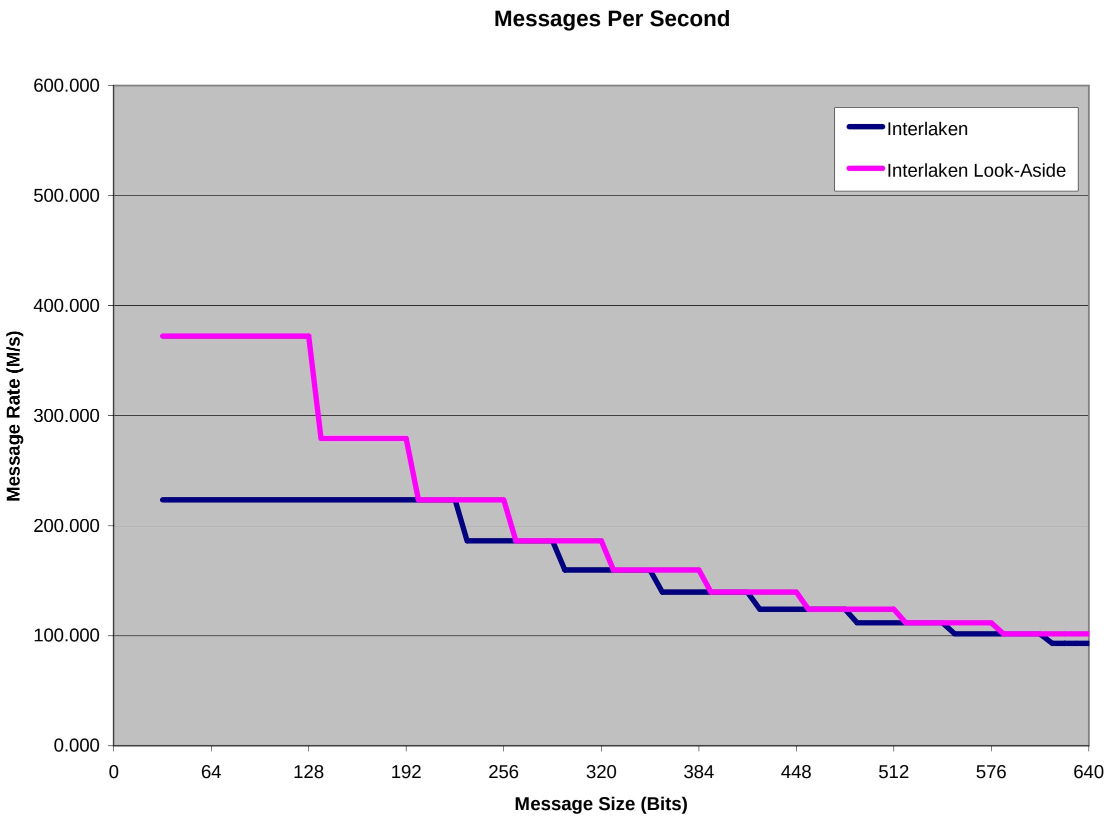
Figure 2.3-1 Transaction Rate Increase[1] with BurstShort = 8 (12 lanes of 6.25 Gbps)



The figure above shows the message rate for Interlaken and Interlaken Look-Aside when the BurstShort is configured for 8 bytes, and 12 lanes of 6.25 Gbps are aggregated to form a single interface. As a real-world comparison, an application performing multiple IP and MAC address lookups is practical at 100 Gbps line rates with Interlaken Look-Aside. Note also that most co-processor response messages are generally short, and Interlaken Look-Aside yields a high rate for short messages.

[1] The InterLaken plot includes 4 bytes of message overhead, carried in data words. The Interlaken-LA plot assumes the same ~4 bytes of overhead is carried in the “application specific” field of the InterLaken-LA control word. Interlaken uses a BurstShort of 32 bytes.

Figure 2.3-2 Transaction Rate Increase[2] with BurstShort = 16 (12 lanes of 6.25 Gbps)



The figure above shows the message rate for Interlaken and Interlaken Look-Aside when the BurstShort is configured for 16 bytes, and 12 lanes of 6.25 Gbps are aggregated to form a single interface. As a real-world comparison, an application performing IP and MAC address lookups is practical at 100 Gbps line rates with Interlaken Look-Aside.

Note also that most co-processor response messages are generally short, and Interlaken Look-Aside yields a high rate for short messages.

2.4 Common Reference Clock

If both the data path device and the look-aside co-processor use the identical reference clock, then there is never a need to send additional skip characters. This significantly simplifies the receiver logic as there is never a need to shift payload data to maintain the constant length metaframe. A modest amount of bandwidth increase is realized by reducing the overhead of synchronization characters.

[2] The InterLaken plot includes 4 bytes of message overhead, carried in data words. The Interlaken-LA plot assumes the same ~4 bytes of overhead is carried in the “application specific” field of the InterLaken-LA control word. Interlaken uses a BurstShort of 32 bytes.

An Interlaken Look-Aside transmitter and receiver *must* use a common reference clock, ensuring there is absolutely no rate difference between the transmit and receive clocks.

An Interlaken Look-Aside metaframe consists of only one skip character; the skip character immediately following the scrambler state word is mandated by Interlaken, and Interlaken Look-Aside does not change this. However, an Interlaken Look-Aside metaframe *must not* contain any additional skip characters.

The exception to this section is for dual-mode Interlaken transmitters connected to dual-mode Interlaken receivers. These are applications that support both Interlaken and Interlaken Look-Aside packets dynamically, using the Protocol Type field to determine how to handle incoming packets. Such applications are permitted to use whatever clocking scheme is appropriate, and are therefore permitted to follow the Interlaken specification[3].

[3] However, to be clear, a protocol stack capable of dual mode, but instantiated in an Interlaken-Look-Aside-only device must still follow the Interlaken Look-Aside rules as it is an Interlaken Look-Aside application.

3 References

[1] Cortina Systems and Cisco Systems: “Interlaken Protocol Specification, revision 1.1”, July 25, 2006.

[2] Interlaken Alliance: “Interlaken Interoperability Recommendations, revision 1.0”, October 31, 2007.