Accredited Standards Committee
X3, Information Processing Systems

Doc: X3T10.1/96a143R0
Date: May 16, 1996
Project: 1147D
Ref Doc.: X3T10.1/1147D rev 0
Reply to: Mark DeWilde

To: X3T10.1 Membership
From: Mark DeWilde

Subject: Resurrection of Longer full speed Link proposal

BACKGROUND

In October of 1994, Ed Gardner proposed a method of extending SSA full speed link length beyond the current 680 meters. That proposal (X3T10.1/94-064r0) was not adopted for inclusion into TL1. For TL2 I believe that it is essential to include a mechanism to extend the full-speed maximum distance, since every doubling of bit rate effectively halves the full-speed maximum length of a link. Note that the maximum length of a link is determined by the ACK timeout, and that length is not changed by this proposal. In order to avoid the need to look Ed's document, the following sections are excerpted from his proposal:

"Longer link lengths and higher data rates are obviously desirable in any interconnect. Recent statements by proponents of other interfaces have made it apparent that we need to address these in SSA soon, for specsmanship if for no other purpose. The following briefly describes one way of accomplishing this superior to other concepts that have been mentioned. I will expand this into a detailed proposal if the committee decides it is worth pursuing.

The key parameter in link performance is either the amount of buffering in a receiving port or the amount of data a transmitting port may have outstanding without acknowledgments. In an efficient link design such as SSA these two parameters are equal, as the minimum of the two dictates the performance limitation. In order to sustain continuous transmission, these parameters must be large enough to accommodate the round-trip delay time of the link. In SSA the following relation must hold to achieve continuous transmission for optimum performance:

a) $$B \geq \left\lceil \frac{T_{rt} * R_{data}}{S_{frame}} \right\rceil + 1$$

where B is the number of buffers required, T_{rt} is the round trip delay time, R_{data} is the data transmission rate, and S_{frame} is the frame size. Equivalently:

a) $$T_{rt} \leq \frac{(B-1) * S_{frame}}{R_{data}}$$

The parameter $(B-1)$ is referred to below as the "window size", a common term for this in communication protocols.

From this it is straightforward to recalculate the maximum distance at full speed numbers given in SSA-PH section 7.5:

$$\text{maximum distance} \le \text{propagation velocity} * \frac{T_{rt}}{2}$$

a)
$$\le \frac{200\,\text{M/us}}{2} * \frac{(2-1)\,\text{frames} * 136\,\text{bytes/frame}}{20\,\text{bytes/us}}$$
$$\le 680\,\text{M}$$

Now, the point of this exercise is to persuade that, if we increase the data transmission rate R_{data}, there are three and only three possibilities:

a) 1. We can reduce the round trip delay T_{rt} and thus the maximum distance proportional to the increase in data transmission rate. However, this is undesirable because it leads to unfortunate specsmanship comparisons with other interfaces.

b) 2. We can increase the frame size S_{frame} proportional to the increase in data transmission rate. However, this is undesirable because it leads to serious compatibility problems in mixed speed environments, which only get worse as additional transmission rates are defined. The notion of defining multiple transmission rates, each with its own frame size, is absurd.

c) 3. We can increase the number of buffers. Maintaining our current maximum distance at double the data transmission rate requires three buffers instead of two. Allowing an option to implement more than three buffers lets us increase the maximum distance in the spec without impacting the majority of devices that have no practical use for such distance.

The remainder of this proposal summarizes how to allow implement multiple buffers to show that it is quite straightforward. This is a routine technique that has been commonly used since nearly the first communication protocols were invented.

First, the ports on either side of a link agree on the number of buffers they will use for communication, presumably equal to the minimum number that either implements. This should use whatever mechanism is used to negotiate a higher data transfer rate. If the ports agree to use two buffers, then operation is absolutely identical to what the spec describes today. (Agreeing to use one buffer also results in operation identical to today's, but I think that using only one buffer ought to be prohibited for other reasons). I will use the phrase "window size" to mean a value one less than the agreed upon number of buffers (e.g., window size is 1 for current using 2 buffers, or is 2 for double data rate operation using 3 buffers). This is a standard term in communication protocols for this parameter.

Second, the Waiting_for_ACK, Waiting_for_RR, and RR_pending flags currently described in the spec become counters. The values that need to be stored are zero through the window size inclusive. Two bit counters are sufficient for the three buffers necessary to maintain current length limits at double speed. Note that ACK_pending remains a flag.

When a port enters the Ready state, it sets its Waiting_for_ACK counter to zero and sets it's Waiting_for_RR and RR_pending counters to the window size (the maximum). This replaces today's clearing of Waiting_for_ACK and setting of Waiting_for_RR and RR_pending. Elsewhere, where the spec today says a flag is cleared, the corresponding counter will be decremented. Where the spec today says a flag is set, the corresponding counter will be incremented.

Transmitting ports reset their ACK timeout timer whenever they receive an ACK. Thus an ACK timeout occurs when the Waiting_for_ACK counter is non-zero and more than 25us has elapsed since either sending a frame or receiving an ACK.

The current rule that a port may not transmit the trailing FLAG when a port is Waiting_for_ACK is modified to say a port may not transmit the trailing FLAG when the Waiting_for_ACK counter is equal to the window size.

The current rule that a port may start to send a frame whenever it is not Waiting_for_RR is modified to say a port may start to send a frame whenever its Waiting_for_RR counter is less than the window size.

The current rule that a port transmits an RR whenever RR_pending is set is modified to say a port transmits an RR whenever RR_pending is non-zero.

Finally, the size of the Frame Sequence Number implemented internally within ports should be increased to be at least as large as the maximum number of buffers we ever anticipate using. While the larger FSN will be maintained internally within ports, only the two low order bits are sent in frame control fields; the control field format does not change. However, the data field in Link Reset frames will be extended to allow the complete FSN to be sent during the Link ERP. Note that I say the size of the FSN "should" be increased, because I believe it leads to a simple and more reliable system. An argument can be made that the larger FSN is unnecessary, however I feel it is safer."

This proposal incorporates much of Ed's original proposal, and adds a means to implement the additional feature in a backwards compatible manner so that nodes incorporating this feature will interoperate with today's nodes.

PROPOSAL

All frames are in the 7 to 140 byte range in length. Using Ed's formula, 7143 buffers are required for full speed operation at 200 Mbits/sec and 7 byte frames. At 400 Mbits/sec, 14286 buffers are required. This number yields the maximum size of the counters required for the Waiting for RR, RR Pending and Waiting for ACK functions. If average frame lengths are 136 for data and 30 for SMSs, and the link carries a 80/20 data/instruction mix, then the average frame is 115 bytes. At 1.8 Gb/sec, full speed is achieved with 39132 buffers. I propose that 16 bit counters be the maximum permitted. The maximum data rate for full speed transmission with 16-bit counters and 115 byte average frames is about 3 Gb/sec. The maximum number of frames that can be outstanding is determined by the port with the lesser number of frame buffers, since the number of send buffers in one port must be in one-to-one correspondence with the number-1 of receive buffers in the other port. This is necessary to ensure that retransmission is possible should there be a link error. If there are insufficient receive buffers, then not all of the transmit buffers in the sending port may be used. This "window size" must be negotiated between the nodes prior to using the extended transfer capability. The establishment of the window size is performed by the Master, using the scheme outlined in the next paragraph.

All ports default to the current mode of operation in respect to RR's and flags when they begin communications. During configuration the presence of the extended buffering capability on a node is determined by the (modified)Query Node Reply SMS. If two nodes capable of this feature are found to be connected, then the Master issues Query Port SMSs to the connected ports to determine if both ports have this feature, and the number of buffers implemented (reported in the QUERY PORT REPLY SMS). If both ports of a link are found to be capable of extended distance operation, then they are enabled to do so by being provided with a window size greater than 2 and equal to the smaller of the window sizes when the master issues the (modified)Configure Port SMSs. Thus, If two ports capable of this feature are interconnected, the feature will be enabled. If one of the interconnected ports cannot support this feature, then it is not enabled. If it is enabled, then the window size will be the smaller of the two port's number of buffers.

When the Master sends the (modified)Configure Port SMS to the port to enable the extended buffer count feature, the port sets it's Waiting_for_RR and RR_pending counters to the window size. The Waiting_for_ACK counter was set to zero when the port entered the ready state. The following sections from Ed's proposal are brought forward:

"Elsewhere, where the spec today says a flag is cleared, the corresponding counter will be decremented. Where the spec today says a flag is set, the corresponding counter will be incremented.

Transmitting ports reset their ACK timeout timer whenever they receive an ACK. Thus an ACK timeout occurs when the Waiting_for_ACK counter is non-zero and more than 25us has elapsed since either sending a frame or receiving an ACK.

The current rule that a port may not transmit the trailing FLAG when a port is Waiting_for_ACK is modified to say a port may not transmit the trailing FLAG when the Waiting_for_ACK counter is equal to the window size.

The current rule that a port may start to send a frame whenever it is not Waiting_for_RR is modified to say a port may start to send a frame whenever its Waiting_for_RR counter is less than the window size.

The current rule that a port transmits an RR whenever RR_pending is set is modified to say a port transmits an RR whenever RR_pending is non-zero.

Finally, the size of the Frame Sequence Number implemented internally within ports should be increased to be at least as large as the maximum number of buffers we ever anticipate using. While the larger FSN will be maintained internally within ports, only the two low order bits are sent in frame control fields; the control field format does not change. However, the data field in Link Reset frames will be extended to allow the complete FSN to be sent during the Link ERP. Note that I say the size of the FSN "should" be increased, because I believe it leads to a simple and more reliable system. An argument can be made that the larger FSN is unnecessary, however I feel it is safer."

The recommendation Ed made on the Frame Sequence Number bit field width is well taken. Since both frames and acknowledges can be lost in a link fault situation, both ports need to synchronize on the frame number to be resent and acknowledged. Since there can be many frames in the pipeline, the FSN/RSN counters may roll over multiple times. This makes for a difficult recovery at best. For these counters, I also recommend 16 bits of width. For ports that have this capability, the link reset frame is extended by two additional bytes, which carry the full 16 bit counter value. The format of the first link status byte remains unchanged.

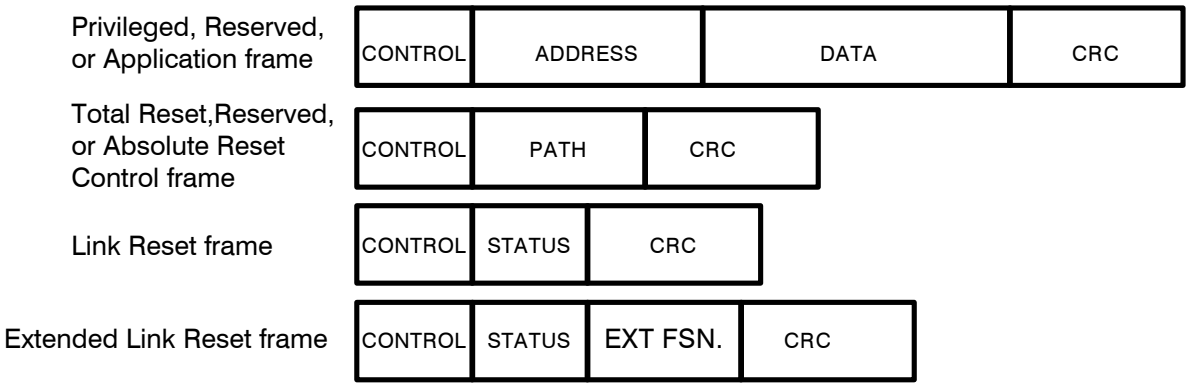

Figure 8 - Specific frame format

The query node reply SMS must be modified to indicate whether extended distance is supported on the node:

Table 33 - QUERY NODE REPLY SMS

Byte	Bit 7	6	5	4	2	1	Bit 0
0	SMS CODE (01h)						
1	PORT						
2	TAG						
3	TAG						
4	UPPER LEVEL PROTOCOL						
5	ITF	MASTER PRIORITY			reserved		
6	TOTAL OTHER PORTS						
7	SSA-TL VERSION (10h)						
8	UNIQUE ID						
9	UNIQUE ID						
10	UNIQUE ID						
11	UNIQUE ID						
12	UNIQUE ID						
13	UNIQUE ID						
14	UNIQUE ID						
15	UNIQUE ID						
16	RETURN PATH ID						
17	RETURN PATH ID						
18	RETURN PATH ID						
19	RETURN PATH ID						
20	P1O	P2O	long	reserved			

The LONG field is set to “1” if any port on the node is equipped with the extended option. If this bit is set, then a QUERY PORT SMS may be sent to inquire for the port numbers and window size.

The query port reply must be modified to permit the reporting of extended distance ports and window size:

Table 1 - QUERY PORT REPLY SMS

Byte	Bit 7	6	5	4	3	2	1	Bit 0
0	SMS CODE (0Bh)							
1	reserved							
2	TAG							
3	TAG							
4	LINK ERP ERROR COUNT							
5	LINK ERP ERROR COUNT							
6	A QUOTA							
7	B QUOTA							
8	EUDC	REFLECT	MODE		reserved			
9	WINDOW SIZE							
10	ALARM THRESHOLD							
11	ALARM THRESHOLD							
12	SUPPORTED SPEED							
13	SUPPORTED SPEED							
14	CURRENT SPEED							
15	CURRENT SPEED							

The WINDOW SIZE field indicates the number of buffers implemented in the port for extended distance operation. A WINDOW SIZE of 1 or 2 indicates a port without the extended distance option. Larger values indicate the port supports the extended distance option.

The CONFIGURE PORT SMS must be modified to set the window size of the port:

Table 2 - CONFIGURE PORT SMS

Byte	Bit 7	6	5	4	3	2	1	Bit 0
0	SMS CODE (02h)							
1	PORT							
2	TAG							
3	TAG							
4	RETURN PATH							
5	RETURN PATH							
6	RETURN PATH							
7	RETURN PATH							
8	reserved							
9	A QUOTA							
10	B QUOTA							
11	EUDC	REFLECT	MODE		reserved			
12	reserved							
13	ALARM THRESHOLD							
14	ALARM THRESHOLD							
15	NEGOTIATE SPEED							
16	NEGOTIATE SPEED							
17	WINDOW SIZE							
18	WINDOW SIZE							

The WINDOW SIZE field is the number of buffers that will be used by the link for communications. If this number is 1 or 2, then link behavior is the default SSA type. If this field is greater than 2, then extended distance link behavior is used. This size is set to the smaller of the window sizes returned by the QUERY PORT REPLY SMSs from the connected ports.

Sincerely,

Mark A. DeWilde
Principal System Architect
Pathlight Technologies

Voice: (607)266-4000 X-403
FAX: (607)266-0352
Email: mark@pathlight.com