Register Insertion Protocol for Fibre Channel Arbitrated Loop

1. Revision History

1.1 Rev 0.9 6/1/96

Initial version

1.2 Rev 1.0 12/15/96

Changed flow control mechanism from end-to-end (a.k.a. multipoint) to nearest neighbor.

Changed fairness mechanism from arbitration on starvation to the SAT algorithm.

Open issues:

⇒ Holding the SAT token to transmit while the Bypass FIFO is not empty requires depleting the upstream neighbor of available credit. This may result in a delay in replenishing credit after the L_Port is satisfied and the SAT token is released.
Solution: employ the Aaron method of receive buffering (common cut through and receive buffers)

⇒ Since RIP packets containing R_RDYs are not flow controlled, they may overflow the Bypass FIFO of a downstream L_Port when it is holding the SAT token.
Solution: Employ the Aaron method of embedding Primitive Signals within frames (RIP packets in the case of RIP).

2. Overview

This proposal documents a new operating mode for FC-AL, which is called the Register Insertion Protocol or RIP. As the name suggests, it is based on a modification to FC-AL which adds a new operating mode based on a register insertion mechanism. The primary motivation is to reduce the latency of acquiring access to and closing the Loop. RIP completely eliminates arbitration for Loop access.

When the Loop is idle, an L_Port may transmit one frame or one or more R_RDYs at any time. RIP encapsulates the frame or R_RDY(s) with a pair of Primitive Signals called Receive or RCV. The resulting string of Transmission Words is called a RIP packet. If another RIP packet is received and must be retransmitted (i.e., it is bound for another L_Port) while the L_Port is transmitting, the packet is temporarily buffered in a Bypass FIFO, which can accommodate a maximum size RIP packet. The buffered packet is retransmitted as soon as the L_Port has completed transmitting a single RIP packet.

The Bypass FIFO must be empty before an L_Port is allowed to transmit, since it may have to temporarily buffer part of all of a maximum length RIP packet after it begins transmitting.

A frame or a set of R_RDYs are routed on the Loop by prepending each with a new Primitive Signal, RCVyx, which contains the source and destination AL_PAs. The frame or set of R_RDYs has a RCVff Primitive Signal that marks the end of the transmission. The RCVyx header, encapsulated Transmission Words, and the RCVff trailer delineate a RIP packet. This packaging and routing scheme preserves much of the existing structure of FC-AL. It allows routing based solely on Primitive Signals and avoids the need to crack the frame header and examine the D_ID.

The RIP mode of operation, based on the register insertion protocol, results in the lowest possible latency, since frame transmission may usually begin at any time. Arbitrated Loop as defined in FC-AL X3, 272-1996, (hereafter referred to as FC-AL1), typically requires one half of a Loop round trip on average for arbitration before an L_Port may begin frame transmission. FC-AL1 also permits only a single point-to-point full duplex circuit per connection. It also requires one round trip to close the Loop. Other L_Ports which have frames to transmit must wait until the current circuit is terminated when the L_Ports relinquish control by closing. Thus, short command packets may be blocked behind large data transfers. In contrast, RIP mode allows multiple L_Ports to interleave frame transmission on the Loop. RIP allows short command packets to be interleaved with large data transfers.

RIP may allow improved class 2 performance, since ACK frames may be transmitted without arbitrating for the Loop. With FC-AL1, if the destination L_Port is closed immediately after class 2 frame reception, it may have to arbitrate and acquire Loop access just to return the ACK frame. Because of the nature of RIP, class 1 is not supported.

The goal of RIP is to minimize latency, which is often at odds with providing bandwidth and latency guarantees. In other words, the average latency of RIP is much lower than FC-AL1. RIP supports a variable packet or frame size. With a variable frame size and on-demand use of available bandwidth, RIP will efficiently utilize Loop resources. As a result, RIP will exhibit higher Loop utilization than FC-AL1.

RIP is designed to support asynchronous or dynamic workloads that typically exhibit a wide range of transfer length and bandwidth demand. It is also designed to support applications that frequently need to synchronize and communicate in a cluster or multi-computer environment. Such applications include Multi-Initiator Disk Arrays, On Line Transaction Processing (OLTP), Data Warehousing, Data Mining, large scale Network File Servers, scaleable Web servers, and parallel processing.

3. Register Insertion Protocol Definition

This proposal defines a new operational mode for Fiber Channel Arbitrated Loop. It is based on a register insertion mechanism, which is an extension for FC-AL2. The new operational mode is called the Register Insertion Protocol or RIP.

RIP mode is entered at the end of loop initialization when the temporary Loop Master discovers whether all L_Ports are capable and enabled for RIP mode. If true, all L_Ports are instructed to enter RIP mode. Once in RIP mode, all L_Ports remain in a Monitoring like state. Thus, L_Ports never enter the OPEN or OPENED states.

3.1 Basic Operation

Since the Loop is a closed, continuous circuit through all L_Ports, a routing mechanism must be provided for each frame or R_RDY Primitive Signal. For unicast operation, this is accomplished by immediately prepending each frame or one or more R_RDYs with a Receive (RCVyx) Primitive Signal. The frame or

R_RDYs have two Fill Words, a RCVff, and two additional Fill Words appended to it. This string of transmission words is called a RIP packet.

The destination L_Port whose AL_PA matches the “y” in the RCVyx Primitive Signal removes from the Loop the RCVyx, the frame or R_RDY(s) and all other Transmission Words between the RCVyx and RCVff inclusive. In addition, a RCVfx (f = hex ‘FF’) Primitive Signal provides support for broadcast, while a RCVef/RCVye pair (e = hex ‘FE’) supports multicast or selective replicate.

An L_Port may transmit one RIP packet whenever it has received at least six consecutive Idles and its Bypass FIFO is empty. During transmission of a RIP packet, if any frame or R_RDYs which are addressed to another L_Port or a MRKtx are received , they are stored in a special purpose receive buffer, called the Bypass FIFO. When the sender completes transmission of a single frame or one or more R_RDYs, it must immediately transmit any Transmission Words in the Bypass FIFO. The Bypass FIFO must be completely drained before any additional frames or R_RDYs may be transmitted. All transmitted frames and R_RDYs are packaged into RIP packets.

3.2 RIP L_Port Structure

The logical structure of a RIP capable L_Port is shown in Figure 1. A 3-to-1 mux connects three different sources to the transmitter output:

1) receiver input (“Idle” state)
2) Bypass FIFO output (“Drain” state)
3) Transmit (TX) FIFO output (“Inject” state)

The Receive (RCV) FIFO is shown for descriptive purposes only and is not required for RIP mode. It is sufficient to deliver the contents of a RIP packet directly to FC-2. The Transmit (TX) FIFO shall be at least as large as the largest RIP packet that the L_Port constructs.

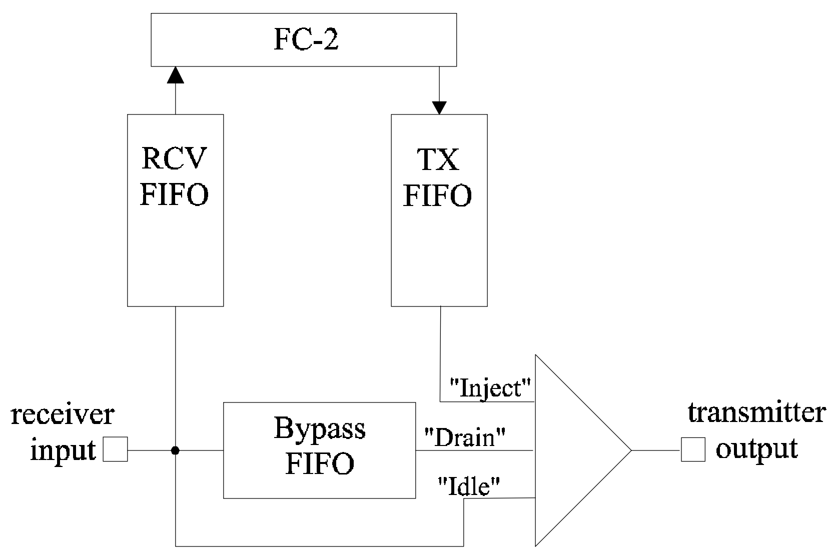

Figure 1: RIP L_Port Structure

When the L_Port is in the Idle state, it is repeating the receiver input bound for other L_Ports to the transmitter output, including any elasticity management. When the L_Port is transmitting a RIP packet the L_Port is in the Inject state and the TX FIFO output is connected to the transmitter output. If one or more RIP packets or MRKtx Primitive Signals for other L_Ports are received while the L_Port is transmitting a RIP packet, then they are buffered in the Bypass FIFO. When the L_Port has emptied the TX FIFO, which shall only contain a single RIP packet, the L_Port enters the Drain state where the transmitter output is connected to the Bypass FIFO output if it is not empty. Otherwise, the receiver input is connected to the transmitter output and the L_Port enters the Idle state.

The Bypass FIFO output is connected to the transmitter output as long as it is not empty. If any subsequent RIP packets destined for other ports are received while the Bypass FIFO is connected to the transmitter output, the received Transmission Words are appended to the Bypass FIFO. When the Bypass FIFO is empty, the receiver input is connected to the transmitter output. Any frames or R_RDYs contained in RIP packets addressed to the L_Port are placed in the RCV FIFO and shall not be buffered in the Bypass FIFO.

3.3 RIP Primitive Signals

3.3.1 New Primitive Signals

RIP defines two new Primitive Signals. One is named Receive or RCV, while the second is named SATisfied or SAT. RCV is defined by the Ordered Set K28.5, Dxx.y, nd1, nd2, where each nd is a neutral disparity transmission character. SAT is defined by the Ordered Set K28.5, Dxx.y, Dxx.y, Dxx.y.

The RCV and SAT Primitive Signals shall follow the FC-PH rule for transmitting R_RDYs (i.e., two Fill Words shall precede and follow each RCV or SAT with at least six Primitive Signals between frames).

RCV provides seven different functions, depending on the values for nd[1, 2] in the third and fourth transmission characters of the RCV Ordered Set. See Table 1. A variation of SAT, SAT’, is a special form of SAT originated by single port Nodes and Fabric ports in certain topologies.

Table 1 - RIP Primitive Signals

RIP Primitive Signal	nd1 (char. 3)	nd2 (char 4)	Function
RCVyx	<= EF	<= EF	Unicast Packet
RCVfx	FF	<= EF	Broadcast Replicate Packet
RCVex	FE	<= EF	Selective Replicate Packet
RCVye	<= EF	FE	Selective Replicate Receive
RCVff	FF	FF	RIP packet EOT
RCVbb	FB	FB	Negotiate RIP mode
RCVdd	FD	FD	Enter RIP mode
SAT	Dxx.y	Dxx.y	Satisfied
SAT’	Dxx.y	Dxx.y	Satisfied’

3.3.1.1 RCVyx (Unicast Packet)

RCVyx, where x <= hex ‘FE’ and y <= hex ‘FE’, is transmitted on a Loop by a participating L_Port to transmit RIP packet to a destination L_Port. The RCVyx shall contain the AL_PD (destination - y value) of the L_Port to receive the frame or R_RDY and the AL_PS (source - x value) of the L_Port transmitting the RCVyx.

An L_Port that receives the RCVyx, where y = AL_PA of the L_Port, shall remove all Transmission Words starting with the RCVyx and the next RCVff inclusive. The L_Port shall also store in the Receive FIFO the received frame or R_RDY(s) contained within the RIP packet.

An L_Port that receives the RCVyx, where y <> AL_PA of the L_Port and x = AL_PA of the L_Port, shall discard all Transmission Words between the RCVyx and the next RCVff inclusive.

3.3.1.2 RCVfx (Broadcast Replicate Packet)

RCVfx, where f = hex ‘FF’ and x <= hex ‘FE’, is transmitted on a Loop by a participating L_Port to transmit a single frame to all other L_Ports on the Loop. The RCVfx shall contain the AL_PS (source - x value) of the L_Port transmitting the RCVfx.

An L_Port that receives the RCVfx, where x <> AL_PA of the L_Port, shall retransmit all Transmission Words between the RCVfx and RCVff inclusive. The L_Port shall also store in the Receive FIFO the received frame contained within the RIP packet.

An L_Port that receives the RCVfx, where x = AL_PA of the L_Port, shall discard all Transmission Words between the RCVfx and the next RCVff inclusive.

3.3.1.3 RCVex/RCVye (Selective Replicate Packet)

RCVex, where e = hex ‘FE’ and x <= hex ‘FE’, is transmitted on a Loop by a participating L_Port to transmit a single frame to a subset or group of L_Ports on the Loop. The RCVex shall contain the AL_PS (source - x value) of the L_Port transmitting the RCVex.

In addition, the L_Port transmitting the RCVex shall transmit immediately following the RCVex one RCVye with appropriate Fill Words for each L_Port in the group. The RCVye, where e = hex ‘FE’, shall contain the AL_PD (destination - y value) of the L_Port to copy the frame.

An L_Port that receives the RCVex, where x <> AL_PA of the L_Port, shall retransmit all Transmission Words between the RCVex and the next RCVff inclusive, with the following exception. If the L_Port receives a RCVye, where y = AL_PA of the L_Port, it shall substitute the CFW; the L_Port shall also store in the Receive FIFO the received frame contained within the RIP packet.

An L_Port that receives the RCVex, where x = AL_PA of the L_Port, shall discard all Transmission Words between the RCVex and the next RCVff inclusive.

3.3.1.4 RCVff (RIP packet EOT)

RCVff, where f = hex ‘FF’, is transmitted on the Loop by a participating L_Port as the last Transmission Word of a RIP packet.

3.3.1.5 RCVbb (negotiate RIP mode)

RCVbb, where b = hex ‘FB’, is transmitted on the Loop in the Open-Init state by the temporary Loop Master if it is capable and enabled for RIP mode. If the temporary Loop master subsequently receives a RCVbb, it shall discard the RCVbb and then transmit a RCVdd. If the E_D_TOV timer expires before receiving a RCVbb, the temporary Loop Master shall transmit a CLS and make the transition to the Monitoring state.

If an L_Port in the OPEN-INIT state that is not the temporary Loop Master receives a RCVbb and it is capable and enabled for RIP mode, it shall retransmit the RCVbb. Otherwise, it shall be discarded.

3.3.1.6 RCVdd (enter RIP mode)

RCVdd, where d = hex ‘FD’, is transmitted on the Loop in the Open-Init state by the temporary Loop Master when it has sent and then receives a RCVbb Primitive Signal. If the temporary Loop master subsequently receives a RCVdd, it shall make the transition to the Idle state. If the E_D_TOV timer expires before receiving a RCVdd, the temporary Loop Master shall transmit a CLS and make the transition to the Initializing state.

If an L_Port in the OPEN-INIT state receives a RCVdd, it shall make the transition to the Idle state.

3.3.1.7 SAT (SATisfied)

SAT is retransmitted on a Loop by an L_Port if it has no RIP packets to originate, or if it has RIP packets to originate and it has reached it’s HOLD quota since the last SAT visit. SAT is unconditionally retransmitted if the L_Port has originated it’s IDLE quota for RIP packets since the last SAT visit. A SAT visit is defined as the next receipt of SAT after an L_Port has transmitted a SAT. In a cyclic path such as a loop, the SAT token will visit each node exactly once during a complete circulation around the loop.

3.3.1.8 SAT’ (SATisfied’)

For dual, counter rotating Loop topologies with single port Nodes and/or Fabric Nodes at either end of the loop, SAT is retransmitted on the Loop by single port or Fabric Node as SAT’. SAT’ is re-transmitted on the Loop by a single port or Fabric Node as SAT. All dual port Nodes immediately retransmit SAT’ without affecting any transmission quotas (Hold, Idle, Frame Counter).

3.3.2 Normal FC-AL Primitive Signals

The MRK tx Primitive Signal functions as defined in FC-AL, Rev. X3,272-1996.

3.3.3 Unused FC-AL Primitive Signals

The unused FC-AL Primitive Signals are ARB, OPN, CLS, and DHD. If received by an L_Port in RIP mode, the CFW shall be substituted.

3.4 RIP Characteristics

3.4.1 RIP Packet Structure

Figure 2 illustrates the 4 different types of RIP packets. The symbol {X} represents one or more occurrences of X (i.e., X, X, X ...). VDW is any Valid Data Word.

Unicast Frame	RCVyx	FW	FW	SOF	{VDW}	EOF	FW	FW	RCVff			
Broadcast Frame	RCVfx	FW	FW	SOF	{VDW}	EOF	FW	FW	RCVff			
Multicast Frame	RCVex	FW	FW	{RCVye	FW	FW}	SOF	{VDW}	EOF	FW	FW	RCVff
R_RDY	RCVyx	FW	FW	{R_RDY	FW	FW}	RCVff					

Figure 2: RIP Packet Structure

3.4.2 Bypass FIFO Size

The Bypass FIFO shall be large enough to hold one maximum length frame plus any required leading and trailing Fill Words and Primitive Signals. The maximum length of a transmitted RIP packet occurs when a full sized frame is selectively replicated to all 126 other L_Ports on the Loop (this is an unlikely occurrence, since a broadcast may be used instead). This is equal to 3688 bytes, as indicated in Figure 3.

Transmission Words	Byte Count
RCVex	4
2 Fill Words	8
126 * (RCVye, 2 Fill Words)	1492
SOF	4
Header	24
Data Field	2112
CRC	4
EOF	4
2 Fill Words	8
RCVff	4
6 Fill Words	24
Total:	3688

Figure 3: Bypass FIFO Size

3.4.3 Drain Behavior

The Bypass FIFO will empty at the rate at which Fill Words are received that are not between a RCVyx, RCVfx or RCVex and the next RCVff. For example, assume the Bypass FIFO has N Transmission Words buffered. If only Fill Words are received while in the Drain state then the Bypass FIFO will be emptied in N Transmission Word times. If however, a sequence of consecutive RIP packets are received while in the Drain state, then the Bypass FIFO cannot drain. It is loaded and unloaded on every Transmission Word. Thus the Bypass FIFO count will remain at N. If a mixture of RIP packets and Fill Words are received, the Bypass FIFO count is decremented for every Fill Word received that is not encapsulated by the RCVyx | RCVfx | RCVex and RCVff pair.

3.4.4 Port Delay

While an L_Port is in Drain mode, the length of the Loop and the delay through the L_Port is increased by the number of Transmission Words buffered in its Bypass FIFO. If N Transmission Words are buffered while in the Inject state, then the port delay will be a maximum of N+6 Transmission Words upon entering the Drain state. As the Bypass FIFO empties, the port delay decreases. Eventually the Bypass FIFO will drain and the delay through the port returns to a maximum of six Transmission Words per L_Port as specified by FC-AL1.

3.4.5 Loop Delay

The length of a Loop in Transmission Words temporarily expands by the combined contents of all Bypass FIFOs. In the worst case, the maximum length occurs when all L_Ports simultaneously transmit a full selective replicate RIP packet. For M L-Ports in a Loop, the transmission length of the Loop temporarily expands by M * 3688/4 or M * 922 Transmission Words.

3.5 RIP Initialization

Before the temporary Loop Master transmits a CLS to exit initialization, it shall determine if RIP mode is possible. If it is capable and enabled for RIP mode, the Loop Master shall transmit a RCVbb Primitive Signal, where b = hex ‘FB’. If an L_Port that is not the temporary Loop Master receives RCVbb and it is capable and enabled for RIP mode, it shall retransmit the RCVbb. If that L_Port is neither capable nor enabled for RIP mode, it shall discard the received RCVbb Primitive Signal. If all L_Ports on the Loop are capable and enabled for RIP mode, the temporary Loop Master receives a RCVbb. The Loop Master shall discard the RCVbb and transmit a RCVdd, where f = hex ‘FD’.

If an L_Port in the OPEN-INIT state that is not the temporary Loop Master receives a RCVdd it shall enter RIP mode and make the transition to the Idle state. If an L_Port receives a CLS, it shall make the transition to the Monitoring state. When the temporary Loop Master receives a RCVdd it shall discard the RCVdd and make the transition to the Idle state. If it receives a CLS, it shall make the transition to the Monitoring state. If the Loop Master does not receive a RCVbb within E_D_TOV and times out, it shall transmit a CLS and make the transition to the Initializing state, since some L_Ports may be in RIP mode and some may not.

If the temporary Loop Master is not RIP capable or not RIP enabled, it shall send a CLS and shall not send a RCVbb or RCVdd. The temporary Loop Master and all other L_Ports that have successfully completed initialization shall make the transition to the Monitoring state.

3.6 Transmission and Reception Rules

3.6.1 Unicast Frame or R_RDY

3.6.1.1 Transmission from Idle State

An L_Port may attempt to transmit whenever it is in the Idle state (i.e., the Bypass FIFO is empty) and it has originated less frames than its Idle Quota since the last SAT visit. In the Idle state the L_Port is in a selective repeat mode with the receiver input is connected to the transmitter output. A transmit request is indicated when any REQ(transmit) loop control is asserted to the LPSM. Transmission onto the Loop may begin after at least six consecutive Idles have been received. This rule prevents an L_Port from injecting its RIP packet in the middle of a cut through RIP packet.

The following conditions must be true before an L_Port begins RIP packet transmission:

- At least six consecutive Idles have been received
- Bypass FIFO is empty
- The Downstream Neighbor BB_Credit is greater than zero.
- Frame Counter <= Idle Quota & RIP packet contains a frame
- REQ(transmit) asserted

When the above conditions are true, the L_Port shall make the transition to the Inject state and the output of the Transmit FIFO shall be connected to the transmitter output on a Transmission Word boundary. The L_Port shall transmit a RCVyx Primitive Signal, where “x” equals the AL_PA of the transmitting L_Port and “y” equals the AL_PA of the intended destination L_Port. Two Fill Words shall immediately follow the RCVyx. A single frame or one or more R_RDYs followed by two trailing Fill Words, a RCVff, and six additional Fill Words shall then be transmitted. The Inject state may be exited when the second Fill Word following the RCVff has been transmitted. If a frame was transmitted (i.e., it was not an R_RDY), the L_Port shall increment the Frame Counter.

For each transmission request, the L_Port shall transmit only a single frame or one or R_RDYs. If the L_Port has multiple frames to transmit (e. g., a sequence), it may transmit one or more RIP packets immediately if the conditions required for transmission are still true. If the L_Port has no more frames to transmit (i.e., REQ(transmit) is deasserted), and if the Bypass FIFO is empty, it shall make the transition to the Idle state. If the Bypass FIFO is not empty, the L_Port shall make the transition to the Drain state.

If a RCVyx Primitive Signal is received, where “y” is equal to the AL_PA of the L_Port, the following RIP packet is addressed to the L_Port. The RCVyx and the RIP packet are removed from the Loop and the frame or R_RDY is stored in a Receive FIFO. All Transmission Words between the RCVyx and the next RCVff inclusive are discarded. RIP packets addressed to the receiving L_Port shall not be stored in the Bypass FIFO.

If a RCVyx Primitive Signal is received, where “y” is not equal to the AL_PA of the L_Port and “x” is equal to the AL_PA of the L_Port, the RIP packet has not been removed by the addressed L_Port (“y” in the RCVyx) and it shall be removed from the Loop.

If a RCVyx Primitive Signal is received, where neither “y” nor “x” is equal to the AL_PA of the L_Port, the RIP packet is not for this L_Port. All Transmission Words between the RCVyx and the next RCVff inclusive shall be placed into the Bypass FIFO. Subsequent Fill Words until the next RCVyx | RCVfx | RCVex are discarded and shall not be placed in the Bypass FIFO.

If additional RCVyx primitives are received where neither “y” nor “x” is equal to the AL_PA of the L_Port, of if a RCVfx or RCVex is received, then all Transmission Words between the RCVyx | RCVfx | RCVex and the RCVff inclusive are placed into the Bypass FIFO.

For each MRKtx received, where x <> AL_PA of the L_Port, it is also stored in the Bypass FIFO, along with up to six trailing Fill Words. This insertion process into the Bypass FIFO may be repeated while in the Inject state. When the L_Port has finished transmitting the RIP packet, the L_Port shall make the transition to the Drain state if the Bypass FIFO is not empty. Otherwise, the L_Port shall make the transition to the Idle state.

In the Drain state, the output of the Bypass FIFO is connected to the transmitter output. Each Transmission Word is shifted out of the Bypass FIFO and transmitted onto the Loop. While in the Drain state, some received Transmission Words may be appended to the Bypass FIFO using the same set of rules as for the Inject state. Since all other received Transmission Words are not loaded into the Bypass FIFO, it will eventually empty and the L_Port shall then make the transition to the Idle state.

3.6.1.2 Transmission in the Drain State

An L_Port may attempt to transmit whenever it is in the Drain state (i.e., the Bypass FIFO is not empty), it has originated less frames than it’s Hold Quota since the last SAT visit, and it is currently holding the SAT token. A transmit request is indicated when any REQ(transmit) loop control is asserted to the LPSM. Before transmission can begin, the L_Port must exhaust the available credit to the Upstream Neighbor by withholding the transmission of R_RDYs, even if receive buffers are available. This enables an L_Port to originate a RIP packet even if the Bypass FIFO is full. Transmission onto the Loop may begin after the Upstream Neighbor BB_Credit is zero and at least six consecutive Idles have been *re-transmitted* by the requesting L_Port. The later rule prevents an L_Port from injecting a RIP packet in the middle of a RIP packet being re-transmitted from the Bypass FIFO.

The following conditions must be true before an L_Port begins RIP packet transmission in the Drain state:

- At least six consecutive Idles have been re-transmitted (from the Bypass FIFO)
- Frame Counter <= Hold Quota
- Holds the SAT token
- The Upstream Neighbor BB_Credit is zero.
- The Downstream Neighbor BB_Credit is greater than zero.
- REQ(transmit) asserted

When the above conditions are true, the L_Port shall make the transition to the Inject state and the output of the Transmit FIFO shall be connected to the transmitter output on a Transmission Word boundary. The L_Port shall transmit a RCVyx Primitive Signal, where “x” equals the AL_PA of the transmitting L_Port and “y” equals the AL_PA of the intended destination L_Port. Two Fill Words shall immediately follow the RCVyx. A single frame one or more R_RDYs followed by two trailing Fill Words, a RCVff, and six additional Fill Words shall then be transmitted.

After the RCVff trailer is transmitted, the L_Port shall increment the Frame Counter if the transmitted RIP packet contained a frame. If the Frame Counter is less than or equal to the Hold Quota and REQ(transmit) is still asserted (there are additional frames or R_RDYs to transmit), the L_Port shall re-enter the Inject state and transmit another frame or one or more R_RDYs. If the Frame Counter is greater than the Hold Quota, the L_Port shall reset the Frame Counter and retransmit the SAT token. It shall then transition to the Drain state. The L_Port shall also replenish the Upstream Neighbor BB_Credit if receive buffers are available.

3.6.2 Broadcast Replicate

3.6.2.1 Transmission in the Idle State

If an L_Port needs to send a frame to every other L_Port on the Loop, it shall indicate this request by asserting REQ(transmit fx) to the LPSM. If the Bypass FIFO is empty, the Downstream Neighbor BB_Credit is greater than zero, and the Frame Counter is less than or equal to the Idle Quota, the L_Port may begin transmission if at least six consecutive Idles have been received. The L_Port shall then transmit a RCVfx Primitive Signal, where f = hex ‘FF’ and x = AL_PA of the L_Port. The L_Port shall then transmit two Fill Words and a single frame, followed by two Fill Words, a RCVff Primitive Signal and six trailing Fill Words. The frame shall be transmitted using class 3 SOF delimiter and R_RDYs are suppressed for frames received via a RCVfx. The L_Port shall increment the Frame Counter if a frame was transmitted.

If the L_Port has additional frames to transmit, it may continue to transmit frames as long as the Bypass FIFO remains empty and it’s Frame Counter is less than or equal to it’s Idle Quota. Each frame is transmitted using the same Transmission Word sequence (i.e., a RCVfx, 2 Fill Words, one frame, a RCVff, and six Fill Words). If the Bypass FIFO is not empty at the end of frame transmission and the L_Port is not holding the SAT token, the L_Port shall make the transition to the Drain state before it transmits additional frames. If the Bypass FIFO is empty and REQ(transmit) is deasserted, the L_Port shall make the transition to the Idle state.

If the L_Port receives a RCVfx, where x = AL_PA of the L_Port and f = hex ‘FF’, the L_Port shall discard all Transmission Words between the RCVyx and the RCVff inclusive. If the L_Port is in the Idle state, it shall transmit the Current Fill Word for each Transmission Word removed from the Loop.

If the L_Port receives a RCVfx, where x <> AL_PA of the L_Port and f = hex ‘FF’, the L_Port is a receiver of a broadcast frame. The L_Port shall cause retransmission of the RCVfx. The L_Port shall also copy the encapsulated frame to the Receive FIFO and it shall also cause retransmission of the frame. The phrase “cause retransmission” has the following interpretation, depending on the current state of the LPSM:
- If the LPSM is in the Idle state, the RIP packet will be retransmitted immediately.
- If the LPSM is in the Inject state, the RIP packet will be stored in the Bypass FIFO for later

retransmission when the L_Port enters the Drain state.
- If the LPSM is in the Drain state, the RIP packet will be appended to the Bypass FIFO for retransmission.

If the L_Port is in the Idle state, it shall substitute the Current Fill Word when required.

3.6.2.2 Transmission in the Drain State

The L_Port may transmit a broadcast frame if it has posted broadcast frame request - REQ(transmit fx), it's Frame Counter is less than it's Hold quota, it is holding the SAT token, its Upstream Neighbor BB_Credit has been depleted to zero, and its Downstream Neighbor BB_Credit is greater than zero. These rules are identical for transmitting a Unicast frame (see Section 2.6.1.2).

3.6.3 Selective Replicate

3.6.3.1 Transmission in the Idle State

If an L_Port needs to send a frame to a subset (up to N -1) of L_Ports on the Loop, it shall indicate this request by asserting REQ(transmit ex, group) to the LPSM. The "group" is a list of AL_PAs (up to 126) that are destinations for the frame transmitted by the L_Port.

If the Bypass FIFO is empty, the Downstream Neighbor BB_Credit is greater than zero, and the Frame Counter is less than or equal to the Idle Quota, the L_Port may begin transmission if at least six consecutive Idles have been received. The L_Port shall then transmit a RCVex Primitive Signal, where x = AL_PA of the L_Port. The L_Port shall then transmit two Fill Words. For each AL_PA in the group, the L_Port shall transmit a RCVye, followed by two Fill Words, where y = AL_PA of each successive AL_PA in the group. After the last RCVye for the group has been transmitted (i.e., RCVye followed by two Fill Words), the L_Port shall transmit a single frame followed by two Fill Words, a RCVff and six trailing Fill Words.

If the L_Port has additional frames to transmit, it may continue to transmit frames as long as the Bypass FIFO remains empty, the Downstream Neighbor BB_Credit is greater than zero, and the Frame Counter is less than or equal to the Idle Quota. Each frame is transmitted using the same Transmission Word sequence (i.e., a RCVex, 2 Fill Words, up to N-1 (RCVye and 2 Fill Words), one frame, two Fill Words, a RCVff and six trailing Fill Words). If the Bypass FIFO is not empty at the end of any frame and the L_Port is not holding the SAT token, or if the L_Port's Frame Counter is greater than it's Hold quota, the L_Port shall make the transition to the Drain state before it transmits additional frames.

If the L_Port receives a RCVex, where x = AL_PA of the L_Port, the L_Port shall discard all Transmission Words between the RCVex and the RCVff inclusive. If the L_Port is in the Idle state, it shall transmit the Current Fill Word for each Transmission Word removed from the Loop.

If the L_Port receives a RCVex, where x <> AL_PA of the L_Port, then the L_Port shall cause retransmission of the RCVex.

If the L_Port receives a RCVye, where y = AL_PA of the L_Port, it shall cause retransmission of the RCVye. The L_Port shall copy the encapsulated frame to the Receive FIFO of the L_Port and it shall also cause retransmission of RIP packet. For any other RCVye (i.e., y <> AL_PA), the L-Port shall cause retransmission of the RIP packet (all transmissions words between the RCVye and RCVff inclusive.

If the L_Port is in the Idle state, it shall substitute the Current Fill Word when required.

3.6.3.2 Transmission in the Drain State

The L_Port may transmit a multicast frame if it has posted a multicast frame request - REQ(transmit ex, group), it's Frame Counter is less than it's Hold quota, and it is holding the SAT token. The rules are identical for transmitting a Unicast frame (see Section 2.6.1.2).

3.7 Fairness

An L_Port is blocked from transmitting a RIP packet when the Bypass FIFO is not empty. If an L_Port cannot empty the Bypass FIFO after an extended period of time, then it must suppress the injection of new RIP packets into the Loop by other L_Ports. This causes the transmission of Fill Words by other L_Ports, which allows its Bypass FIFO to drain. The SAT algorithm described in this section is employed to equally share loop bandwidth between originators and achieve fairness.

3.7.1 SAT

The origination of RIP packets is controlled by the SAT algorithm where SAT refers to Satisfied. The SAT algorithm helps balance between spatial reuse of the individual links and any L_Port being starved of bandwidth.

3.7.2 Terminology

The Hold Quota is used to determine how many RIP packets an L_Port may originate while holding cut through RIP packets in the Bypass FIFO. The Idle Quota is used to determine the number of frames an L_Port may originate if no cut through traffic exists (i.e., the Bypass FIFO is empty). The Reflection Flag is one of the factors used to instruct the L_Port to reflect SAT and SAT' Primitive Signals, breaking the loop or string into two SAT regions. The Frame Counter is a counter that counts the number of frames originated since the last time the SAT Primitive Signal was forwarded or reflected by the L_Port. The origination of R_RDYs does not change the value of the Frame Counter. Forwarding or reflecting a SAT Primitive Signal sets the Frame Counter to zero. Originated means that the trailing RCVff of an originating RIP packet has been sent. Satisfied refers to the condition of an L_Port when either of the two following conditions is met: 1) The L_Port has originated at least Hold Quota number of frames since it previously forwarded or reflected a SAT Primitive Signal; 2) The L_Port has no frames to originate when holding the SAT Primitive Signal.

3.7.3 Reflection versus forwarding

If the Reflection Flag is cleared, an L_Port shall forward SAT and SAT' Primitive Signals. If the Reflection Flag is set, an L_Port shall reflect SAT and SAT' Primitive Signals.

An L_Port shall forward a SAT Primitive Signal after receiving a SAT Primitive Signal by holding it according to the rules of the SAT algorithm and then originating a SAT Primitive Signal. A dual port Node shall unconditionally forward a SAT' Primitive Signal after receiving a SAT' Primitive Signal.

When the reflection flag is set, an L_Port shall reflect a SAT Primitive Signal after receiving a SAT Primitive Signal by holding it according to the rules of the SAT algorithm and then generating a SAT' Primitive Signal on the outbound link of the same L_Port that received the SAT Primitive Signal. A SAT' Primitive Signal is reflected unconditionally by generating a SAT Primitive Signal on the outbound link of the same L_Port that received the SAT' Primitive Signal.

3.7.4 The SAT algorithm

The SAT algorithm will operate correctly on the following topologies:

- single loop
- dual loops rotating in the same direction (Iso loops)
- dual loops rotating in the opposite direction (counter rotating)
- strings (counter rotating loops where the two end Nodes and single port Nodes)

The Node at the end of a String may be a single port Node, a dual port Node with one port not operational or a switch or Fabric Node (a dual port Node with both ports operational is not at the end of a string by definition).

3.7.4.1 Counter rotating loops

The SAT algorithm is described initially for a loop of dual port Nodes, with enhancements for a String shown in the next section. Frames entering a loop are controlled by a SAT Primitive Signal that circulates in the opposite direction to the frames that it controls, as shown in Figure 4. Circulating the SAT Primitive Signal in the opposite direction maximizes spatial reuse.

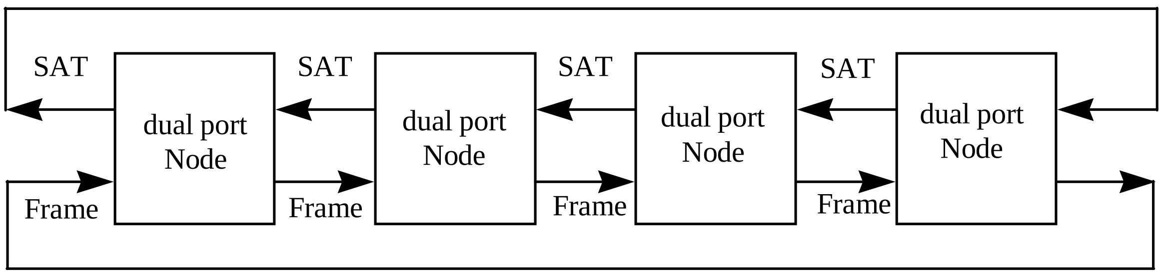

Figure 4: SAT operation on a dual, counter rotating loop

The SAT algorithm is applied independently to clockwise and counter-clockwise traffic. Therefore there are two SAT Primitive Signals circulating on the loop, one in each direction.

When an L_Port receives a SAT Primitive Signal it forwards or reflects the Primitive Signal via the other L_Port on the same Node to the next Node if it is Satisfied. Otherwise the L_Port holds the Primitive Signal until it is Satisfied. Then it forwards or reflects the SAT Primitive Signal and sets the Frame Counter to zero.

R_RDYs are not controlled by the SAT algorithm and, subject to the link protocol, they may be sent at any time. An L_Port may originate a frame, subject to the link protocol, under either of the following conditions: a) The L_Port is holding the SAT Primitive Signal and the L_Port is not Satisfied; b) There is no frame waiting to be forwarded from another L_Port, and the Frame Counter is less than the Idle Quota.

The ability of an L_Port to originate frames is broken into two cases (a and b), and one case (c) not able to originate frames: a) Frame Counter <= Hold quota. Before the SAT Primitive Signal arrives, the L_Port shall originate frames only when there are no cut through frames pending (the Bypass FIFO is empty). After the SAT Primitive Signal arrives, the L_Port shall hold RIP packets in the Bypass FIFO until Satisfied, then the SAT Primitive Signal is forwarded or reflected and the Frame Counter shall be set to zero; b) Frame Counter > Hold quota and Frame Counter <= Idle Quota. Before the SAT Primitive Signal arrives, the L_Port shall originate frames only when there are no cut through RIP packets pending (the Bypass FIFO is not empty). When the SAT Primitive Signal is received, it shall be forwarded or reflected and the Frame Counter shall be set to zero; c) Frame Counter > Idle Quota. The L_Port shall not originate any frames. When the SAT Primitive Signal is received, it shall be forwarded or reflected, the Frame Counter shall be set to zero and frames may be originated according to the SAT algorithm.

3.7.4.2 Strings

The extensions to the algorithm for a String are as follows: a) If an L_Port at the end of a String receives a SAT Primitive Signal then it reflects the SAT Primitive Signal as a SAT' Primitive Signal traveling in the opposite direction after it is Satisfied (see Figure 5). b) A dual port Node with both ports operational forwards SAT' received from either port unconditionally. If an L_Port at the end of a String receives a

SAT' Primitive Signal then it reflects the SAT' character as a SAT Primitive Signal traveling in the opposite direction immediately without resetting the Frame Counter.

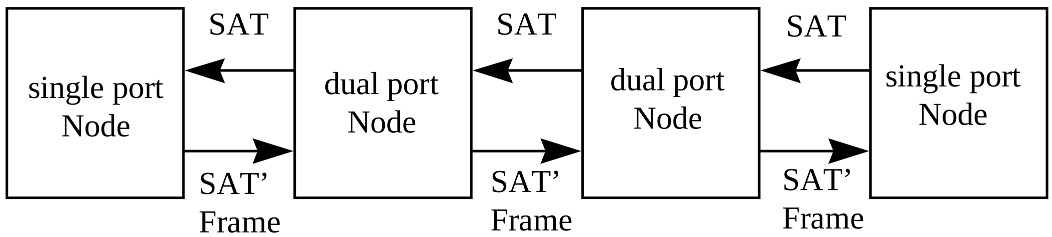

Figure 5: SAT/SAT’ on a String

The reflection of SAT as SAT' Primitive Signal and SAT' as SAT Primitive Signal is dynamic. If a String of dual port nodes is connected into a closed loop then SAT' Primitive Signal are no longer generated. To avoid a previous SAT' Primitive Signal circulating indefinitely a dual port Node with both ports operational shall discard SAT' Primitive Signals as follows: a) a SAT' Primitive Signal received on port 1 is discarded if no SAT Primitive Signal has been received on port 2 since the previous SAT' Primitive Signal was received on port 1; b) a SAT' Primitive Signal received on port 2 is discarded if no SAT Primitive Signal has been received on port 1 since the previous SAT' Primitive Signal was received on port 2.

3.7.4.3 Single Loops

On single loop topologies, the SAT token must flow in the same direction as frames and RIP packets. All other aspects of the SAT algorithm operation are identical.

3.7.4.4 Dual Iso Loops

A dual, iso loop configuration is identical to a single loop topology with respect to operation of the SAT algorithm. The dual loops operate as independent single loops.

3.7.5 Timing requirements of the SAT algorithm

All ports shall incorporate a SAT TIME-OUT timer to detect loss of the SAT Primitive Signal due to a link error. The SAT TIME-OUT timer is set to zero when the port forwards a SAT Primitive Signal or reflects a SAT Primitive Signal as a SAT’ Primitive Signal. If the SAT TIME-OUT timer counts 100 ms, then a SAT Primitive Signal is created and treated as if it has been received. Multiple SAT Primitive Signals circulating in the same direction result if several ports generate a new SAT Primitive Signal simultaneously. However the fairness algorithm still operates correctly and the Primitive Signals eventually merge into a single Primitive Signal as described below.

If an L_Port receives a SAT Primitive Signal while it is holding a SAT then it discards each new SAT Primitive Signal. Each of the following functions shall not introduce a delay exceeding six Transmission Word periods per node when the Bypass FIFO is empty: a) Forwarding a SAT Primitive Signal when the port is Satisfied, subject to the delay described in the paragraph below; b) Reflecting a SAT Primitive Signal as a SAT' Primitive Signal when the port is Satisfied; c) Forwarding a SAT' Primitive Signal; d) Reflecting a SAT' Primitive Signal as a SAT Primitive Signal.

For a loop or String with only a few nodes the SAT Primitive Signal rotates very rapidly if all Nodes are Satisfied. Therefore L_Ports shall perform the following two delays: a) An L_Port shall not forward a SAT Primitive Signal until at least 100 Transmission Word periods have elapsed since it forwarded the previous SAT Primitive Signal; b) An L_Port shall not reflect a SAT Primitive Signal as a SAT' Primitive Signal until at least 100 Transmission Word periods have elapsed since it reflected the previous SAT Primitive Signal.

3.7.6 Configuring the quotas

The Idle Quota and Hold Quota values for each L_Port are allocated by the loop initialization master during loop initialization. The optimum values depend on the total number of Nodes in each String or loop and the traffic pattern. This clause describes how the values are determined. The SAT ROTATION time is the time taken by the SAT character to circulate once around a String or loop, as measured in Transmission Word periods. SAT ROTATION time is a minimum when all L_Ports are Satisfied and they forward the SAT Primitive Signal immediately (SAT' Primitive Signals are always forwarded immediately). SAT ROTATION time increases when the Network becomes heavily loaded and ports hold the SAT Primitive Signal.

The SAT DELAY time is the minimum time to propagate or reflect SAT or SAT' Primitive Signal through a single Node whose inbound port is Satisfied and the Bypass FIFO is empty, as measured in Transmission Word periods.

For a String with N nodes the minimum SAT ROTATION time is 2 * N * SAT DELAY Transmission Words. The corresponding value for a loop is only N * SAT DELAY Transmission Words since the propagation delay for SAT' Primitive Signal is eliminated. SAT ROTATION time is always greater than 100 Transmission Word periods. However after a link failure the loop may become a String and the minimum SAT ROTATION time doubles. Therefore it is advisable to always allow for a minimum SAT ROTATION time of 2 * N * SAT DELAY time.

For long Strings SAT DELAY time is a major contributor to SAT ROTATION. Consequently SAT DELAY time has been specified as six Transmission Words maximum. To avoid throttling when a single node wants to originate data frames, Idle Quota >= (SAT ROTATION time / FRAME LENGTH) >= (2 * N * SAT DELAY time / FRAME LENGTH). For the extreme case of a String with 240 nodes this requires a minimum Idle Quota of 9 frames.

In general there is a compromise between fairness and spatial reuse. Setting Idle Quota =Hold Quota guarantees perfect fairness but it limits spatial reuse. Choosing Idle Quota » Hold Quota permits maximum spatial reuse and therefore maximum total throughput. However in some scenarios this favors the active Node that is furthest upstream since cut through traffic normally has priority. Therefore the upstream Node is able to originate Idle Quota frames for each rotation of the SAT Primitive Signal, whereas each down-stream Node only originates Hold Quota frames. A reasonable compromise for most applications is Idle Quota = 4 * Hold Quota.

3.8 RIP Credit Model

RIP requires a new BB_Credit model that is different from the FC-AL1 BB_Credit model. Since there is no OPN and CLS between L_Ports to establish credit, BB_Credit between communicating L_Ports must be established in advance. One approach would require an L_Port to establish BB_Credit with all L_Ports it may communicate with. This would require a significant number of buffers in an L_Port. To reduce this amount of buffering, RIP employs a Nearest Neighbor credit model, where credit is established only with an L_Port's upstream neighbor Node. In effect, the Nearest Neighbor credit model may be envisioned as link level flow control with routing capability.

R_RDYs encapsulated in RIP packets never consume buffer space and therefore are not flow controlled. They always cut through all intermediate nodes between the source and destination. However, RIP packets containing R_RDYs may be temporarily buffered in the Bypass FIFO. They are processed in hardware at the destination node. As a result, RIP packets containing R_RDYs do not themselves generate an R_RDY response from the DSN. This requires an L_Port to examine the contents of a RIP packet with a RCVyx header to determine whether it contains a frame or R_RDY(s)[1]. Note that a RIP packet with a RCVfx or RCVex can only contain frames - R_RDYs cannot be broadcast or multicast. If a RIP packet contains one or more R_RDYs the DSN shall not originate an R_RDY response to the USN.

Frames encapsulated in RIP packets are flow controlled. As a result all intermediate Nodes on the path between a source and destination Node shall generate a RIP packet containing one or more R_RDYs to it's USN. For example, on a loop with N Nodes, the average distance between a source and destination for a uniform traffic load is N/2. As a result, N/2 - 1 RIP packets containing R_RDYS will circulate the loop, while the RIP packet containing the frame will traverse only the links between the source and destination Nodes.

3.8.1 Single Loop Topology

This credit model is illustrated for a single loop in Figure 6. Before an L_Port may originate a RIP packet containing a frame it must have positive BB_Credit with it's Downstream Neighbor. The Downstream Neighbor or DSN is the Node at the receive end of an L_Port's output link. The Upstream Neighbor or USN is the Node at the transmit end of an L_Port's input link. When the DSN has freed a buffer for a RIP packet, it shall transmit a RIP packet encapsulating an R_RDY with a RCVyx header, where y = AL_PA of the USN. The AL_PA of the USN and DSN are acquired from the LIRP/LILP AL_PA position map transmitted during loop initialization.

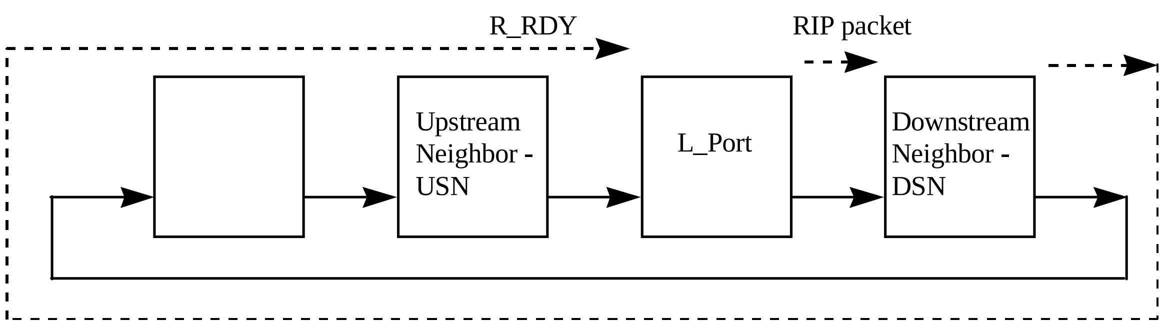

Figure 6: Nearest Neighbor Credit Model

3.8.2 Dual Loop Topologies

3.8.2.1 Iso Rotating Rings

On a dual loop topology with dual port Nodes, where both loops rotate in the same direction (i.e., iso), an R_RDY may be originated on either port by the DSN. Since both loops present an identical path between source and destination, there is no inherent performance advantage in choosing one path over the other.

[1] An alternative implementation may employ a different Primitive Signal to distinguish between a RIP packet containing frames or R_RDYs. For example, a RCVyx could employ two different second characters in the Primitive Signal to distinguish frames from R_RDYs.

3.8.2.2 Counter Rotating Rings

On a dual loop topology with counter rotating loops, an R_RDY may be originated from either port. However, by employing shortest path routing, an R_RDY will only traverse a single link to the L_Port's USN. This is illustrated in Figure 7.

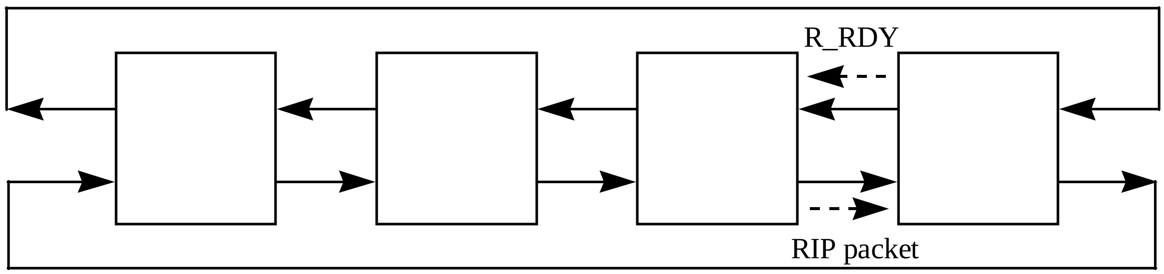

Figure 7: Flow control on counter rotating loops

3.9 RIP Changes to Loop Port State Machine (LPSM)

RIP requires three new states and seven state transitions in the LPSM, which are shown in Figure 8. After RIP mode has been negotiated all LPSMs enter the Idle state. A LPSM shall be in the Idle state when it has neither frames nor R_RDYs to transmit and the Bypass FIFO is empty.

From the Idle state, when the LPSM has a frame or R_RDY to transmit is shall assert REQ(transmit), and if at least six consecutive Idles have been received and the L_Port's Frame Counter is less than or equal to it's Idle Quota, the LPSM shall make the transition to the Inject state. If neither frames nor R_RDYs destined for other ports nor the MRKtx Primitive Signal are received while the LPSM is in the Inject state, the Bypass FIFO will remain empty and the LPSM shall make the transition to the Idle state if it has no more frames or R_RDYs to transmit. If the Bypass FIFO is empty and the LPSM has additional frames or R_RDYs to transmit (REQ(transmit) is reasserted), it may remain in the Inject state.

If any frames or R_RDYs destined for other ports are received while transmitting, the LPSM shall make the transition to the Drain state after the frame or R_RDY and six subsequent Fill Words have been transmitted in the Inject state. When the Bypass FIFO has been drained of all Transmission Words (i.e., it is empty), the LPSM shall make the transition to the Idle state if REQ(transmit) is no longer asserted; or if REQ(transmit) is asserted and the L_Port's Frame Counter is greater than it's Idle Quota; or if REQ(transmit) is asserted and the L_Port's Frame Counter is greater than it's Hold Quota and it is holding the SAT token.

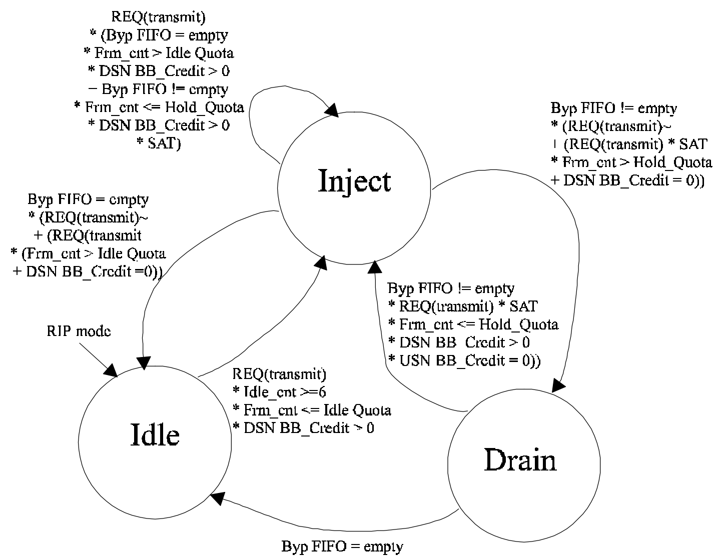

Figure 8: RIP LPSM

3.9.1 Idle State actions:

If Idle is received, the current Fill Word shall be set to Idle. The CFW shall be transmitted.

The LPSM shall make a transition to the Inject state when the L_Port requests transmission (REQ(transmit)), the Downstream Neighbor BB_Credit is greater than zero, the Frame Counter is less than or equal to the Idle Quota, and at least six consecutive Idles have been received.

If a Fill Word is to be transmitted, the current Fill Word shall be used.

If RCVfx is received, where x <> AL_PA of the L_Port, the LPSM shall retransmit the received RCVfx and all transmission words up to the RCVff inclusive (i.e., the L_Port shall retransmit the RIP packet). The LPSM shall also pass the encapsulated frame to the Receive FIFO.

If RCVfx is received, where x = AL_PA of the L_Port, the LPSM shall discard all received Transmission Words between RCVfx and the subsequent RCVff inclusive. The L_Port shall substitute the Current Fill Word for each Transmission Word discarded.

If RCVex is received, where x <> AL_PA of the L_Port, the LPSM shall retransmit the RCVex.

If RCVex is received, where x = AL_PA of the L_Port, the LPSM shall discard all received Transmission Words between RCVex and the subsequent RCVff inclusive. The L_Port shall substitute the Current Fill Word for each Transmission Word discarded.

If RCVye is received, with y <> AL_PA of the L_Port, then the LPSM shall retransmit the received RCVye. The LPSM shall also retransmit the encapsulated frame.

If RCVye is received, where y = AL_PA, the LPSM shall substitute the CFW. The L_Port shall also pass the encapsulated frame to the Receive FIFO and shall also retransmit the encapsulated frame.

If RCVyx is received, where y = AL_PA of the L_Port, then the LPSM shall pass the encapsulated frame or R_RDY(s) to the Receive FIFO. The LPSM shall remove all received Transmission Words between RCVyx and the subsequent RCVff inclusive. The L_Port shall substitute the Current Fill Word for each Transmission Word removed.

If RCVyx is received, where y <> AL_PA of the L_Port and x = AL_PA of the L_Port, then the LPSM shall discard all received Transmission Words between RCVyx and the subsequent RCVff inclusive. The L_Port shall substitute the Current Fill Word for each Transmission Word discarded.

If a RCVyx is received, where y <> AL_PA and x <> AL_PA, the LPSM shall retransmit all Transmission Words received between the RCVyx and the RCVff inclusive.

If MRKtx is received:
- if x = AL_PA of the L_Port, the LPSM shall transmit the current Fill Word; the MRKtx is discarded;
- if x <> AL_PA of the L_Port and the MK_TP and AL_PS match the expected values, the MRKtx shall be loaded into the Receive FIFO and the received MRKtx shall be retransmitted.

If SAT is received, the Frame Counter shall be set to zero and the SAT shall be retransmitted.

If SAT' is received and the Node is not a single port device, it shall be retransmitted.

If SAT' is received and the Node is a single port device, it shall discard the SAT' and transmit SAT.

If the LPSM requests to transmit a MRKtx (REQ(mark as tx)), the LPSM shall transmit one MRKtx at the next appropriate Fill Word (see clause 7), unless REQ(mark tx) is removed before MRKtx is transmitted.

If CLS is received, the current Fill Word shall be transmitted.

If OPN is received, the current Fill Word shall be transmitted.

If ARB is received, the current Fill Word shall be transmitted.

If LIP is received, the LPSM shall make the transition to the OPEN-INIT state.

IF LPByx is recognized:
- if x = AL_PA of the L_Port, the received LPByx shall be discarded and the Current Fill Word shall be transmitted;
- if y = AL_PA of the L_Port, or the L_Port requests to be bypassed (REQ(bypass L_Port)), the LPSM shall retransmit the LPByx, shall set the bypass circuit (if present), shall set LP_BYPASS to True (1), and shall go to nonparticipating mode in the Idle state;
- if y <> AL_PA of the L_Port, the LPByx shall be retransmitted.

If LPEyx is recognized:
- if x = AL_PA, the received LPEyx shall be discarded and the Current Fill Word shall be transmitted;
- if y <> AL_PA and y <> hex 'FF', retransmit the LPEyx;
- if y = AL_PA or y = hex 'FF', set LP_BYPASS = FALSE(0) and retransmit the LPEyx.

The LPSM shall retransmit all other received valid Transmission Words on the Loop.

Invalid Transmission Character substitution shall be performed as specified in 8.3.2.

If the LPSM detects a Loop failure on its inbound fibre and LP_BYPASS is set to FALSE (0), or the L_Port requests initialization (REQ(initialize)), the LPSM shall make the transition to the Initializing state.

If the L_Port requests not to participate on the Loop (REQ(nonparticipating)), the LPSM shall transmit at least 12 LIPs with the rightmost two characters equal to hex 'F7F7' to invoke Loop initialization. This allows another L_Port to acquire the relinquished AL_PA. The 12 LIPs are only transmitted once for each REQ(nonparticipating). The L_Port shall not participate further in Loop initialization until REQ(initialize) or REQ(participating) is set.

3.9.2 Inject State actions:

If Idle is received, the Current Fill Word shall be set to Idle.

If a Fill Word is to be transmitted, the Current Fill Word shall be used.

If a Fill Word is to stored in the Bypass FIFO, the Current Fill Word shall be used.

If a Fill Word is received that is not stored in the Bypass FIFO, it shall be discarded.

If the L_Port requests additional transmission (REQ(transmit) asserted), the Bypass FIFO is empty, the Downstream Neighbor BB_Credit is greater than zero, and the Frame Counter is less than or equal to the Idle Quota, the LPSM shall remain in the Inject state.

If the L_Port requests additional transmission (REQ(transmit) asserted), the Bypass FIFO is not empty, the Downstream Neighbor BB_Credit is greater than zero, the Frame Counter is less than or equal to the Hold Quota, and the SAT token is held, the LPSM shall remain in the Inject state.

After a RIP packet is transmitted by the L_Port (RCVff is transmitted), the L_Port shall increment the Frame Counter.

If RCVfx is received, where x <> AL_PA of the L_Port, the LPSM shall load into the Bypass FIFO all Transmission Words between the RCVfx and RCVff inclusive. The LPSM shall also pass the frame encapsulated in the RIP packet to the Receive FIFO.

If RCVfx is received, where x = AL_PA of the L_Port, the LPSM shall discard all received Transmission Words between RCVfx and the subsequent RCVff inclusive. The L_Port shall substitute the Current Fill Word for each Transmission Word discarded.

If RCVex is received, where x <> AL_PA of the L_Port, the LPSM shall load into the Bypass FIFO all Transmission Words between the RCVex and RFVff inclusive.

If RCVex is received, where x = AL_PA of the L_Port, the LPSM shall discard all received Transmission Words between RCVex and the subsequent RCVff inclusive. The L_Port shall substitute the Current Fill Word for each Transmission Word discarded.

If RCVye is received, where y = AL_PA, the LPSM shall substitute the CFW and load it into the Bypass FIFO. The L_Port shall also pass the encapsulated frame to the Receive FIFO.

If RCVyx is received, where y = AL_PA of the L_Port, then the LPSM shall pass the encapsulated frame or R_RDY(s) to the Receive FIFO. The LPSM shall remove all received Transmission Words between RCVyx and the subsequent RCVff inclusive.

If RCVyx is received, where y <> AL_PA of the L_Port and x = AL_PA of the L_Port, then the LPSM shall discard all received Transmission Words between RCVyx and the subsequent RCVff inclusive.

If a RCVyx is received, where y <> AL_PA and x <> AL_PA, the LPSM shall load into the Bypass FIFO all Transmission Words between the RCVyx and RFVff inclusive.

If CLS is received, it shall be discarded.

If OPN is received, it shall be discarded.

If ARB is received, it shall be discarded.

If SAT is received, it shall be loaded into the Bypass FIFO. The Frame Counter shall be set to zero.

If SAT’ is received and the Node is a dual port device, it shall be loaded into the Bypass FIFO.

If SAT’ is received and the Node is a single port device, it shall be discarded and SAT shall be loaded into the Bypass FIFO.

If a MRKtx is received, where x = AL_PA of the L_Port, the LPSM shall discard the MRKtx.

If a MRKtx is received where MK_TP and AL_PS match the expected values, and x <> AL_PA, the MRKtx shall be processed. The MRKtx and up to 6 trailing Fill Words that are received shall be buffered in the Bypass FIFO. The MRKtx shall also be placed in the Receive FIFO.

If LIP is received, the LPSM shall make the transition to the OPEN-INIT state.

If LPByx is recognized:
- if x = AL_PA of the L_Port, the received LPByx shall be discarded;
- if y = AL_PA of the L_Port, or the L_Port requests to be bypassed (REQ(bypass L_Port)), the LPSM shall finish the current frame or R_RDY transmission; set LP_BYPASS to TRUE (1); make the transition to the Drain state if the Bypass FIFO is not empty; or make the transition to the Idle state if the Bypass FIFO is empty.

If the LPSM detects a Loop failure on its inbound fibre or the L_Port requests initialization (REQ(initialize)), the LPSM shall make the transition to the Initializing state.

If the LPSM requests to transmit a MRKtx (REQ(mark as tx)), the LPSM shall transmit one MRKtx at the next appropriate Fill Word (see clause 7), unless REQ(mark tx) request is removed before MRKtx is transmitted.

3.9.3 Drain State actions:

If Idle is received, the current Fill Word shall be set to Idle.

If a Fill Word is to be transmitted, the Current Fill Word shall be used.

If a Fill Word is to stored in the Bypass FIFO, the Current Fill Word shall be used.

If a Fill Word is received that is not stored in the Bypass FIFO, it shall be discarded.

If the L_Port requests transmission (REQ(transmit) asserted), the SAT token is held, the Frame Counter is less than or equal to the Hold Quota, the Upstream Neighbor BB_Credit is greater than zero, the Downstream Neighbor BB_Credit is zero, and six Idle characters have been transmitted from the head of the Bypass FIFO, the LPSM shall make the transition to the Inject state.

If RCVfx is received, where x <> AL_PA of the L_Port, the LPSM shall load into the Bypass FIFO all Transmission Words between the RCVfx and RFVff inclusive. The LPSM shall also pass the frame encapsulated in the RIP packet to the Receive FIFO.

If RCVfx is received, where x = AL_PA of the L_Port, the LPSM shall discard all received Transmission Words between RCVfx and the subsequent RCVff inclusive. The L_Port shall substitute the Current Fill Word for each Transmission Word discarded.

If RCVex is received, where x <> AL_PA of the L_Port, the LPSM shall load into the Bypass FIFO all Transmission Words between the RCVex and RFVff inclusive.

If RCVex is received, where x = AL_PA of the L_Port, the LPSM shall discard all received Transmission Words between RCVex and the subsequent RCVff inclusive. The L_Port shall substitute the Current Fill Word for each Transmission Word discarded.

If RCVye is received, where y = AL_PA, the LPSM shall substitute the CFW and load it into the Bypass FIFO. The L_Port shall also load encapsulated frame into the Receive FIFO.

If RCVyx is received, where y = AL_PA of the L_Port, then the LPSM shall pass the encapsulated frame or R_RDY(s) to the Receive FIFO. The LPSM shall remove all received Transmission Words between RCVyx and the subsequent RCVff inclusive.

If RCVyx is received, where y <> AL_PA of the L_Port and x = AL_PA of the L_Port, then the LPSM shall discard all received Transmission Words between RCVyx and the subsequent RCVff inclusive.

If a RCVyx is received, where y <> AL_PA and x <> AL_PA, the LPSM shall load into the Bypass FIFO all Transmission Words between the RCVyx and RFVff inclusive.

If CLS is received, it shall be discarded.

If OPN is received, it shall be discarded.

If ARB is received, it shall be discarded.

If SAT is received, it shall be loaded into the Bypass FIFO. The Frame Counter shall be set to zero.

If SAT’ is received and the Node is a dual port device, it shall be loaded into the Bypass FIFO.

If SAT’ is received and the Node is a single port device, it shall be discarded and SAT shall be loaded into the Bypass FIFO.

If a MRKtx is received where x = AL_PA of the L_Port, the LPSM shall discard the MRKtx.

If a MRKtx is received where MK_TP and AL_PS match the expected values, and x <> AL_PA, the MRKtx shall be processed. The MRKtx and up to 6 trailing Fill Words that are received shall be buffered in the Bypass FIFO. The MRKtx shall be placed in the Receive FIFO.

If LIP is received, the LPSM shall make the transition to the OPEN-INIT state.

If LPByx is recognized:
- if x = AL_PA of the L_Port, the received LPByx shall be discarded;
- if y = AL_PA of the L_Port, or the L_Port requests to be bypassed (REQ(bypass L_Port)), the LPSM shall finish the current frame or R_RDY transmission; set LP_BYPASS to TRUE (1); make the transition to the Drain state if the Bypass FIFO is not empty; or make the transition to the Idle state if the Bypass FIFO is empty.

If the LPSM detects a Loop failure on its inbound fibre or the L_Port requests initialization (REQ(initialize)), the LPSM shall make the transition to the Initializing state.

If the LPSM requests to transmit a MRKtx (REQ(mark as tx)), the LPSM shall transmit one MRKtx at the next appropriate Fill Word (see clause 7), unless REQ(mark tx) request is removed before MRKtx is transmitted.

3.9.4 LPSM State Tables

State table legends:

RFW = Received Fill Word

CFW = Current Fill Word

DSN = Downstream Neighbor

USN = Upstream Neighbor

Table 2: Idle Transitions

Entry Actions			
COPY = 0	RETRANSMIT = 0	CFW = N/C	
LP_BYPASS = 0			
Input	**Output**	**Next State**	notes
loss of sync < R_T_TOV	IDLE	Idle	
loop failure			
LP_BYPASS = 0	none/inst.	Initializing	1
LP_BYPASS = 1	none/inst	Idle	1
Invalid trans. character	any valid trans. Character	Idle	1
running disparity at O.S.	CFW	Idle	
elasticity word required	CFW	Idle	
valid trans. word = O.S.:			
RCVyx: y = AL_PA	Copy = 1, Retransmit = 0		
	CFW	Idle	
RCVyx: y <> AL_PA & x <> AL_PA	Copy = 0, Retransmit = 1		
	same word	Idle	
RCVyx: y <> AL_PA & x = AL_PA	Copy = 0, Retransmit = 0		1
	CFW	Idle	
RCVfx: x <> AL_PA	Copy = 1, Retransmit = 1		
	same word	Idle	
RCVfx: x = AL_PA	Copy = 0, Retransmit = 0		
	CFW	Idle	
RCVex: x <> AL_PA	Retransmit = 1		
	same word	Idle	
RCVex: x = AL_PA	Retransmit = 0		

	CFW	Idle	
RCVye: y <> AL_PA			
Retransmit = 1	same word	Idle	
Retransmit = 0	CFW	Idle	
RCVye: y = AL_PA			
Retransmit = 1	Copy = 1		
	same word	Idle	
Retransmit = 0	CFW	Idle	
SOF | VDW | EOF | R_RDY			
Copy = 1	load Receive FIFO		
Retransmit = 1	same word	Idle	
Retransmit = 0	CFW	Idle	
RCVff	Copy = 0		
Retransmit = 1	Retransmit = 0		
	same word	Idle	
Retransmit = 0	CFW	Idle	
IDLE	CFW = IDLE		
	CFW	Idle	
OPN	CFW	Idle	1
CLS	CFW	Idle	1
ARB	CFW	Idle	
MRKtx			
x = AL_PA	CFW	Idle	
x <> AL_PA	same word	Idle	
SAT			
set Frame Counter = 0			
if dual port Node	SAT	Idle	
if single port Node	SAT’	Idle	
SAT’			
if dual port Node	SAT’	Idle	
if single port Node	SAT	Idle	
primitive sequences			
NOS	same word	Idle	
OLS	same word	Idle	
LR	same word	Idle	

LRR	same word	Idle	
LIP	same word	OPEN-INIT	
LPByx			
x = AL_PA	CFW	Idle	
y <> AL_PA	same word	Idle	
y = AL_PA	LP_BYPASS = 1		
	same word	Idle	
LPEyx | LPEfx			
x = AL_PA	CFW	Idle	
y <> AL_PA & y <> hex ‘FF’	same word	Idle	
y = AL_PA | f = hex ‘FF’	LP_BYPASS = 0		
	same word	Idle	
any other O.S.	same word	Idle	
L-Port controls			
REQ(monitor)	none/inst.	Idle	1
REQ(arbitrate as x)	none/inst.	Idle	1
REQ(transmit i): i = yx | fx | ex group | R_RDY & (DSN BB_Credit = 0 | Frame Counter > Idle Quota)	same word	Idle	
REQ(transmit i): i = yx | fx | ex group | R_RDY & DSN BB_Credit > 0 & Frame Counter <= Idle Quota			
FW_cnt < 6 Fill Words	same word	Idle	
FW_cnt >= 6 Fill Words	CFW = Idle	Inject	
REQ(open yx) f-d	none/inst.	Idle	1
REQ(open yy) h-d	none/inst.	Idle	1
REQ(open fr)	none/inst.	Idle	1
REQ(open yr)	none/inst.	Idle	1
REQ(close)	none/inst.	Idle	1
REQ(transfer)	none/inst.	Idle	1
REQ(old-port)	none/inst.	Idle	1
REQ(participating)	none/inst.	Idle	1
REQ(nonparticipating)	transmit 12 LIP(F7, F7)s	Idle	
REQ(mark as tx)	MRKtx at the next appropriate		
TW = appropriate Fill Word	MRKtx	Idle	
TW <> appropriate Fill Word	none/inst.	Idle	
REQ(bypass L_Port)	LP_BYPASS = 1	Idle	

REQ(bypass L_Port y)	LPByx at the next app. Fill Word	Idle	
REQ(enable L_Port)	none/inst.	Idle	1
REQ(enable all)	LPEfx at the next app. Fill Word	Idle	
REQ(enable L_Port y)	LPEyx at the next app. Fill Word	Idle	
REQ(initialize)	none/inst.	Initializing	

Notes:

1) error

Table 3: Inject Transitions

Entry Actions			
COPY = 0	RETRANSMIT = 0	LP_BYPASS = 0	
CFW = N/C			
Input	**Output**	**Next State**	notes
loss of sync < R_T_TOV	FC-2 FP/PSig/PSeq	Inject	
loop failure	none/inst.	Initializing	
Invalid trans. character	FC-2 FP/PSig/PSeq	Inject	
running disparity at O.S.	FC-2 FP/PSig/PSeq	Inject	
elasticity word required	n/a	Inject	
valid trans. word = O.S.:			
RCVyx: y = AL_PA	Copy = 1, Retransmit = 0		
	FC-2 FP/PSig/PSeq	Inject	
RCVyx: y <> AL_PA & x <> AL_PA	Copy = 0, Retransmit = 1		
	Load Bypass FIFO		
	FC-2 FP/PSig/PSeq	Inject	
RCVyx: y <> AL_PA & x = AL_PA	Copy = 0, Retransmit = 0		1
	FC-2 FP/PSig/PSeq	Inject	
RCVfx: x <> AL_PA	Copy = 0, Retransmit = 1		
	Load Bypass FIFO		
	FC-2 FP/PSig/PSeq	Inject	
RCVfx: x = AL_PA	Copy = 0, Retransmit = 0		
	FC-2 FP/PSig/PSeq	Inject	
RCVex: x <> AL_PA	Retransmit = 1		
	Load Bypass FIFO		
	FC-2 FP/PSig/PSeq	Inject	
RCVex: x = AL_PA	Retransmit = 0		
	FC-2 FP/PSig/PSeq	Inject	
RCVye: y <> AL_PA			
Retransmit = 1	Load Bypass FIFO		
	FC-2 FP/PSig/PSeq	Inject	
RCVye: y = AL_PA			

Retransmit = 1	Copy = 1, Load Bypass FIFO (CFW)		
	FC-2 FP/PSig/PSeq	Inject	
SOF | VDW | EOF | R_RDY			
Retransmit = 1	load Bypass FIFO		
Copy = 1	load Receive FIFO		
RCVff	Copy = 0		
Retransmit = 1	Retransmit = 0, Load Bypass FIFO		
	FC-2 FP/PSig/PSeq	Inject	
IDLE	CFW = IDLE		
	FC-2 FP/PSig/PSeq	Inject	
ARBx:	FC-2 FP/PSig/PSeq	Inject	1
OPN	FC-2 FP/PSig/PSeq	Inject	1
CLS	FC-2 FP/PSig/PSeq	Inject	1
MRKtx			
x = AL_PA	FC-2 FP/PSig/PSeq	Inject	
x <> AL_PA	load Bypass FIFO		
	load Receive FIFO		
	FC-2 FP/PSig/PSeq	Inject	
SAT			
set Frame Counter = 0			
if dual port Node	load SAT into Bypass FIFO	Idle	
if single port Node	load SAT’ into Bypass FIFO	Idle	
SAT’			
if dual port Node	load SAT’ into Bypass FIFO	Idle	
if single port Node	load SAT into Bypass FIFO	Idle	
primitive sequences			
NOS	FC-2 FP/PSig/PSeq	Inject	
OLS	FC-2 FP/PSig/PSeq	Inject	
LR	FC-2 FP/PSig/PSeq	Inject	
LRR	FC-2 FP/PSig/PSeq	Inject	
LIP	FC-2 FP/PSig/PSeq	Initializing	
LPByx			
x = AL_PA	FC-2 FP/PSig/PSeq	Inject	
y <> AL_PA	FC-2 FP/PSig/PSeq	Inject	
y = AL_PA			

Bypass FIFO = empty	LP_BYPASS = 1	Idle	
Bypass FIFO <> empty	LP_BYPASS = 1	Drain	
LPEyx | LPEfx			
x = AL_PA	FC-2 FP/PSig/PSeq	Inject	
y = AL_PA	FC-2 FP/PSig/PSeq	Inject	1
y <> AL_PA	FC-2 FP/PSig/PSeq	Inject	
y = ff	FC-2 FP/PSig/PSeq	Inject	
any other O.S.	FC-2 FP/PSig/PSeq	Inject	
L-Port controls			
REQ(monitor)	none/inst.	Monitoring	
REQ(transmit i): i = yx | fx | ex group | R_RDY			
& Bypass FIFO = empty & Frame Counter <= Idle Quota & DSN BB_Credit > 0	none/inst.	Inject	
REQ(transmit i): i = yx | fx | ex group | R_RDY			
& Bypass FIFO = empty & (Frame Counter > Idle Quota & DSN BB_Credit = 0)	none/inst.	Idle	
REQ(transmit i): i = yx | fx | ex group | R_RDY			
& Bypass FIFO <> empty & Frame Counter <= Hold Quota & SAT token held & DSN BB_Credit > 0 & USN BB_Credit = 0	none/inst.	Inject	
REQ(transmit i): i = yx | fx | ex group | R_RDY			
& Bypass FIFO <> empty & (Frame Counter > Hold Quota | DSN BB_Credit > 0)	none/inst.	Idle	
REQ(open yx) f-d	none/inst.	Inject	
REQ(open yy) h-d	none/inst.	Inject	
REQ(open fr)	none/inst.	Inject	
REQ(open yr)	none/inst.	Inject	
REQ(close)	none/inst.	Inject	
REQ(arbitrate as x)	none/inst.	Inject	1
REQ(transfer)	none/inst.	Inject	
REQ(old-port)	none/inst.	Inject	
REQ(participating)	none/inst.	Inject	1
REQ(nonparticipating)	transmit 12 LIP(F7, F7)s	Inject	
REQ(mark as tx)	none/inst.	Inject	
REQ(bypass L_Port)			

Bypass FIFO = empty	LP_BYPASS = 1	Idle	
Bypass FIFO <> empty	LP_BYPASS = 1	Drain	
REQ(bypass L_Port y)	none/inst.	Inject	
REQ(enable L_Port)	none/inst.	Inject	
REQ(enable all)	none/inst.	Inject	
REQ(enable L_Port y)	none/inst.	Inject	
REQ(initialize)	none/inst.	Initializing	

Notes:

1) error

Table 4: Drain Transitions

	Entry Actions		
COPY = 0	RETRANSMIT = 0	LP_BYPASS = 0	
CFW = N/C			
Input	**Output**	**Next State**	notes
loss of sync < R_T_TOV	FC-2 FP/PSig/PSeq	Drain	
loop failure	none/inst.	Initializing	
Invalid trans. character	FC-2 FP/PSig/PSeq	Drain	
running disparity at O.S.	FC-2 FP/PSig/PSeq	Drain	
elasticity word required	n/a	Drain	
valid trans. word = O.S.:			
RCVyx: y = AL_PA	Copy = 1, Retransmit = 0		
	FC-2 FP/PSig/PSeq	Drain	
RCVyx: y <> AL_PA & x <> AL_PA	Copy = 0, Retransmit = 1		
	Load Bypass FIFO		
	FC-2 FP/PSig/PSeq	Drain	
RCVyx: y <> AL_PA & x = AL_PA	Copy = 0, Retransmit = 0		1
	FC-2 FP/PSig/PSeq	Drain	
RCVfx: x <> AL_PA	Copy = 0, Retransmit = 1		
	Load Bypass FIFO		
	FC-2 FP/PSig/PSeq	Drain	
RCVfx: x = AL_PA	Copy = 0, Retransmit = 0		
	FC-2 FP/PSig/PSeq	Drain	
RCVex: x <> AL_PA	Retransmit = 1		
	Load Bypass FIFO		
	FC-2 FP/PSig/PSeq	Drain	
RCVex: x = AL_PA	Retransmit = 0		
	FC-2 FP/PSig/PSeq	Drain	
RCVye: y <> AL_PA			
Retransmit = 1	Load Bypass FIFO		
	FC-2 FP/PSig/PSeq	Drain	
RCVye: y = AL_PA			

Retransmit = 1	Copy = 1, Load Bypass FIFO (CFW)		
	FC-2 FP/PSig/PSeq	Drain	
SOF | VDW | EOF | R_RDY			
Retransmit = 1	load Bypass FIFO		
Copy = 1	load Receive FIFO		
RCVff	Copy = 0		
Retransmit = 1	Retransmit = 0, Load Bypass FIFO		
	FC-2 FP/PSig/PSeq	Drain	
IDLE	CFW = IDLE		
	ACCESS = TRUE (1)		
	FC-2 FP/PSig/PSeq	Drain	
ARB:	FC-2 FP/PSig/PSeq	Drain	1
OPN	FC-2 FP/PSig/PSeq	Drain	1
CLS	FC-2 FP/PSig/PSeq	Drain	1
MRKtx			
x = AL_PA	FC-2 FP/PSig/PSeq	Drain	
x <> AL_PA	load Bypass FIFO		
	load Receive FIFO		
	FC-2 FP/PSig/PSeq	Drain	
SAT			
if Frame Counter > Hold Quota set Frame Counter = 0			
if dual port Node	load SAT into Bypass FIFO	Drain	
if single port Node	load SAT’ into Bypass FIFO	Drain	
if (Frame Counter <= Hold Quota & REQ(transmit)			
& DSN BB_Credit > 0) hold SAT token	FC-2 FP/PSig/PSeq	Drain	
SAT’			
if dual port Node	load SAT’ into Bypass FIFO	Drain	
if single port Node	load SAT into Bypass FIFO	Drain	
primitive sequences			
NOS	FC-2 FP/PSig/PSeq	Drain	
OLS	FC-2 FP/PSig/PSeq	Drain	
LR	FC-2 FP/PSig/PSeq	Drain	
LRR	FC-2 FP/PSig/PSeq	Drain	
LIP	FC-2 FP/PSig/PSeq	Initializing	

LPByx			
x = AL_PA	FC-2 FP/PSig/PSeq	Drain	
y <> AL_PA	FC-2 FP/PSig/PSeq	Drain	
y = AL_PA			
Bypass FIFO = empty	LP_BYPASS = 1	Idle	
Bypass FIFO <> empty	LP_BYPASS = 1	Drain	
LPEyx | LPEfx			
x = AL_PA	FC-2 FP/PSig/PSeq	Drain	
y = AL_PA	FC-2 FP/PSig/PSeq	Drain	1
y <> AL_PA	FC-2 FP/PSig/PSeq	Drain	
y = ff	FC-2 FP/PSig/PSeq	Drain	
any other O.S.	FC-2 FP/PSig/PSeq	Drain	
L-Port controls			
REQ(monitor)	none/inst.	Drain	
REQ(arbitrate as x)	none/inst.	Drain	1
REQ(transmit i): i = yx | fx | fe group | R_RDY			
FW_cnt < 6 Fill Words	same word	Drain	
FW_cnt >= 6 Idles & Byp FIFO empty & Frame Counter <= Idle Quota & DSN BB_Credit > 0	none/inst.	Inject	
FW_cnt >= 6 Idles & Byp FIFO empty & Frame Counter > Idle Quota | DSN BB_Credit = 0	same word	Idle	
FW_cnt >= 6 Idles & Byp FIFO not empty & Frame Counter <= Hold Quota & DSN BB_Credit > 0 & USN BB_Credit = 0 & SAT token held	none/inst.	Inject	
FW_cnt >= 6 Idles & Byp FIFO not empty | Frame Counter > Hold Quota | DSN BB_Credit = 0 | USN BB_Credit != 0 | SAT token not held	FC-2 FP/PSig/PSeq	Drain	
REQ(open yx) f-d	none/inst.	Drain	
REQ(open yy) h-d	none/inst.	Drain	
REQ(open fr)	none/inst.	Drain	
REQ(open yr)	none/inst.	Drain	
REQ(close)	none/inst.	Drain	
REQ(transfer)	none/inst.	Drain	

REQ(old-port)	none/inst.	Drain	
REQ(participating)	none/inst.	Drain	1
REQ(nonparticipating)	none/inst.	Drain	
REQ(mark as tx)	none/inst.	Drain	
REQ(bypass L_Port)	none/inst.	Monitoring	
REQ(bypass L_Port y)	none/inst.	Drain	
REQ(enable L_Port)	none/inst.	Drain	
REQ(enable all)	none/inst.	Drain	
REQ(enable L_Port y)	none/inst.	Drain	
REQ(initialize)	none/inst.	Initializing	
Notes: 1) error			