Loop Prevention

This specification provides a mechanism to allow the bus to continue to operate in the presence of physical loops in the bus topology. The mechanism works by not allowing connections to become active if activating the connection would create a loop. This is called a 'loop free build' because the topology is built, one connection at a time.

After a node has completed the startup of a port, the port is placed in the 'untested' state. A node with ports in the untested state will select one of the untested ports (the 'test port') and perform a loop test procedure to determine if enabling the port will create a loop. When it is determined that enabling a port will not create loop, the port is placed in the active state.

Only one port on a bus segment may be enabled at time. In order to exclude other ports on the same bus from being tested, a node must win arbitration to take control of the bus. While it has control, it is the only node on that bus that may cause a port on the bus to be enabled.

In order to test for loops, a controlling node sends a uniquely formatted loop test packet (LTP) the format of which is shown below. This packet contains a 6-bit test_value, a mode bit and a single bit generation number. The LTP is sent as a legacy PHY packet at S100.

An LTP is sent on the active bus by the controlling node when there has been a bus reset since the last LTP was sent/received or if the test_value is changed by the controlling node as a result of a 'collision'. The mode of the LTP is set to *test* for the packets sent in the initial test interval.

| 00 | $3F_{16}$ | 00 | type($E_{16}$) | 00 0000 0000 | m | G | test_value |
|---|---|---|---|---|---|---|---|

| Field | Comment |
|---|---|
| m | LTP mode:<br>0 - test<br>1 - attach in progress |
| G | Generation number - alternating value. Whenever new test_value is chosen, the G value of the LTP sent on the bus, is the inverse of the G value that was previously in the test_value holding register. |
| test_value | Loop Test Packet comparison value |

When an LTP is received, the test_value, mode and generation number are captured in the 'test_value holding resister' (HR). If the mode of the LTP is *attach in progress* then the mode will be set back to *test* when a bus reset is received/sent on the active bus, or when any other packet is received on the active bus.

## Loop Test Symbol

The contents of the HR is used in a loop test symbol (LTS). The LTS is any data symbol that follows a data_prefix token that is received on an untested port. This LTS is considered to be valid if the running disparity is valid. The LTS is send repeatedly on the untested ports at the speed of the connection.

## Test Interval

After an LTP is sent, a node waits for a period of time for the LTP to propagate to the furthest extremes of the bus and for an LTS with the same test_value to be received on the test port. This delay is called the test interval and is (TBD, 5-10?)us in duration. It starts whenever an LTP is received/sent on the active bus. If, the node does not have control of the bus then the test interval ends, then the test interval is extended until the node becomes the controlling node.

During the test interval, the controlling node compares the test value and generation number of the LTS received on the test port to the values in the HR. It also checks the mode of the received LTS. If the received LTS mode is *attach in progress* or if the received test_value is greater than the value in the HR, then the test of that port is aborted (note: the generation number is not used to determine if the received test_value is greater that the value in the HR). The node will then select another port to test. The test timer does not need to be reset when a test is aborted. If the node aborting the test is the controlling node, it will release the local bus before selecting another test port.

## Collision

If the test_value and generation number in the LTS received on the test port equals the value in the HR, then a collision condition, and potentially a loop, exists. When a collision condition exists, the node must have control of the bus before it may increment its collision counter. When it has control, it will increment its collision counter, select a new test_value from its random number generator and send a new LTP with a different generation number. If the collision counter reaches (TBD, 5-12?), the port is placed in the loop_disabled state and the bus is released.

When a node becomes the controlling node it will not select a new test port or relinquish the bus as long as the collision condition exists. It will relinquish the bus if/when the port is placed in the loop_disabled state.

While waiting to win arbitration, an LTS may be received on the test port that clears the collision condition. If this occurs, the collision condition does not exist and the node will not increment its collision counter when it wins the bus. However, once a new LTP is sent, the test timer is reset and the test interval must expire before it is determined that no collision exists and the attach can be completed.

## Completing the Attach

If the test interval expires when a node has control of the bus, and no collision exists, then the controlling node may begin the attach process. It starts the process by sending an LTP with the test_value and generation number in the HR and a mode of *attach in progress.* This LTP is sent on the active bus and an LTS with the same data is sent on all untested ports except for the test port. A continuous sequence of attach_request tokens are sent on the test port. When the LTP is sent, the test timer is reset and started.

After sending an attach_request token, the controlling node will wait until a bus_reset token is received on the test port. When the reset is received, the port is placed in the active state and the bus reset is propagated to all active ports.

If a bus reset has not been received when both the maximum occupancy timer and the test timer have both expired, then the port is placed in the active state and bus_reset is sent to all active ports.

If the controlling node receives an attach_request on the test port, it will place the port in the active state and send bus_reset to all active ports.

## Received Attach_request

When a node receives an attach_request token on any untested port, it should begin arbitrating for its local bus. When it wins arbitration, it will place all untested ports that on which attach_requests have been received in the active state. It will then generate send bus_reset to all its active ports.

If a node receives a bus_reset on a port from which it had received an attach_request, the port is placed in the active state and the reset is propagated to the active bus. If attach_request had been received on any other ports, those ports are not placed in the active state and the reset is not propagated on those ports.

## Maximum Occupancy Timer

The maximum occupancy timer runs as long as the node is holing the bus to run a test or complete a connection. It's duration is TBD us.

## Retest

A port that is in the loop_disabled state is placed in the untested state when a bus reset occurs.

## Test Value

The 6-bit test_value in the LTP is generated by one of two methods. If the PHY has access to a fixed value that is known to be unique among all possible PHY's (e.g., a EUI-64), that value may be used to provide the successive values for the test value. If the PHY has no access to such fixed value, then the 6-bit value may be provided by a random number generator. If the random number generator is a polynomial type, then the generator should contain at least 32 bits. The generator should run continuously while power is present on the PHY. Six bits should be selected from the generator whenever a new test value is needed.

## Test_value Holding Register Initialization (HR)

The test_value in the HR is initialized from the random number source when the first port on the PHY is placed in the untested state. This provides an initial value that can be sent in an LTP as soon as the port has been synchronized. The generation number value in the HR need not be set to any specific initial value.

## Holding the Bus

The controlling node only sends the LTP once, or not at all. It does not release the bus until the test is complete and/or a connection is made. During the intervals while no packet data is being sent, the node will hold the bus by sending data_null.

## Loss of Synchronization

When synchronization is lost on the test port, the port returns to the training state. If the node was holding the bus for a loop test, then the bus is released and another untested port, if any, is selected for test. If, however, an LTP with a mode of *attach in progress* had been sent on the active bus, then a bus reset must be sent to the active bus (note: this bus reset is not sent on the port that had lost synchronization since it is no longer the test port.)

If synchronization is lost on a connection that is active <<fill this in when we know what we are going to do under the normal circumstances.>>

## No Active Ports

If a PHY has no ports in the active state, then the test interval may be set to zero.

## Connections to 1394-1995 and 1394a Nodes

When an 1995 or 1394a node is attached to a border node, there is no exchange of LTP's. Instead, the 1995/a connection is activated according to 1995/a rules. If a configuration timeout occurs, all beta mode ports are placed in the untested state and a loop free build is done. The loop free build does not start until (TBD)us after the loop timeout.. A node attached to 1995/a nodes may not begin testing any of it's beta ports until tree ID completes on all 1995/a ports.

## Minimal LTP Support

A PHY with only a single beta-mode capable port is not required to support the full loop free build process. If a PHY has limited support for loop free build, it shall arbitrate for its bus and send a attach_request token on the beta mode port as soon as it is granted the bus. It should continue to send the attach_request tokens until it receives a bus reset or attach_request token from the other node. If an attach_request is received after the node has been granted its active bus, then it will send a bus reset to all ports.