Fast Ethernet Consortium

Clause 27 Repeater Conformance Test Suite v1.2 Report

UNH-IOL — 121 Technology Drive, Suite 2 — Durham, NH 03824 — +1-603-862-0090
Consortium Manager: Peter Scruton — pjs@iol.unh.edu — +1-603-862-0239

Joe Vendor
Company X
123 Main St.
Pleasantville, AK 55442

January 15, 2005
Report Rev. 1.0

Enclosed are the results from the Clause 27 Repeater Conformance testing performed on:

Device Under Test (DUT):	Company X Repeater 3
Hardware Version:	Rev. 44
Firmware Version:	N/A
Software Version:	N/A
Miscellaneous:	Port 5

The test suite referenced in this report is available at the UNH-IOL website:

ftp://ftp.iol.unh.edu/pub/ethernet/test_suites/CL27_REPEATER/Repeater_Test_Suite_v1.2.pdf

Issues Observed While Testing
27.2.4 – Cessation of Jam Propagation Delay – The DUT was observed to jam
27.5.1 – False Carrier Detect – The DUT was observed to detect carrier falsely

For specific details regarding issues please see the corresponding test result.

Testing Completed 01/15/2005

John Q. Tester
johnqtester@iol.unh.edu

Digitally signed by UNH-IOL
Date: 2005.01.28 13:19:39 -05'00'

Review Completed 01/15/2005

John Q. Reviewer
johnqreviewer@iol.unh.edu

Digital Signature Information

This document was created using an Adobe digital signature. A digital signature helps to ensure the authenticity of the document, but only in this digital format. For information on how to verify this document's integrity proceed to the following site:

http://www.iol.unh.edu/certifyDoc

If the document status still indicates "Validity of author NOT confirmed", then please contact the UNH-IOL to confirm the document's authenticity. To further validate the certificate integrity, Adobe 6.0 should report the following fingerprint information:

MD5 Fingerprint: DB27 087D 94C8 CB63 7679 50E1 2239 C564
SHA-1 Fingerprint: 5411 C271 9458 ECB2 F401 E0C9 0026 25C3 98D3 E8FE

Result Key

The following table contains possible results and their meanings:

Result	Interpretation
PASS	The Device Under Test (DUT) was observed to exhibit conformant behavior.
PASS with Comments	The DUT was observed to exhibit conformant behavior however an additional explanation of the situation is included, such as due to time limitations only a portion of the testing was performed.
FAIL	The DUT was observed to exhibit non-conformant behavior.
Warning	The DUT was observed to exhibit behavior that is not recommended.
Informative	Results are for informative purposes only and are not judged on a pass or fail basis.
Refer to Comments	From the observations, a valid pass or fail could not be determined. An additional explanation of the situation is included.
Not Applicable	The DUT does not support the technology required to perform these tests.
Not Available	Due to testing station or time limitations, the tests could not be performed.
Borderline	The observed values of the specified parameters are valid at one extreme, and invalid at the other.
Not Tested	Not tested due to the time constraints of the test period.

GROUP 1: Data Handling Functions

Test # and Label	Part(s)	Result(s)
27.1.1 – Data Frame Forwarding	**a**	**PASS**
	b	**PASS**

Expected Results and Procedural Comments

Purpose: To verify that the repeater set properly forwards data frames

Requirements: A 100Base-Tx stream is the 4B5B encoded MAC frame encapsulated with valid starting and ending delimiters (/J/K/ and /T/R/ respectively). A 100Base-Txrepeater shall accept all such streams and propagate them to all other ports.

Procedure: Command testing station 1 to send a properly encapsulated, 64-byte, valid MAC frame. Testing station 1 will monitor transmit activity from port A. Testing station 2 will monitor transmit activity from port B.

Observable results:
a. Verify that the MAC frame sent by testing station 1 is reproduced on the transmitter of port B
b. Verify that the transmitter of port A sources only /I/ code groups.

Comments on Test Results

None

Test # and Label	Part(s)	Result(s)
27.1.2 – Received Code Violation Handling	**a**	**PASS**

Expected Results and Procedural Comments

Purpose: To verify that the repeater set properly forwards invalid code groups.

Requirements: If a stream contains an invalid code group, that code group is either propagated as is or is replaced with the /H/ code group. If /H/ substitution is used, every subsequent code group in the incoming stream shall be propagated as /H/ (/T/R/ is still used to delimit the end of the stream).

Procedure: Command the testing station to send a valid preamble (defined as the physical layer stream /J/K/A/A/A/A/A/A/A/A/A/A/A/A/A/B) followed by 64 code group pairs. This stream will contain at least one invalid code group. The invalid code group will not be the last code group sent. The testing station will observe transmit activity from port B.

Observable results:
a. Verify that the repeater forwards the invalid code group and all subsequent code groups, unaltered, to all other attached ports, or…
b. Verify that the repeater substitutes the /H/ code group for the violation code group and all subsequent code groups.

Comments on Test Results

The DUT forwards the frame and replaces the invalid code with a /H/ code group and apparently replaces all subsequent code groups with /H/ as well.

Test # and Label	Part(s)	Result(s)
27.1.3 – Speed Handling	**a**	**PASS**

Expected Results and Procedural Comments

Purpose: To verify that the repeater blocks the flow of non-100Mbps signals.

Requirements: 10Mbps signals incident on any repeater port shall not affect any other repeater port.

Procedure: Command the testing station to apply a 10Mbps Manchester encoded, 64-byte, valid MAC frame to port A of the repeater under test. The testing station will monitor transmit activity on port B.

Observable results:
a. Verify that the repeater under test blocks the flow of the 10Mbps signal.

Comments on Test Results

The DUT was not observed to transmit non-100Mbps signaling to a port connected at 100Mbps.

GROUP 2: Propagation Delays

Test # and Label	Part(s)	Result(s)
27.2.1 – Start of Packet Propagation Delay	**a**	**PASS**
	b	**PASS**

Expected Results and Procedural Comments

Purpose: To measure the start of packet propagation delay for the repeater set.

Requirements: Let SOP(xy) be the start of packet propagation delay from port x to port y. Let SOJ(xy) be the collision-jam propagation delay from port x to port y. For a repeater to meet class I requirements, SOP(xy) + SOJ(xy) ≤ 140BT. 100Base-Tx and 100Base-Fx Class II repeaters must have SOP(xy) ≤ 46BT and SOJ(xy) ≤ 46BT. In either case, given any three ports A, B, and C, SOP(AC) < SOP(AB) + SOP(BC).

Procedure:

1. Connect the testing station and the device under test so that x=A and y=B. Command the testing station to send a properly encapsulated, 64-byte, valid MAC frame. Let $t=t_0$ represent the 50% point of the mid-cell transition corresponding to the leading bit of the /J/ code group arriving at port x. Let $t=t_1$ represent the 50% point of the mid-cell transition corresponding to the leading bit of the /J/ code group appearing on port y. The time difference $t_1 - t_0$ is SOP(AB).
2. Connect the testing station and the DUT so that x=B and y=C. Repeat step 1. This time the difference is SOP(BC).
3. Connect the testing station and the DUT so that x=A and y=C. Repeat step 1. This time the difference is SOP(AC).

Observable results:

a. Verify that SOP(xy), in conjunction with the results of 27.2.3, meets the constraints that SOP + SOJ ≤ 140BT. For Class II (all ports FX/TX) repeater units, verify that SOP ≤ 46BT.
b. Verify that SOP(AC) < SOP(AB) + SOP(BC)

Comments on Test Results

a. SOP was observed to be 42±2BT which meets the specified Class II limit of SOP ≤ 46BT.
b. The DUT meets the requirement SOP(AC) < SOP(AB) + SOP(BC).

Test # and Label	Part(s)	Result(s)
27.2.2 – Start of Packet Propagation Delay Variablility	**a**	**PASS**

Expected Results and Procedural Comments

Purpose: To measure the start of packet propagation delay variability for the repeater set.

Requirements: The Start of Packet propagation delay from port x to y (SOP(xy)) for successive packets shall not vary by more than 7BT.

Procedure: Command the testing station to send consecutive, properly encapsulated valid MAC frames spaced by 96BT for class I repeaters and 89BT for class II repeaters. The MAC frame length should alternate between 1,518 and 64 bytes, the first packet sent being 1,518 bytes long. The difference in SOP(AB) for successive packets is the start of packet propagation delay variability.

Observable results:
a. Verify that SOP(AB) does not vary by more than 7BT for successive packets.

Comments on Test Results

No variability was measured. This meets the requirement that the variance shall not exceed 7BT.

Test # and Label	Part(s)	Result(s)
27.2.3 – Collision-Jam Propagation Delay	**a**	**PASS**
	b	**PASS**

Expected Results and Procedural Comments

Purpose: To measure the collision-jam propagation delay for the repeater set.

Requirements: Let SOP(xy) be the start of packet propagation delay from port x to port y. Let SOJ(xy) be the collision-jam propagation delay from port x to port y. For a repeater to meet class I requirements, SOP(xy) + SOJ(xy) ≤ 140BT. Class II repeaters must have SOP(xy) ≤ 46BT and SOJ(xy) ≤ 46BT.

Procedure: Command testing station 1 to send a properly encapsulated, 64-byte, valid MAC frame. The loopback connection between ports B and C will cause the repeater under test to set command(ALL)=collision. Testing station 2 will monitor transmit activity from port B. Let $t = t_0$ represent the 50% point of the mid-cell transition corresponding to the leading bit of the /J/ code group arriving at port B (appearing on port C). Let $t = t_1$ represent the 50% point of the mid-cell transition corresponding to the first bit of Jam sourced by port B. The time difference $t_1 - t_0$ is SOJ(AB).

Observable results:
a. Verify that Jam message (JamX or any other well formed arbitrary data pattern) is sourced on all ports SOJ(AB) bit times after the appearance of the first bit of the /J/ code group on ports B and C.
b. Verify that SOJ(AB), in conjunction with the results from test #27.2.1, meets the constraint that SOP + SOJ ≤ 140BT. For class II (all ports FX/TX) repeater units verify that SOJ ≤ 46BT.

Comments on Test Results

SOJ was observed to be 32±4BT which does meet the Class II requirement SOJ ≤ 46BT. The DUT also meets the specified Class I limit of SOP + SOJ ≤ 140BT.

Test # and Label	Part(s)	Result(s)
27.2.4 – Cessation of Jam Propagation Delay	**a**	**PASS**
	b	**FAIL**

Expected Results and Procedural Comments

Purpose: To measure the cessation of collision jam propagation delay for the repeater set.

Requirements: Let EOJ(xy) be the cessation of jam propagation delay from port x to port y. SOJ(xy)−4BT ≤ EOJ(xy) ≤ SOJ(xy).

Procedure: Command the testing station to send a properly encapsulated, 64-byte, valid MAC frame. The loopback connection between ports B and C will cause the repeater under test to set command(ALL)=collision. Let L_1 be the number of bit times from the first bit of the /J/ code group to the first bit of idle pattern arriving at port A. Let L_2 be the number of bit times from the first bit of the /J/ code group to last bit of Jam appearing on port B. EOJ(AB) = SOP(AB) + L_2 - L_1.

Observable results:

a. Verify that the Jam message (JamX or any other well formed arbitrary data pattern) is sourced on all ports SOJ(AB) bit times after the appearance of the first bit of the /J/ code group on ports B and C.
b. Verify that SOP(AB) ≥ EOJ(AB) and that EOJ(AB) ≥ SOJ(AB) - 4BT.

Comments on Test Results

a. EOJ was calculated to be 36±4BT.
b. The DUT does not meet the requirement that EOJ(AB) ≥ SOJ(AB) – 4BT. The DUT also meets the requirement that SOP(AB) ≥ EOJ(AB).

GROUP 3: Receive Jabber Function

Test # and Label	Part(s)	Result(s)
27.3.1 – jabber_timer	**a**	**PASS**
	b	**PASS**

Expected Results and Procedural Comments

Purpose: To measure the value of jabber_timer for the repeater port.

Requirements: 40,000BT ≤ jabber_timer ≤ 75,000BT. Jabber_timer is started when a new stream arrives at port x. If the stream does not end before jabber_timer expires, jabber(x) is set to true.

Procedure: Command the testing station to send a valid preamble (defined as the physical layer stream /J/K/A/A/A/A/A/A/A/A/A/A/A/A/A/B/) followed by 10,000 code group pairs. Let $t = t_0$ represent the 50% point of the mid-cell transition corresponding to leading bit of the /K/ code group appearing on port A. Let $t = t_1$ represent the 50% point of the mid-cell transition corresponding to the last bit of data sourced by port B. The time difference $t_1 - t_0$ is jabber_timer

Observable results:
a. Verify that the output of the DUT transitions from repeated data to idle pattern (the end of shell delimiter is optional) once jabber_timer expires.
b. Verify that jabber_timer is in the range of 40,000BT to 75,000BT.

Comments on Test Results

a. The DUT was observed to transition from repeated data, to an ESD, then transitioned to idle once jabber_timer expired.
b. The jabber_timer was observed to be within the specified range of 40,000 to 75,000 BT.

Test # and Label	Part(s)	Result(s)
27.3.2 – Receive Jabber Function	**a**	**PASS**
	B	**PASS**

Expected Results and Procedural Comments

Purpose: To verify the conditions for a jabber(X)=false to jabber(X)=true transition, to observe the effects of jabber(X)=true, and to verify the conditions for a jabber(X)=false to jabber(X)=true transition.

Requirements: While jabber(x) = true, the stream arriving on port x will no longer propagate to other ports. Furthermore, streams arriving on other ports will no longer propagate to port x. Jabber(x) will be set to false when idle is detected on port x.

Procedure:

1. Command the testing station to send a valid preamble (defined as the physical layer stream /J/K/A/A/A/A/A/A/A/A/A/A/A/A/B/) followed by a continuous code group sequence. This will set and hold jabber(A)=true. Command testing station 2 to send a properly encapsulated, 64-byte, valid MAC frame. Testing station 1 will monitor transmit activity from port A.
2. Command testing station 1 to transition from continuous data generation to idle pattern generation. This will reset and hold jabber(A)=false. Command testing station 2 to send a properly encapsulated, 64-byte, valid MAC frame. Testing station 1 will monitor transmit activity from port A.

Observable results:

a. Verify that the packets are not forwarded to port A when link_status(A)=OK and jabber(A)=true.
b. Verify that packets are forwarded to port A when link_status(A)=OK and jabber(A)=false.

Comments on Test Results

None

GROUP 4: Partition Function

Test # and Label	Part(s)	Result(s)
27.4.1 – CCLimit	**a**	**PASS**
	b	**PASS**

Expected Results and Procedural Comments

Purpose: To measure the value of the CCLimit for the repeater set.

Requirements: CCLimit is an integer greater than 60.

Procedure:
1. Power-cycle reset the repeater under test.
2. Command the testing station to send a valid preamble (defined as the physical layer stream /J/K/A/A/A/A/A/A/A/A/A/A/A/A/B/) followed by four bytes of data and the end of shell delimiter (/T/R/). The loopback connection on port B will guarantee that this transmission will suffer a collision. The testing station will monitor activity from port A.
3. The repeater shall enforce the collision by sourcing the Jam message (JamX or any other well formed arbitrary data pattern) on all ports. Verify that the testing station has detected the Jam message.
4. Repeat steps 1 and 2 until the testing station no longer detects the Jam message. This is an indication that the carrier activity from port A is not reaching the repeater unit or that the carrier activity is not being propagated to port B. The former is a result of the partition condition being detected on port A and latter is a result of the partition condition being detected on port B (a conformant repeater will partition both ports). The number of collisions required to reach this state is CCLimit.

Observable results:
a. Verify that the partition condition is detected on port A and port B after a number of consecutive collisions equaling CCLimit.
b. Verify that CCLimit is greater than 60.

Comments on Test Results

Note: CCLimit was observed to be 61, which meets the specified limit CCLimit ≥ 60.

Test # and Label	Part(s)	Result(s)
27.4.2 – Effects of Partition	**a**	**PASS**
	b	**PASS**
	c	**PASS**

Expected Results and Procedural Comments

Purpose: To verify that the repeater port under test disables its receive path upon detection of the partition condition.

Requirements: Let x be a repeater port for which partition(x) = true. Streams arriving on other ports shall propagate to port x but streams arriving on port x will not propagate to other ports. Furthermore, simultaneous transmit and receive activity on port x shall not propagate through the repeater as a collision.

Procedure

1. Power-cycle the repeater under test.
2. Command testing station 1 to send a valid preamble (defined as the physical layer stream /J/K/A/A/A/A/A/A/A/A/A/A/A/A/A/B/) followed by four bytes of data and the end of shell delimiter (/T/R/). The loopback connection on port B will guarantee that this transmission will suffer a collision.
3. Repeat step 1 until CCLimit collisions have been generated (refer to test #27.4.1). Both port A and port B of the repeater under test should be partitioned.
4. Command testing station 1 to send two properly encapsulated, 64-byte, valid MAC frames separated by the minimum inter-frame gap (96BT). Testing station 2 will monitor transmit activity from port C.
5. Command testing station 2 to send a valid preamble followed by four bytes of data and the end of shell delimiter. Testing station 1 will monitor transmit activity from port A.

Observable results

a. Verify that the partition condition is detected on port A and port B after a number of consecutive collisions equaling CCLimit.
b. Verify that the packets generated by testing station 1 were not repeated to port C of the repeater under test.
c. Verify that the fragment generated by testing station 2 was repeated to port A of the repeater under test.

Comments on Test Results

None

Test # and Label	Part(s)	Result(s)
27.4.3 – no_collision_timer	**a**	**PASS**
	b	**PASS**
	c	**PASS**

Expected Results and Procedural Comments

Purpose: To measure the value of no_collision_timer for the repeater port.

Requirements: 450BT ≤ no_collision_timer ≤ 560BT

Procedure:
1. Command testing station 1 to send a valid preamble (defined as the physical layer stream /J/K/A/A/A/A/A/A/A/A/A/A/A/A/A/B/) followed by four bytes of data and the end of shell delimiter (/T/R/). The loopback connection on port B will guarantee that this transmission will suffer a collision.
2. Repeat step 1 until CCLimit collisions have been generated (refer to test #27.4.1). Both port A and port B of the repeater under test should be partitioned.
3. Command testing station 2 to send a valid preamble, eight nibbles of data, and end of shell delimiter. Command testing station 1 to send a properly encapsulated, 64-byte, valid MAC frame. Testing station 2 will monitor transmit activity from port C.
4. Testing station 2 will fail to capture the carrier event generated by testing station 1 as long as port A is partitioned. Repeat step 3, incrementing the number of nibbles sent by testing station 2 with each iteration, until testing station 2 captures the frame sent by testing station 1. The terminal length (in bit times) of the fragment sent by testing station 2 is no_collision timer.

Observable results:
a. Verify that the partition condition is detected on port A and port B after a number of consecutive collisions equaling CCLimit.
b. Verify that the port resets the partition function when there is transmit activity on the port for more than the number of bits specified by no_collision_timer.
c. Verify that no_collision_timer is in the range of 450 to 560 bit times.

Comments on Test Results

Note: no_collision_timer was observed to be 520±4BT, which is within the specified range of 450 to 560 BT.

Test # and Label	Part(s)	Result(s)
27.4.4 – Partition Function Reset	**a**	**PASS**
	b	**PASS**
	c	**PASS**
	d	**PASS**
	e	**PASS**

Expected Results and Procedural Comments

Purpose: To verify the conditions under which a repeater port will reset the partition function.

Requirements: Let x be a repeater port for which partition(x) = true. Partition(x) will transition to false on a power-cycle reset or if stream propagates through to port x for longer than no_collision_timer bits without suffering a collision. Partition(x) will not be set to false as a result of link_status(x) being set to FAIL. In addition 27.3.1.6 and figure 27-8 have been modified to include additional optional measures.

Procedure:

1. Power-cycle the repeater under test.
2. Command testing station 1 to send a valid preamble (defined as the physical layer stream /J/K/A/A/A/A/A/A/A/A/A/A/A/A/A/A/B/) followed by four bytes of data and the end of shell delimiter (/T/R/). The loopback connection on port B will guarantee that this transmission will suffer a collision.
3. Repeat step 1 until CCLimit collisions have been generated (refer to test #27.4.1). Both port A and port B of the repeater under test should be partitioned.
4. Disconnect testing station 1 from port A of the repeater under test. This will force link_status(A)≠OK. Reconnect testing station 1 to port A of the repeater under test. Verify that the partition function was not reset on port A.
5. If the partition function was reset as a result of step 3, repeat steps 1 and 2 to ensure that both port A and port B of the repeater under test are partitioned. Command testing station 2 to send a valid preamble followed by a sufficient umber of data nibbles to make the overall fragment length exceed no_collision_timer (refer to test #27.4.3). Verify that the partition function is reset on port A of the repeater under test.
6. Repeat steps 1 and 2 to ensure that both port A and port B of the repeater under test are partitioned. Power-cycle reset the repeater under test. Verify that the partition condition is reset on both port A and port B of the repeater under test.

Observable Results:

a. Verify that the partition condition is detected on port A and port B after a number of consecutive collisions equaling CCLimit.
b. Verify the transition to link_status(A)≠OK does not reset the partition function on port A of the repeater under test.
c. Verify that port A resets the partition function when there is transmit activity on the port for more than the number of bits specified by no_collision_timer.
d. Verify that a power-cycle reset will reset the partition function on both port A and port B of the repeater under test.
e. Verify that the DUT will not reset the partition function on a port for a transmitted event exceeding no_collision_timer if there is still receive activity at the port.

Comments on Test Results

None

Test # and Label	Part(s)	Result(s)
27.4.5 – Consecutive Collision Count	**a**	**PASS**
	b	**PASS**
	c	**PASS**

Expected Results and Procedural Comments

Purpose: To verify that the repeater port consecutive collision count increments and clears under the appropriate conditions.

Requirements: Let CC(x) be the collision counter for port x. CC(x) shall be incremented for each stream arriving on port x that suffers a collision before no_collision_timer expires. If no_collision_timer expires and the incoming stream has not suffered a collision, CC(x) is set to 0. Incoming streams shorter than no_collision_timer bits in length that do not suffer a collision shall cause no change in CC(x). If CC(x) reaches CCLimit, partition(x) shall be set to true.

Procedure:
1. Power-cycle the repeater under test.
2. Command the testing station to send a valid preamble (defined as the physcial layer stream /J/K/A/A/A/A/A/A/A/A/A/A/A/A/A/A/B/) followed by four bytes of data and the end of shell delimiter (/T/R/). The loopback connection on port B will guarantee that this transmission will suffer a collision. The testing station will monitor transmit activity on port A.
3. Repeat step 2 until CCLimit/2 (refer to test #27.4.1) collisions have been generated. Remove the loopback plug and command the testing station to send a valid preamble followed by a sufficient number of data nibbles to make the overall fragment length greater than no_collision_timer (refer to test #27.4.3).
4. Re-insert the loopback plug and repeat step 2 until the testing station fails to detect the Jam message from port A of the repeater under test. Verify that exactly CCLimit collisions were required to reach this state.
5. Power-cycle reset the repeater under test and repeat step 2 until CCLimit/2 collisions have been generated. Remove the loopback plug and command the testing station to send a valid preamble and any number of data nibbles that make the overall fragment length less than no_collision_timer.
6. Re-insert the loopback plug and repeat step 2 until the testing station fails to detect the Jam message from port A of the repeater under test. Verify that exactly CCLimit/2 collisions were required to reach this state.

Observable Results
a. Verify that the Partition condition is detected on port A and port B after a number of consecutive collisions equaling CCLimit.
b. Verify that the port clears the consecutive collision count when there is transmit activity on the port for more than the number of bits specified by no_collision_timer without incurring a collision.
c. Verify that the port does not increment or clear the consecutive collision count when there is transmit activity on the port for a number of bits less than no_collision timer without incurring a collision.

Comments on Test Results

None

GROUP 5: Carrier Integrity Monitor

Test # and Label	Part(s)	Result(s)
27.5.1 – False Carrier Detect	**a**	**FAIL**
	b	**FAIL**

Expected Results and Procedural Comments

Purpose: To verify that the repeater set can detect false carrier events

Requirements: A false carrier is defined to be a non-idle code group sequence that does not begin with a valid start of shell delimiter (SSD). A repeater shall detect a false carrier event occurring on one of its ports and propagate the event to all ports in the form of a valid SSD followed by jam. The repeater will continue to jam all ports for the duration of the false carrier event or until false_carrier_timer expires.

Procedure:

1. Let bad_ssd be a vector of 10 code-bits and let bad_ssd[0] be fixed at ZERO. Initialize bad_ssd[9:2] to the code-bit pattern "1111110". Command the testing station to send bad_ssd (most significant bit first) followed by four bytes of data. The testing station will monitor transmit activity from port B of the repeater under test.
2. Shift bad_ssd[9:2] left one code-bit, discarding the carry bit and setting bad_ssd[2] to ONE. Command the testing station to send bad_ssd followed by four bytes of data. The testing station will monitor transmit activity from port B of the repeater under test.
3. Repeat step 2 until bad_ssd[9:2] contains the pattern "11111111".
4. Set bad_ssd[9:5] to the /J/ code group and set bad_ssd[4:0] to the code-bit pattern "00000". Command the testing station to send bad_ssd followed by four bytes of data. The testing station will monitor transmit activity from port B of the repeater under test.
5. Increment bad_ssd[4:0]. Command the testing station to send bad_ssd followed by four bytes of data. The testing station will monitor transmit activity from port B of the repeater under test.
6. Repeat step 5 until bad_ssd[4:0] exceeds "11111". Skip the iteration in which bad_ssd[4:0] equals "10001" as this is the /K/ code-group (this makes bad_ssd[9:0] /J/K/, the valid start of shell delimiter).

Observable results:

a. Verify that, for each bad_ssd applied to port A, port B sources a valid start of shell delimiter followed by the Jam message (JamX or any other well formed arbitrary data pattern).
b. Verify that the length of the Jam message sourced by port B does not exceed the length of the false carrier event applied to port A by more than 4 BT.

Comments on Test Results

The DUT was observed to source invalid SSDs.

Test # and Label	Part(s)	Result(s)
27.5.2 – FCCLimit	**a**	**PASS**
	b	**PASS**

Expected Results and Procedural Comments

Purpose: To measure the value of FCCLimit for the repeater set.

Requirements: FCCLimit = 2

Procedure:
1. Power-cycle the repeater under test
2. Command the testing station to generate three consecutive false carrier events 40 BT in length and spaced by the minimum inter-frame gap (96 BT). A false carrier event is defined as a stream of non-idle code groups that does not begin with a valid start of shell delimiter (/J/K/). The testing station will monitor transmit activity from port B.

Observable results:
a. Verify that, after FCCLimit consecutive false carrier events, the link unstable condition is detected on port A of the repeater under test. The port will disable its transmit and receive paths while in the link unstable state. This will prevent any remaining false carrier events from propagating to port B of the repeater under test.
b. Verify that FCCLimit is 2.

Comments on Test Results

None

Test # and Label	Part(s)	Result(s)
27.5.3 – false_carrier_timer	**a**	**PASS**
	b	**PASS**

Expected Results and Procedural Comments

Purpose: To measure the value of false_carrier_timer for the repeater set.

Requirements: 450BT ≤ false_carrier_timer ≤ 500BT. False_carrier_timer is started when a non-idle code group sequence that does not start with a valid SSD arrives at port x. If sequence has not ended when false_carrier_timer expires, isolate(x) is set to true.

Procedure: Command the testing station to send an invalid start of shell delimter followed by 100 code group pairs. Let $t = t_0$ represent the 50% point of the mid-cell transition corresponding to the first bit of data (following the invalid start of shell delimiter) arriving on port A. Let $t = t_1$ represent the 50% point of the mid-cell transition corresponding to the last bit of Jam sourced by port B. The time difference $t_1 - t_0$ is false_carrier_timer.

Observable results:
a. Verify that a false carrier event with duration exceeding false_carrier_timer will cause the link unstable condition to be detected on port A of the repeater under test. The port will disable its transmit and receive paths while in the link unstable state. This will terminate the propagation of the false carrier event to port B of the repeater under test.
b. Verify that false_carrier_timer is in the range of 450 to 500 BT.

Comments on Test Results

Note: false_carrier_timer was observed to be 468±4BT which is within the specified range of 450 to 500 BT.

Test # and Label	Part(s)	Result(s)
27.5.4 – ipg_timer	**a**	**PASS**
	b	**PASS**

Expected Results and Procedural Comments

Purpose: To measure the value of ipg_timer for the repeater set.

Requirements: 64BT ≤ ipg_timer ≤ 86BT. Given that isolate(x) = true, if ipg_timer expires and there is no carrier arriving on port x for the duration of idle_timer, isolate(x) is set to false. If ipg_timer expires and a stream arrives on port x exceeding valid_carrier_timer BT in length, isolate(x) is set to false. Note that the above stream will not be propagated to other ports.

Procedure:
1. Power-cycle the repeater under test.
2. Let S be a sequence of carrier events. The sequence begins with FCCLimit (refer to test 27.5.2) false carrier events 40BT long and seperated by a 96BT inter-packet gap. Two properly encapsulated, 64-byte, valid MAC frames seperated by 96BT follow the false carrier events after an *n*BT delay. Set *n* to 48 and command the testing station to send S. The testing station will monitor transmit activity on port
3. A valid carrier event of duration greater than valid_carrier_timer preceeded by idle of duration greater than ipg_timer will cause the repeater port under test to reset the link unstable state. This will allow any subsequent carrier events received on port A to be propagated to the other ports of the repeater under test. Increment *n* by 4 and command the testing station to send S. The testing station will monitor transmit activity on port B.
4. Repeat step 3 until the second valid carrier event is detected by the station under test. The terminal value of *n* is ipg_timer.

Observable results:
a. Verify that a valid carrier event of duration greater than valid_carrier_timer preceeded by idle of duration greater than ipg_timer will cause the repeater port under test to reset the link unstable state.
b. Verify that ipg_timer is in the range of 64 to 86 BT.

Comments on Test Results

Note: The value of ipg_timer was observed to be 76±4BT, which is not within the specified range of 64 to 86 BT.

Test # and Label	Part(s)	Result(s)
27.5.5 – valid_carrier_timer	**a**	**PASS**
	b	**PASS**

Expected Results and Procedural Comments

Purpose: To measure the value of valid_carrier_timer for the repeater set.

Requirements: 450BT ≤ valid_carrier_timer ≤ 500BT

Procedure:

1. Power-cycle the repeater under test.
2. Let S be a sequence of carrier events. The sequence begins with FCCLimit (refer to test 27.5.2) false carrier events 40BT long and seperated by a 96BT inter-packet gap. A fragment consisting of a valid preamble (defined as the physical layer stream /J/K/A/A/A/A/A/A/A/A/A/A/A/A/A/B/) and *n* bits of data follows after a delay of ipg_timer bit times (refer to test 27.5.4). The sequence ends with a properly encapsulated, 64-byte, valid MAC frame that follows the fragment after 96BT. Set *n* to 32 and command the testing station to send S. The testing station will monitor transmit activity on port B.
3. A valid carrier event of duration greater than valid_carrier_timer preceeded by idle of duration greater than ipg_timer will cause the repeater port under test to reset the link unstable state. This will allow any subsequent carrier events received on port A to be propagated to the other ports of the repeater under test. Increment *n* by 4 and command the testing station to send S. The testing station will monitor transmit activity on port B.
4. Repeat step 3 until the second valid carrier event is detected by the station under test. The terminal value of the first valid carrier event is valid_carrier_timer.

Observable results:

a. Verify that a valid carrier event of duration greater than valid_carrier_timer preceeded by idle of duration greater than ipg_timer will cause the repeater port under test to reset the link unstable state.
b. Verify that valid_carrier_timer is in the range of 450 to 500 BT.

Comments on Test Results

Note: The value of valid_carrier_timer was found to be 480±4BT which is not within the standard specified range of 450 to 500BT.

Test # and Label	Part(s)	Result(s)
27.5.6 – idle_timer	**a**	**PASS**
	b	**PASS**

Expected Results and Procedural Comments

Purpose: To measure the value of idle_timer for the repeater set.

Requirements: 24,750BT ≤ idle_timer ≤ 41,250BT

Procedure:
1. Power-cycle the repeater under test.
2. Let S be a sequence of carrier events. The sequence begins with FCCLimit consecutive false carrier events 40BT long and spaced by a 96BT inter-packet gap. A properly encapsulated, 64-byte, valid MAC frame follows after an *n*BT delay. Set n to ipg_timer (refer to test 27.5.4) and command the testing station to send S.
3. When the false carrier count for port A exceeds FCCLimit, the port will enter the link unstable state and disable its transmit and receive paths. The link unstable state will be reset if no activity is detected on port A for more than ipg_timer plus idle_timer bit times. Repeat step 1, incrementing *n* by 4 with each interation, until the testing station detects the valid MAC frame applied to port A. The terminal value of *n* is ipg_timer plus idle_timer.

Observable results:
a. Verify that the link unstable state is reset when no activity is detected on a port for more than ipg_timer plus idle_timer bit times.
b. Verify that idle_timer is in the range of 24,750 BT to 41,250 BT.

Comments on Test Results

Note: The value of idle_timer was observed to be within the specified range of 24,750 to 41,250 BT.

Test # and Label	Part(s)	Result(s)
27.5.7 – False Carrier Count	**a**	**PASS**
	b	**PASS**

Expected Results and Procedural Comments

Purpose: To verify consecutive false carrier count for the repeater port increments and clears under the appropriate conditions.

Requirements: Let FCC(x) be the false carrier counter for port x. FCC(x) is incremented for each non-idle code group sequence arriving at port x that does not begin with a valid SSD. FCC(x) is set to 0 if port x receives a non-idle code group sequence that begins with a valid SSD. If FCC(x) reaches FCCLimit, isolate(x) shall be set to true. Note that while in isolate(x) = true, streams arriving on port x are not propagated to other ports and streams arriving on other ports are not propagated to port x.

Procedure:
1. Power-cycle the repeater under test.
2. Command the testing station to send a false carrier event 40BT long (a false carrier event is defined to be a non-idle data stream that does not begin with a valid start of shell delimiter).
3. Command the testing station to send a properly encapsulated, 64-byte, valid MAC frame.
4. Command the testing station to send FCCLimit+1 (refer to test 27.5.2) false carrier events 40BT long and seperated by a 96BT inter-packet gap. The testing station will monitor transmit activity from port B. It should detect FCCLimit Jam messages from port B indicating that the valid MAC frame send in step 3 cleared the false carrier count.

Observable results:
a. Verify that the consecutive false carrier count is reset upon the reception of a valid carrier event.
b. Verify that, after FCCLimit consecutive false carrier events, the link unstable condition is detected on port A of the repeater under test. The port will disable it's transmit and receive paths while in the link unstable state. This will prevent any remaining false carrier events from propagating to port B of the repeater under test.

Comments on Test Results

None

Annex A: Test Setup

Test Equipment

The following test equipment was used in performing Clause 27 Repeater testing:

Testing Equipment	Brand and Version Information
PC Requirements	Windows 2000
Software	Smartwindows
Traffic Generator/Sniffer	Smartbits 2000
Logic Analyzer	N/A
Tiger/GMII System	N/A

Test Configuration

The following configuration was used in performing Clause 27 Repeater testing: