Implementation Receiver Interoperability Testing

#	Test Name	IP Mode	Sender	Receiver	Setup	Shows	Expected	Result, Green Indicates Passed, Red Indicates Failed. Each individual file contains further annotation / explaination specific to that test.
Pre-Test Setup Verification Testing - Ensures the setup is as described in the test setup diagram.								
0.1	Traceroute	IPv4 Unicast	OS traceroute utility	-	The traceroute command is run on the sender show the route of packets from the sender to receiver.	That the test setup is as described in the setup diagram.	The traceroute should show only a single hop from the sender to receiver machine with no routers present.	1 hop
0.2	Ping	IPv4 Unicast	OS ping utility	-	The ping command is run on the sender to the receiver to show the status of the network in terms of delay and loss.	That the network is sending packets correctly between the machines with delays below the timeout value specified in ANSI E1.31-2018 (E131_NETWORK_DATA_LOSS_TIMEOUT = 2.5 seconds) and no loss.	The ping should show no data loss as these tests assume a perfect network, the delay should be below the timeout value used within the protocol.	< 1ms
0.3	Check Addresses	IPv4	ipconfig	ifconfig	The sender and receiver test machines run a command to display their assigned addresses.	That the sender and receiver ip addresses are setup as shown on the test setup diagram.	The sender should have an IPv4 192.168.0.2 address, the receiver should have the IPv4 192.168.0.6 address. Both should have a subnet mask of 255.255.255.0	Sender: 192.168.0.2 Receiver: 192.168.0.6
0.4	Setup Control	IPv4 Multicast	Avolites Titan	sACN viewer	Avolites titan sends a full universe of DMX zero-startcode data on universe 1 (all values 255), sACN viewer receives and displays the data.	The industry source sending a full universe of data and an external receiver showing the received data. This acts as a control to show the expected output as the implementation receiver isn't involved. It also verifies that the setup itself works so any problems are with the receiver.	The sACN viewer should display the full universe of data from the source with all values listed correctly.	"test-0-4-receiver.png" - Shows receiver is receiving the data "test-0-4-sender.png" - Shows sender is sending the data
sACN Receiver Implementation Interoperability Testing								
1	Simple Data Send	IPv4 Multicast	Avolites Titan	Implementation	Avolites titan sends a full universe of DMX zero-startcode data on universe 1, Implementation receives the data and writes it to a file.	That the implementation receiver can receive data from an industry sender.	The output from the receiver should show the full universe of data from the sender with the correct payload (0 startcode + all 255) and universe number.	"test-1-out.csv" - Shows the received data which is all 255 as expected. "test1-receiver-cmd.png" - Shows the commands run on the receiver side "test-0-4-sender.png" - Shows the sender side (same as test 0-4).
2	Two Universes	IPv4 Multicast	Avolites Titan	Implementation	Avolites titan sends 2 distinct universes of DMX zero-startcode data on universes 1 (all 255) and 2 (all 128), the implementation receiver receives the data continously and writes it to a file.	That the implementation receiver can handle multiple universes of data from an industry sender.	The output file should contain the data from both universes with the correct values for each universe.	"test-2-out.csv" - Shows the received data which is all 255 for universe 1 and 128 for universe 2 as expected. "test-2-receiver-cmd" - Shows the commands run on the receiver side. "test-2-sender.png" - Shows the setup on the sender side.
3	Independent moving channels	IPv4 Multicast	Avolites Titan	Implementation	Avolites titan sends a universe of data with a value bump moving through the channels of the universe with each channel having a small offset from the previously, this means all the universe channels should continously change seperately.	That the implementation receiver can handle the channels within a universe changing repeatedly and with different values for each channel	The output file should show the channels of the universe moving up and down in a sine wave with each channel being offset from the previous.	"test-3-out.csv" - Shows the received data "test-3-data-processed.xlsx" - Excel spreadsheet used to process data into graph as visualisation. "test-3-processed-values-chart.png" - Shows the values of the first 5 channels of the universe (excluding startcode) which shows that the data is received in the form expected. "test-3-receiver-cmd.png" - Shows the receiver commands used for the test "Test-3-Sender-DMX-Annotated.mkv" - Shows the sender running with the values produced shown in a live table.
4	Rapid Changes	IPv4 Multicast	Avolites Titan	Implementation	Avolites titan rapidly increases and decreases the channels of the universe as a 'pulse' (0 value jumping to 255 full value briefly and then dropping back down waiting a short time and repeating)	That the implementation receiver can handle rapid short lived channel changes.	The output file should show the pulses in the universe channels.	"test-4-out.csv" - Shows the received data "test-4-data-processed.xlsx" - Excel spreadsheet used to process data into graph as visualisation. "test-4-processed-first-value-chart.png" - Shows the values of the first channel of the universe (excluding startcode) which shows that the data is received in the form expected. "test-4-receiver-cmd.png" - Shows the receiver commands used for the test "Test-4-Sender-Annotated.mkv" - Shows the sender running with the values produced shown in a live table.
5	High data rate	IPv4 Multicast	Avolites Titan	Implementation	Avolites titan sends data on the maximum number of universes it supports concurrently (16), the channels of each universe vary up and down between specified ranges *.	That the implementation receiver can handle the size of continously changing data stream possible from a single industry source. The distinct ranges for each universe show that the receiver is correctly not mixing up universes.	The output file should show the values from all 16 universes with all the values from a specific universe within the predefined ranges.	"test-5-out.csv" - Shows the received data "test-5-data-processed.xlsx" - Excel spreadsheet used to process data into graph as visualisation. "test-5-processed-first-value-chart.png" - Shows the values of the first channel of the universe (excluding startcode) which shows that the data is received in the form expected. "Test-5-Sender-Annotated.mkv" - Shows the sender running with the values produced shown in a live table. "Test-5-Sender-DMX-Setup.png" - Shows the DMX setup of the sender Test-5-Receiver-Control-sACNView-Annotated.mkv" - Shows the industry sACN viewer running to show the output that is expected. This is then compared to the actual produced graph to check they are similar as expected.
Tests which could not be performed as unsupported by industry implementation. Expected to pass as previous intergration tests show this functionality works.								
A	IPv6 Multicast Two Universes	IPv6 Multicast	Avolites Titan	Implementation	Follows the same format as test 2 however over IPv6 multicast rather than Ipv4.	That the implementation receiver can interoperate and receive multiple universes of data from an industry source utilising IPv6 multicast	Same results as test 2 as the ip-versions should be treated the same with regards to the data sent by the protocol as per ANSI E1.31-2018 Section 9.1	N/A, No industry source supporting IPv6 found/avaliable.
B	IPv6 Unicast Two Universes	IPv6 Unicast	Avolites Titan	Implementation	Follows the same format as test 3 however over IPv6 unicast rather than Ipv4.	That the implementation receiver can interoperate and receive multiple universes of data from an industry source utilising IPv6 unicast	Same results as test 3 as the ip-versions should be treated the same with regards to the data sent by the protocol as per ANSI E1.31-2018 Section 9.1	
C	Universe Syncronisation Hold	IPv4 Multicast	Avolites Titan	Implementation	Avolites titan sends a data packet with no synchronisation and then sends a data packet with its synchronisation address set to universe 2. It then waits a predefined period before sending the synchronisation packet to trigger the previously sent data packet.	That the implementation receiver correctly holds data awaiting syncronisation when sent by an industry source.	The receiver should receive and output the first not synchronised packet and then after the predefined period it should output the second data packet.	N/A, No industry source supporting universe synchronisation found/avaliable.
D	Universe Synchronisation Two Universes	IPv4 Multicast	Avolites Titan	Implementation	Avolites titan sends 2 data packets with different universes but the same synchronisation address. It then waits before sending a synchronisation packet to trigger the previously sent data packets.	That the implementation receiver correctly synchronises data from 2 distinct universes by correctly waiting for a synchronisation packet.	The receiver should receive and output both universes of data together rather than as seperate bits of data.	
E	Universe Discovery No Universes	IPv4 Multicast	Avolites Titan	Implementation	Avolites titan starts up and all universes are de-registered. The implementation receiver then periodically checks the discovered sources.	That the implementation receiver can receive and process universe discovery packets with no universes listed sent by an industry source.	The implementation receiver should (within 20 seconds**) show the industry source in its discovered sources list with an empty list of universes.	N/A, No industry source supporting universe discovery found/avaliable.
F	Universe Discovery Multiple Universes	IPv4 Multicast	Avolites Titan	Implementation	Avolites titan starts up and registers universe 1, 2 and 3. The implementation receiver then periodically checks the discovered sources.	That the implementation receiver can receive and process universe discovery packets with multiple universes listed sent by an industry source.	The implementation receiver should (within 20 seconds**) show the industry source in its discovered sources list with the list of universes containing universes 1, 2 and 3 as registered on the source.	
G	Unicast Two Universes	IPv4 Unicast	Avolites Titan	Implementation	Same as test 22 except using unicast for data sending.	That the implementation receiver can work over unicast aswell as multicast.	Same as test 2	N/A, Avolites Titan does not support sACN unicast

Key:
OS traceroute utility: The traceroute utility that exists on the sending test machine windows 10 operating system. Run using the command "".
OS ping utility: The ping utility that exists on the sending test machine windows 10 operating system. Run using the command "".
ipconfig: Displays the network configuration of the interfaces on a windows machine, shows the IP that each interface is bound to. Run using the command "ipconfig".
ifconfig: Similar functionality to ipconfig but for the Fedora/Linux operating system. Run using the command "ifconfig".
Avolites Titan: The industry source implementation used for interoperability testing, further details described in the tools section of the report. The showfile used for each test is included in the "Avolites Titan Test Showfiles" folder.
sACN viewer: The industry receiver implementation used as a control in the interoperability testing to make sure the setup works correctly.
Implementation: The 'demo_rcv' implementation of the library written as part of this project. This is the focus of the tests.

Note that the # column refers to the test number which corresponds to the folder number used for that test within the 'Test Results' folder.

* For each universe x, x in [1, 16], the range for that universes channels is [(x - 1) * 10, x * 10) so for example for universe 7 the expected range is: [(7-1) * 10, 7 * 10) = [6 * 10, 7 * 10) = [60, 70) so values are expected to be from 60 inclusive to 70 exclusive.
** A universe discovery packet should be sent at least every ANSI E1.31-2018 Appendix A E131_E131_UNIVERSE_DISCOVERY_INTERVAL of 10 seconds and so by waiting 20 seconds it means the source should have sent a packet and it been processed and added to the discovered sources by the receive even accounting for processing delays.

Implementation Sender Interoperability Testing

#	Test Name	IP Mode	Sender	Receiver	Setup	Shows	Expected	Result, Green Indicates Passed, Red Indicates Failed. Each individual file contains further annotation / explaination specific to that test.
Pre-Test Setup Verification Testing - Ensures the setup is as described in the test setup diagram.								
0.1	Traceroute	IPv4 Unicast	OS traceroute utility	-	The traceroute command is run on the sender show the route of packets from the sender to receiver.	That the test setup is as described in the setup diagram.	The traceroute should show only a single hop from the sender to receiver machine with no routers present.	Shown with Test Setup Folder, "ipconfig-receiver.png", "ipconfig-sender.png", "test-3-4-5-6-7-8-9-ipconfig-receiver.png", "test-3-4-5-6-7-8-9-ipconfig-tracert-ping-sender.png", "traceroute-ping.png". Each file is annotated with explainations.
0.2	Ping	IPv4 Unicast	OS ping utility	-	The ping command is run on the sender to the receiver to show the status of the network in terms of delay and loss.	That the network is sending packets correctly between the machines with delays below the timeout value specified in ANSI E1.31-2018 (E131_NETWORK_DATA_LOSS_TIMEOUT = 2.5 seconds) and no loss.	The ping should show no data loss as these tests assume a perfect network, the delay should be below the timeout value used within the protocol.	
0.3	Check Addresses	IPv4	ipconfig	ipconfig	The sender and receiver test machines run a command to display their assigned addresses.	That the sender and receiver ip addresses are setup as shown on the test setup diagram.	The sender should have an IPv4 192.168.0.2 address, the receiver should have the IPv4 192.168.0.6 address. Both should have a subnet mask of 255.255.255.0	
0.4	Setup Control sACN viewer	IPv4 Multicast	Avolites Titan	sACN viewer	Avolites titan sends a full universe of DMX zero-startcode data on universe 1 (values increasing from 0 to full to 0), sACN viewer receives and displays the data.	The industry source sending a full universe of data and an external receiver showing the received data. This acts as a control to show the expected output as the implementation receiver isn't involved. It also verifies that the setup itself works so any problems are with the receiver.	The sACN viewer should display the full universe of data from the source with all values listed correctly.	Shown within Test Setup Folder, "test-0-4-annotated.mkv"
0.5	Setup Control Vision Visualiser	IPv4 Multicast	Avolites Titan	Vision Visualiser	Avolites titan sends a full universe of DMX zero-startcode data on universe 1 (values increasing from 0 to full to 0), vision visualiser receives the data and visualises it.	The industry source sending a full universe of data and an external receiver showing the received data. This acts as a control to show the expected output as the implementation receiver isn't involved. It also verifies that the setup itself works so any problems are with the receiver.	The sACN viewer should display the full universe of data from the source with all values listed correctly.	Shown within Test Setup Folder, "test-0-5-annotated.mkv"
sACN Sender Implementation Interoperability Testing								
1	Simple Data Send	IPv4 Multicast	Implementation	Vision Visualiser	The implementation sender sends a single universe (universe 1) of static data (with a zero startcode and first channels set to 255, rest 0) to the vision visualiser	That the implementation sender can send data to the visualiser industry receiver.	The first three lights on sACN universe 1 should come on at full and stay there. The rest should stay off.	Test-1-Annotated.mkv
2	Simple Data Send	IPv4 Multicast	Implementation	sACN viewer	Same as test 1 but with the sACN receiver. Conducted in parallel with test 1 to show receivers side by side.	That the implementation sender can send data to the debug external receiver.	The data readout for sACN universe 1 should show the first three channels at full and the rest zero.	
3	Two Universes	IPv4 Multicast	Implementation	sACN viewer	The implementation sender sends 2 distinct universe of data (with zero startcode) to universe 1 (all channels 100) and 2 (all channels 255). sACN viewer displays the data output.	That the implementation sender can send data on multiple universes to an industry receiver.	The data readout for sACN universes 1 and 2 should show all channels at 100 for universe 1 and all channels at 255 for universe 2.	Test-3-Annotated.mkv
4	Unicast	IPv4 Unicast	Implementation	sACN viewer	Same as test 3 but using Unicast	That the implementation sender can send data on multiple universes using unicast to an industry receiver.	Same as test 3	Test-4-Annotated.mkv
5	Broadcast	IPv4 Broadcast	Implementation	Vision Visualiser	Same as test 3 but using Broadcast	That the implementation sender can send data on multiple universes using broadcast to an industry receiver.	Same as test 3	Test-5-Annotated.mkv
6	Preview Data	IPv4 Multicast	Implementation	sACN Viewer	Same as test 3 but showing preview data.	That the preview data flag is correctly applied and the data is treated the same otherwise.	The packets should have the preview_data flag set.	Test-6-Annotated.mkv
7	Independent moving channels	IPv4 Multicast	Implementation	Vision Visualiser	The implementation sender starts a predefined sequence where every channel in universe 1 follows a sine wave pattern with each channel offset slightly from the previous. This creates an effect which is commonly used in entertainment lighting refered to as a 'dim chase'.	That the implementation sender can handle sending data where each channel is continously updating.	The visualiser should show a sine wave pattern moving throughout the fixtures.	Test-7-Annotated.mkv
8	Rapid Changes	IPv4 Multicast	Implementation	Vision Visualiser	The implementation sender starts a predefined sequence where every channel is brought upto 100 briefly and then dropped back down to 0 repeatdly to cause a strobing effect.	That the implementation sender can handle sending data which is rapidly changing.	The visualiser should show all the lighting fixtures turning on and off quickly.	Test-8-Annotated.mkv
9	High data rate	IPv4 Multicast	Implementation	sACN viewer	The implementation sender sends a predefined sequence on 16 universes (chosen to match the limit on the industry sender Avolites Titan) where each universe follows a wave but with a slight offset for each universe.	That the implementation sender can handle sending on the same number of universes simultaneously as a source actually used within real-world industry.	sACN viewer to display all 16 universes working at once with independent distinct values for each. Vision visualiser could not be used for this as the version avaliable only supports 4 universes.	The expected wave is visualised using wolfram alpha in "wave-expected.png"
10	Universe Discovery	IPv4 Multicast	Implementation	sACN viewer	sACN viewer starts with no universes discovered. Then the implementation sender starts up with no universes registered. The sender then registers universe 1, 2, 400 and 5000 with gaps between each to show them appearing in the discovered universe list. Universe 400 is then terminated. The entire sender is then terminated.	That the implementation sender sends universe discovery packets correctly even with no universes registered so that they can be received and used by an industry receiver.	The sACN viewer should initially display no universes as the source is transmitting no universes. The universes should then appear as they are registered. Universe 400 should then be removed from the list on sACN viewer when it is terminated by the source. All the universes/source should then be removed when they are all terminated.	Test-10-Annotated.mkv
11	Stream Termination	IPv4 Multicast	Implementation	sACN Viewer	The implementation sender starts up and sends some arbitary data to the receiver on universe 1 to verify they are connected. The sender then shuts down and in doing so sends stream termination packets to the receiver on universe 1.	That the implementation sender sends stream termination packets correctly so that they are interoperable/compatiable with the industry receiver.	Sacn view should reset the channels within universe 1 as the universe has been terminated.	Test-1-Annotated.mkv
Tests which could not be performed as unsupported by industry implementation								
A	Universe Syncronisation Hold	IPv4 Multicast	Implementation	Vision Visualiser	The implementation sender sends a data packet synchronised to universe 2 with all lighting levels at full. The sender then sends a synchronisation packet to universe 2.	That the implementation sender correctly sends universe synchronisation packets which are compatiable with the industry receiver.	The visualiser doesn't show any change in output until the sychronisation packet is sent.	Not possible to test using the industry receiver as universe sychronisation is unsupported. Test is expected to pass based on previous intergration tests which show this functionality works.
B	Universe Synchronisation Two Universes	IPv4 Multicast	Implementation	Vision Visualiser	The implementation sender sends data packets with sychronisation address 1 to two universes with the first universe of data bringing one set of lights to full and the second universe of data bringing another set of lights to half. After short period a synchronisation packet is then send with synchronisation address 1.	That the implementation sender correctly sends universe synchronisation packets which are compatiable with the industry receiver.	The first two data packets produce no change to the visualiser input initially as they are awaiting the sychronisation packet. Once the synchronisation packet is sent the data packets then both take effect at the same time causing one set of lights to turn on at full brightness and the other set to turn on at half brightness.	
C	Traceroute IPv6	IPv6 Unicast	OS traceroute utility	-				Failed, IPv6 not supported by either receiver. Traceroute, Ping and addresses were set correctly as shown in Test Ipv6 folder.
	Ping IPv6	IPv6 Unicast	OS ping utility	-				
	Check Addr IPv6	IPv6	ipconfig	ipconfig				
	Setup Control sACN Viewer IPv6	IPv6 Multicast	Avolites Titan	Vision Visualiser	Same setup as IPv4 version (Test 0.4 / 0.5) but using IPv6.			
D	IPv6 Multicast Two Universes	IPv6 Multicast	Implementation	sACN viewer	Same as test 3 but using IPv6 Multicast.	That the implementation sender can send data on multiple universes using IPv6 multicast to an industry receiver. Test 5 + 6 shows the library works over IPv6.	Same as test 3	Failed, IPv6 not supported by either receiver
E	IPv6 Unicast Two Universes	IPv6 Unicast	Implementation	sACN viewer	Same as test 3 but using IPv6 Unicast.	That the implementation sender can send data on multiple universes using IPv6 unicast to an industry receiver. Test 5 + 6 shows the library works over IPv6.	Same as test 3	Failed, IPv6 not supported by either receiver

Key:
OS traceroute utility: The traceroute utility that exists on the sending test machine windows 10 operating system. Run using the command "".
OS ping utility: The ping utility that exists on the sending test machine windows 10 operating system. Run using the command "" .
ipconfig: Displays the network configuration of the interfaces on a windows machine, shows the IP that each interface is bound to. Run using the command "ipconfig".
Avolites Titan: The industry source implementation used for interoperability testing, further details described in the tools section of the report. The showfile used for each test is included in the "Avolites Titan Test Showfiles" folder.
sACN viewer: An industry receiver implementation in the form of a debug tool which receives sACN input and displays information about the received input. Further details in the tools section of the report.
Vision Visualiser: An industry receiver implementation in the form of a visualiser which receives sACN input and simulates lighting output. Further details in the tools section of the report.
Implementation: The 'demo_src' implementation of the sending side of the library written as part of this project. This is the focus of the tests.

Note that the # column refers to the test number which corresponds to the folder number used for that test within the 'Test Results' folder.

During these tests the visualiser uses a predefined lighting layout that allows easily seeing all the channels of a universe. This layout is described in more detail in the report.

The choice between sACN and Visualiser to use as the receiver for each test was decided based on which would show the output of the test in the clearest way. This means for checking strict data values sACN is preferred however for checking showing lots of channels changing overtime a visualiser is more effective.

* For each universe x, x in [1, 16], the range for that universes channels is [(x - 1) * 10, x * 10) so for example for universe 7 the expected range is: [(7-1) * 10, 7 * 10) = [6 * 10, 7 * 10) = [60, 70) so values are expected to be from 60 inclusive to 70 exclusive.
** A universe discovery packet should be sent at least every ANSI E1.31-2018 Appendix A E131_E131_UNIVERSE_DISCOVERY_INTERVAL of 10 seconds and so by waiting 20 seconds it means the source should have sent a packet and it been processed and added to the discovered sources by the receive even accounting for processing delays.

Acceptance Testing

#	Test Name	IP Mode	Sender	Receiver	Setup	Shows	Expected	Result, Green Indicates Passed, Red Indicates Failed. Each individual file contains further annotation / explaination specific to that test.
Pre-Test Setup Verification Testing - Ensures the setup is as described in the test setup diagram.								
0.1	Traceroute	IPv4 Unicast	OS traceroute utility	-	The traceroute command is run on the sender show the route of packets from the sender to receiver.	That the test setup is as described in the setup diagram.	The traceroute should show only a single hop from the sender to receiver machine with no routers present.	Test could not be setup and completed due to COVID-19 Shutdown. Expected to have passed based on the interoperability testing which uses a similar setup
0.2	Ping	IPv4 Unicast	OS ping utility	-	The ping command is run on the sender to the receiver to show the status of the network in terms of delay and loss.	That the network is sending packets correctly between the machines with delays below the timeout value specified in ANSI E1.31-2018 (E131_NETWORK_DATA_LOSS_TIMEOUT = 2.5 seconds) and no loss.	The ping should show no data loss as these tests assume a perfect network, the delay should be below the timeout value used within the protocol.	
0.3	Check Addresses	IPv4	ipconfig	ipconfig	The sender and receiver test machines run a command to display their assigned addresses.	That the sender and receiver ip addresses are setup as shown on the test setup diagram.	The sender should have an IPv4 192.168.0.2 address, the receiver should have the IPv4 192.168.0.6 address. Both should have a subnet mask of 255.255.255.0	
0.4	Setup Control Vision Visualiser	IPv4 Multicast	Avolites Titan	Vision Visualiser	Avolites titan sends a full universe of DMX zero-startcode data on universe 1 (all values 255), vision visualiser receives the data and visualises it.	The industry source sending a full universe of data and an external receiver showing the received data. This acts as a control to show the expected output as the implementation receiver isn't involved. It also verifies that the setup itself works so any problems are with the receiver.	The sACN viewer should display the full universe of data from the source with all values listed correctly.	
Acceptance Test								
100-1	Implementation Source	IPv4 Multicast	Avolites Titan	Vision Visualiser	The sender is setup to send a predefined pattern to the visualiser. This should cause all lights to turn on in white, then the far lights (back-lights) to switch to red, switch to blue and then all lights to turn off.	That the implementation sender works with a real-world industry sACN receiver (Vision Visualiser) implementation and can work in a realistic lighting setup to perform a realistic lighting sequence that is similar to one produced using real-world commercial software (Avolites Titan).	The pattern as described.	Test could not be completed in situ due to COVID-19 shutdown but using the same test setup as the interoperability tests was able to record what would have happened and show it was as expected.
100-2			Implementation	Vision Visualiser	The implementation sender recreates the exact same			
101	Implementation Receiver	IPv4 Multicast	Avolites Titan	Implementation	The lighting board should send a sequence of data which causes 1 of 4 groups of 5 channels to turn on in sequence so lights 1-5 turn on then turn off as lights 6-10 turn on.	That the implementation receiver works with a real-world industry sACN sender implementation and can work in a realistic lighting setup to perform a realistic lighting sequence.	The pattern as described.	Test could not be setup and completed due to COVID-19 Shutdown. Expected to have passed based on the interoperability testing which uses a similar setup
102	Implementation Source + Receiver	IPv4 Multicast	Implementation	Implementation	The implementation sender starts a predefined sequence where every channel in universe 1 follows a sine wave pattern with each channel offset slightly from the previous. This creates an effect which is commonly used in entertainment lighting refered to as a 'dim chase'.	The implementation receiver and sender working together to perform a realistic lighting sequence.	The pattern as described.	
103	Implementation Source + Receiver Discovery	IPv4 Multicast	Implementation	Implementation	Straight after test 3 the receiver checks for discovered universes. Done immediately after test 3 to reduce the time the demonstration takes.	That universe discovery works.	The implementation source appearing in the implementation receiver discovery list.	
104	Universe Synchronisation	IPv4 Multicast	Implementation	Implementation	Two data packets on universe 1 and 2 which are synchronised to universe 2 are manually sent using the implementation sender. The implementation receiver is shown to not output any values. A synchronisation packet is then sent to universe 2 and the receiver is shown to now output both universes of data simultaneously.	That universe sychronisation works between the implementation sender and receiver.	No data shown until the synchronisation packet is sent to trigger synchronisation.	

Key:
OS traceroute utility: The traceroute utility that exists on the sending test machine windows 10 operating system. Run using the command "".
OS ping utility: The ping utility that exists on the sending test machine windows 10 operating system. Run using the command "".
ipconfig: Displays the network configuration of the interfaces on a windows machine, shows the IP that each interface is bound to. Run using the command "ipconfig".
Avolites Titan: The industry source implementation used for interoperability testing, further details described in the tools section of the report. The showfile used for each test is included in the "Avolites Titan Test Showfiles" folder.
Vision Visualiser: An industry receiver implementation in the form of a visualiser which receives sACN input and simulates lighting output. Further details in the tools section of the report.
Implementation: The 'demo_src' implementation of the sending side of the library written as part of this project. This is the focus of the tests.

Note that the # column refers to the test number which corresponds to the folder number used for that test within the 'Test Results' folder.

During the acceptance test the lighting layout used with the visualiser is detailed in the report.

* For each universe x, x in [1, 16], the range for that universes channels is [(x - 1) * 10, x * 10) so for example for universe 7 the expected range is: [(7-1) * 10, 7 * 10) = [6 * 10, 7 * 10) = [60, 70) so values are expected to be from 60 inclusive to 70 exclusive.