	Test Name	IP Mode	Sender	Receiver	Setup	Shows	Expected	Result
Pre-1	Test Setup Verifica	tion Testing - Ens	ures the setup is	as described in th	e test setup diagram.			
0.1 7	Fraceroute	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.	
0.2 F	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.	
		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 address. Both should have a subnet mask of 255.255.255.0	
0.4 8	Setup Control	IPv4 Multicast	Avolites Titan	sACN viewer	Avoiltes 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.	
sACt	N Receiver Implem	entation Interope	rability Testing					
1 5	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 continously 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.	
2 1	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.	
	Jnicast Two Jniverses	IPv4 Unicast	Avolites Titan	Implementation	Same as test 3 except using unicast for data sending.	That the implementation receiver can work over unicast aswell as multicast.	Same as test 3	
	ndependent noving channels	IPv4 Multicast	Avolites Titan	Implementation	Avoiltes titan sends a universe of data with a sine wave 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.	
5 F	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.	
6 H	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 output file should show the values from all 16 universes with all the values from a specific universe within the predefined ranges.	
Tests	s which could not b	e performed as u	nsupported by inc	ustry implementa	tion. Expected to pass as previous intergration tests sh	now this functionality works.		
	Pv6 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/available.
	Pv6 Unicast Two Jniverses	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	
	Jniverse Syncronisation Hold	IPv4 Multicast	Avolites Titan	Implementation	Avoiltes 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/available.
8	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.	
0	Jniverse Discovery No Jniverses	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.
	Jniverse Discovery Multiple Jniverses	IPv4 Multicast	Avolites Titan	Implementation	Avoiltes 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.	

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 ".
operating Displays the network configuration of the interfaces on a windows machine, shows the IP that each interface is bound to. Run using the command "pconfig". Interfaces the provide but for the Fedoral Linux operating system. Run using the command "interfaces to bound to. Run using the command "interfaces" interfaces to bound to. Run using the command "interfaces to bound to. Run using the command "command "interfaces" interfaces to bound to. Run using the command "interfaces to bound to. Run using the command to. Run us

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.

# T		IP Mode	0	Describer	0-4	Ob	Francisco	Domite.		
			Sender			Shows	Expected	Result		
Pre-Tes	re-Test Setup Verification Testing - Ensures the setup is as described in the test setup diagram.									
0.1 Tra	aceroute	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.			
0.2 Ping	ıg	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.			
		IPv4	ipconfig	ipconfig		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 Set sAC	tup Control CN viewer	IPv4 Multicast	Avolites Titan	sACN viewer	Avoilites 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.			
Visi	tup Control ion Visualiser	IPv4 Multicast	Avolites Titan	Vision Visualiser		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.			
		ntation Interoperal								
1 Sim	nple Data Send	IPv4 Multicast	Implementation	Vision Visualiser	The implementation sender sends a single universe (universe 1) of static data (with a zero startcode and all channels set to full) to the vision visualiser	That the implementation sender can send data to the visualiser industry receiver.	All the lights on sACN universe 1 should come on at full and stay there.			
	nple Data Send		Implementation	sACN viewer	Same as test 1 but with the sACN receiver.	That the implementation sender can send data to the debug external receiver.	The data readout for sACN universe 1 should show all channels at full and a zero startcode.			
		IPv4 Multicast	Implementation	sACN viewer	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.			
	icast Two iverses	IPv4 Unicast	Implementation	Vision Visualiser	Same as test 3 but using IPv4 unicast.	That the implementation sender can send data on multiple universes using unicast to an industry receiver.	Same as test 3			
Two	o Universes	IPv6 Multicast	Implementation		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			
	6 Unicast Two iverses	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			
	lependent ving channels	IPv4 Multicast	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'.	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.			
8 Rap	pid 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.			
		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 varies the values of the channels between different ranges depending on the 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 the channels for each universe varying between the correct ranges. Vision visualiser could not be used for this as the version available only supports 4 universes.			
	scovery No iverses	IPv4 Multicast	Implementation		and eventually displays the started sender in its discovered sources list with no universes.	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 eventually** displays the started sender in its discovered sources list with no universes.			
Uni	scovery Multiple iverses	IPv4 Multicast	Implementation		and eventually displays the started sender in its discovered sources list with universes 1, 2 and 3 registered correctly.	That the implementation sender sends universe discovery packets with universes registered correctly so that they can be received and used by an industry receiver.	The sACN viewer eventually** displays the started sender in its discovered sources list with universes 1, 2 and 3 registered.			
			nsupported by ind							
Syn		IPv4 Multicast			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.		
Syn	iverse nchronisation to Universes	IPv4 Multicast	Implementation		The implementation sender sends data packets with sychronisation address 1 to but universes with the first universe of data bringing one set of lights to full and the second universe of data bringing another et 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 compatable 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.	nuncuonany WORKS.		

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 "pconfig".

Avoites Titan: The industry source implementation used for interopreshibly testing, further details described in the tools section of the report. The showfile used for each test is included in the "Avoites 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 Visualizer: An industry receiver implementation in the form of a visualizer within receives acXN 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.

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, 18], 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.

#	Test Name	IP Mode	Sender	Receiver	Setup	Shows	Expected	Result		
Pre	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.		The traceroute should show only a single hop from the sender to receiver machine with no routers present.			
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		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	command to display their assigned addresses.	and the state of t	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		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.			
Ac	ceptance Test									
1	Implementation Source	IPv4 Multicast	Implementation		predefined pattern to the visualiser. This should	That the implementation sender works with a real-world industry sACN receiver implementation and can work in a realistic lighting setup to perform a realistic lighting sequence.	The pattern as described.			
2	Implementation Receiver	IPv4 Multicast	Avolites Titan			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.			
3	Implementation Source + Receiver	IPv4 Multicast	Implementation		The implementation source sends a predefined sequence which increases each channel 1-10 from 0 to full to 0 independently. This pattern moves up the channel count and then back down appearing to bounce. The receiver outputs the received data to the command line showing the changing values.	The implementation receiver and sender working together to perform a realistic lighting sequence.	The pattern as described.			
4	Implementation Source + Receiver Discovery	IPv4 Multicast	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.			
5	Universe Synchronisation	IPv4 Multicast	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.		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 ".
of ping utility: The ping utility that exists on the sending test machine windows 10 operating system. Run using the command ".
of ping utility: The ping utility that exists on the sending test machine windows 10 operating system. Run using the command ".
of ping utility: The ping utility that exists on the sending side of the ping utility that in the tools send on the report. The showfile used for each test is included in the "Avoiltes 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. The 'demo_src' implementation of the sending side of the library written as part of this project. This is the focus of the tests.

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