sACN Project Timeline

April 2018
Existing Base Library Development Stops

Existing library adopted and maintenance stage ends

September 2019
Project Start

16th September 2019
Requirements Specified in DOER

Initial coding as part of discovering extent of existing implementation
End of requirements phase

27th September 2019
Milestone Timeline Created

End of Design Phase

20th October 2019
Initial implementation of sending/receiving

23rd October 2019
Initial implementation of universe synchronisation

26th December 2019
Problem found via initial testing with data sending/receiving

30th December 2019
Universe Synchronisation and data sending/receiving working

31st December 2019
Initial implementation of universe discovery receiving mechanism

12th January 2020
Design change to use multiple threads to allow sending periodic universe discovery packets

17th January 2020
New threaded design implemented

3rd February 2020
All universe discovery tests pass

End of Implementation Phase

6th February 2020
Working on SSH tests across lab machines

9th February 2020
First universe send/recv, discovery and synchronisation SSH lab machine tests pass

12th February 2020
A preliminary test of sender with a real-world software acting as receiver suceeds

7th March 2020
Critical bug found preventing receiving on linux, implementation of new error system to find source of bug and fix

10th March 2020
Bug found and fixed with windows/unix specific code sections

26th March 2020
More tests created, mainly focused on single machine integration tests

2nd April 2020
Compliance Tests Completed

13th April 2020
Interoperability and Acceptance Tests Completed

18th April 2020
Final Testing

End of Testing Phase

Did not happen due to COVID-19, Acceptance test acts as stand in.
System demo'd in real-world environment with real-world software

24th April 2020
Library packaged for deployment (inc. report writing).

27th April 2020
'Final Deployment' (Submission)

End of Deployment Phase

Summer 2020
Once marking finished and permission granted code is uploaded back to community to start maintaince phase.
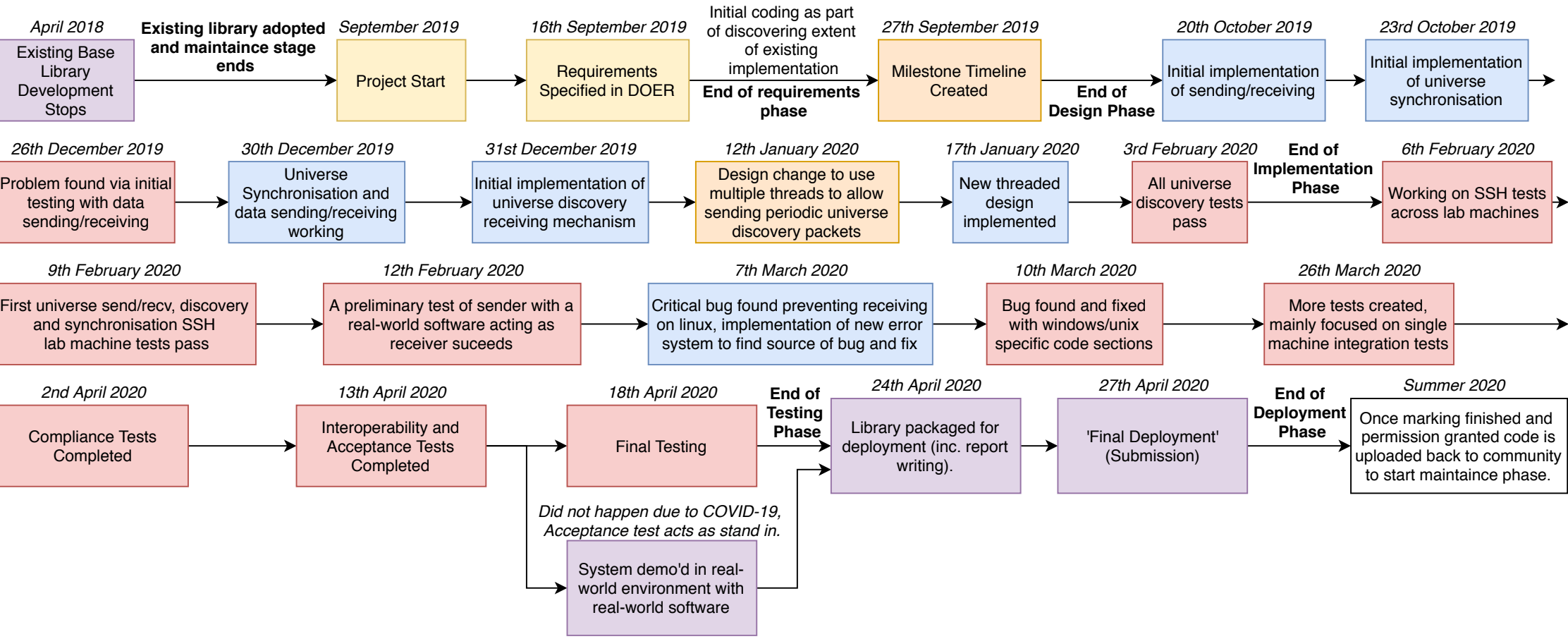
Key: → Arrows show the passage of time*, not to scale

Waterfall development stages:

Requirement Analysis Stage	Deployment Stage
Design Stage	Implementation Stage
Testing Stage	

End of waterfall development stages marked in bold

Dates shown in italics, dates drawn from git commits and approximate

* During this time the previously marked task continues and the next task begins, this diagram should be read as 'from the previous date to the next date the current task was completed'