# Two phase commit

# (Setup 1)

# Assumptions

1. There is a single coordinator and the coordinator does not fail.
2. There is a single client and is aware of the coordinator.
3. The service implements a key-value store which is replicated by all the nodes in the network. (The key value store has only add operation so there are no destructive updates)
4. Replicas do not fail
5. Messages sent may be delayed (bounded delay)
6. Messages are not reordered. (They are delivered in the send order)
7. Messages are not lost. (This is similar to reliable eventual delivery of messages).
8. Links in the network do not fail. (A link failure is like connection between two machines is broken and all messages sent between this pair are always dropped)

# (Setup 2)

# Assumptions

1. There is a single coordinator and the coordinator does not fail.
2. There is a single client and is aware of the coordinator.
3. The service implements a key-value store which is replicated by all the nodes in the network. (The key value store has only add operation so there are no destructive updates)
4. Replicas do not fail
5. Messages sent may be delayed (bounded delay)
6. Messages are ~~not~~ reordered.
7. Messages are ~~not~~ lost.
8. Links in the network do not fail. (A link failure is like connection between two machines is broken and all messages sent between this pair are always dropped)

# (Setup 3)

# Assumptions

1. There is a single coordinator and the coordinator does not fail.
2. There is a single client and is aware of the coordinator.
3. The service implements a key-value store which is replicated by all the nodes in the network. (The key value store has only add operation so there are no destructive updates)
4. Replicas ~~do not~~ fail
5. Messages sent may be delayed (bounded delay)
6. Messages are not reordered. (They are delivered in the send order)
7. Messages are not lost. (This is similar to reliable eventual delivery of messages).
8. Links in the network do not fail. (A link failure is like connection between two machines is broken and all messages sent between this pair are always dropped)

# (Setup 4)

# Assumptions

1. There is a single coordinator and the coordinator does not fail.
2. There is a single client and is aware of the coordinator.
3. The service implements a key-value store which is replicated by all the nodes in the network. (The key value store has only add operation so there are no destructive updates)
4. Replicas ~~do not~~ fail
5. Messages sent may be delayed (bounded delay)
6. Messages are ~~not~~ reordered.
7. Messages are ~~not~~ lost.
8. Links in the network do not fail. (A link failure is like connection between two machines is broken and all messages sent between this pair are always dropped)

# Properties:

In all the above setups the challenge is to design Failure Detection (FD), Failure Recovery (FR), Membership Manager (MM) such that the following properties are satisfied.

1. Safety Property :
   1. All replicas are always consistent: The values at same index on two different replica should always be same.
   2. A successful write by a client followed by a read from any replica should always be successful.
   3. A failed write by a client followed by a read on any replica should also always fail.
2. Convergence Property:
   1. Eventually all the replicas converge on a unique state. (if the client is bounded)