Manual code analysis report

During the analysis we found the following problems:

Problem	Comment/Suggestion
Method setSignalMapper(): calling is not guaranteed, object can be null (NullReferenceException)	It should be called from/initialized in the constructor to avoid NullReferenceExceptions in the future.
Assert in makedDecision() is not so elegant	Instead of assert, throwing an exception would be the best way to handle.
Method sendStatus() is not called (responsible for periodic heartbeat) but the comment says: "informs its neighbors about its status". This class does not meet this requirement.	Calling sendStatus() periodically.
Constructor: what if neighbourStatuses or other parameters are null?	A null check in the constructor, setting these private attributes to null by default.
Method combineDecision(): local variable names 'd1' and 'd2' do not say much.	A more communicative name would be sufficient e.g. 'localDecision' and 'distributedDecision'.
Values of the private attributes should be set through setter methods.	Setter methods for the private fields.
Logging name inconsistency e.g. 'METHOD_ENTER'	A 'METHOD_END' would be also needed at the end of the method or it should be named simply as 'METHOD'.
Method makeLocalDecision() is not so readable.	It should be divided into multiple functions e.g. makeLocalFacingDecision, makeLocalStraightDecision etc.
Class is not thread safe e.g. no atomicity, no volatile attributes etc.	Prepare the class for running in a multithread environment if necessary.
The expiry of the heartbeat signal should not be only checked when a new signal comes.	Periodic check would be required e.g. on another thread.
makeLocalDecision(): in the lower 2 branches checking the first condition in unnecessary	Checking should be removed.