**Topology tests**

This section tests common topologies. All graphs start from sources, to flows and then to sinks. Sinks cannot reconnect back to flows and sources. In these tests STRIMMBuffer is the element which passes through the graph. This buffer has only a dataID which is set when the sample is created in a source and a status value (1 is OK).

Each flow has a Merge object at input and a Broadcast object at output. The Merge object will take several inputs and put them into a List<STRIMMBuffer>. Every time one of the input changes then the run function for that flow will be activated and a new sample will be emitted. This likely means that a flow with several inputs will be running much faster than flows with 1 output.

In much the same way Sink objects also have Merge inputs, and a new sample is passed through the Sink when there is a change at the input. It is also possible to pass a Source directly to a Sink.

The STRIMMBuffer.status flag indicates whether a sample continues to flow and be processed by the akka graph. If it is set to 0 then it will no longer be processed.

The STRIMMBuffer object is the base class to all other packets flowing through the akka graph.

These tests all uses SourceBaseMethod to create the STRIMMBuffer, FlowBaseMethod and SinkBaseMethod which just prints the dataID and status to the console. Non have a config attached to their entry in the JSON.

This setup shows that all common topologies are supported as well as the ability to control the flow of samples through the akka graph. Also the use of config files for source, flow and sink and using source,flow and sink Methods with the name of the method identified in the json.