# CPS:Small:The Roles of Communications in Lane Merging Systems

N. F. Maxemchuk, Columbia University, Project 1035178

Our objective is to design a collaborative lane merging system, and to use probabilistic verification to prove that it operates safely.

Our initial system is simple. We assume that all cars have identical systems, consisting of sensors that can determine the distance to other cars, automatic cruise control that can select a speed that is less than or equal to the speed that the driver sets and the speed of the car in front of it, and communications with nearby vehicles. The objective is for one vehicle to merge between two others in a traffic lane.

We use a locking system, similar to that in databases to guarantee that cars only participate in one merge operation at a time. The cars communicate to obtain locks, then the automatic cruise control system in the following car in the traffic lane slows down to make room for the merging car. The cruise control system in the merging car adjusts it speed to a value that is less than or equal to the speed of the lead vehicle in the traffic lane and the driver's setting. The lead car is responsible for maintaining the pace and warning the others when it is necessary to slow down or brake. The lane change operation is manual. The driver receives audio or visual signals to indicate that the merge is in progress, that it is safe to merge, or that the merge must be aborted.

We model the control portion of the protocols in the three vehicles as extended finite state machines, and use probabilistic verification to guarantee that the likelihood of failure in the composite machine for the three vehicles is less than a specified number. The failures that we consider are locks that are not released at the end of an operation, which prevents vehicles from participating in another merge, and locks that are released in some, but not all of the vehicles, and can result in an accident.

We are working on our third generation of the protocol. The first version was built on top of the transmission layer. Each transmitted message was recovered in the protocol. The second and third versions are built on top of a communications layer. A simple ARQ protocol was used in the second version. It consolidated the recovery procedures in the first version. Although the merge protocol seemed simple, using probabilistic verification, we found a combination of two unlikely errors that would cause accidents. In the third version we are using a subset of the services provided by the PI's "Reliable Neighborcast Protocol." With this communication layer we are able to eliminate all timers in the merge protocol, and guarantee consistent information in the three vehicles. We are in the process of designing and testing this protocol. Our objective is to determine a small set of communications services that can provide a large reduction in the operations that must be performed in cooperative driving systems.