Proceedings of the IEEE ITSC 2006
2006 IEEE Intelligent Transportation Systems Conference
Toronto, Canada, September 17-20, 2006

Real-Time Coordination of Autonomous Vehicles

Mélanie Bouroche, Barbara Hughes and Vinny Cahill
Distributed Systems Group, Computer Science Department,Trinity College Dublin
{melanie.bouroche, barbara.hughes, vinny.cahill}@cs.tcd.ie

***Abstract*— Autonomous vehicles seem to be a promising approach to both reducing traffic congestion and improving road safety. However, for such vehicles to coexist safely, they will need to coordinate their behaviour to ensure that they do not collide with each other. This coordination will typically be based on (wireless) communication between vehicles and will need to satisfy stringent real-time constraints. However, real-time message delivery cannot be guaranteed in dynamic wireless networks which means that existing coordination models that rely on continuous connectivity cannot be employed.**

In this paper, we present a novel coordination model for autonomous vehicles that does not require continuous real-time connectivity between participants in order to ensure that system safety constraints are not violated. This coordination model builds on a real-time communication model for wireless networks that provides feedback to entities about the state of communication. The coordination model uses this feedback to ensure that vehicles always satisfy safety constraints, by adapting their behaviour when communication is degraded. We show that this model can be used to coordinate vehicles crossing an unsignalised junction.

I. Introduction

Autonomous vehicles seem to be a promising approach to both reducing accidents and alleviating traffic congestion by improving road usage [1]. These vehicles could be private vehicles, shared cars like cybercars [2], or commercial vehicles on dedicated infrastructures (e.g., load transportation units). Such vehicles would drive autonomously, using sensors to follow the road and to detect other vehicles and possible obstacles. Autonomous vehicles may also communicate with each other to cooperate by sharing information and to coordinate their actions. Situations where vehicles need to coordinate their actions include crossing unsignalised junctions, overtaking [3] and platooning [4]. This coordination typically requires real-time communication over a wireless network. Furthermore, to allow vehicles to operate within existing infrastructures without the need for new road facilities, this network may need to be infrastructure-free (or *ad hoc*). However, since radio communication quality varies hugely over time and space, communication in wireless networks, in particular in ad hoc networks, is not reliable. Real-time communication is even harder to provide over wireless networks, and cannot be guaranteed under realistic assumptions about the network [5].

For this reason, we believe that a coordination model for safety-critical applications such as autonomous vehicles, needs to take into account the fact that real-time communication is unpredictable. To allow vehicles to make progress in the presence of unreliable communication, they need to receive some feedback in real-time about the state of communication. We are using a communication model in which every communicating entity is informed in real-time about the proximity (geographical area) in which real-time communication, within an application-specified latency, can be achieved. This proximity varies over time, depending on the topology of the network. This Space-Elastic Communication Model enables us to build a coordination model for autonomous vehicles in which vehicles can adapt their behaviour depending on the state of communication to ensure that specified safety constraints are never violated. For this purpose, system-wide safety constraints are first formalised, and then translated into constraints on the behaviour of individual entities.

In this paper, we show that this coordination model can be applied to coordinate vehicles crossing an unsignalised junction. An unsignalised junction is one where the traffic flow is not governed by traffic lights, stop signs, or give-way signs [3]. The goal is for vehicles to coordinate their behaviour so that there is at any time at most one vehicle on the junction. We assume that vehicles are informed of the presence of the junction either a priori (e.g., via onboard maps), or in real-time (e.g., via road markers), and we do not consider this aspect. Also, vehicles are assumed to be able to sense whether the junction is empty. We aim to provide a solution in which vehicles can cross without stopping or slowing down, provided that the communication coverage is sufficient and that there is nobody on the crossing. This scenario is particularly challenging as vehicles arriving from different directions need to coordinate their behaviour, under strong real-time constraints. Furthermore, the number and the identity of the vehicles are not known in advance.

We begin by reviewing related work on autonomous vehicles, unsignalised junctions, and existing coordination models. We then briefly summarise the guarantees of the Space-Elastic Model in Section III, before introducing our coordination model in Section IV. In Section V, we demonstrate that the coordination model can be used to coordinate the crossing of an unsignalised junction, while Section VI presents some concluding remarks.

II. Related Work

In the last decade, the idea of autonomous, driverless vehicles has moved from the domain of pure science fiction, to a vision that should be achievable in the not-too-distant future [3]. Research in this area includes the search for adequate sensors and actuators [6], vehicle control algorithms [4], and assessment of the usability of such vehicles [2]. This has led to a number of results, including a practical demonstration

1-4244-0094-5/06/$20.00 ©2006 IEEE

of driverless vehicles following a road lane, overtaking a slower vehicle, and crossing an unsignalised junction [3], [7]. This scenario was also studied from a communication point of view in [8]. A similar scenario is also presented in [9], where robots coordinate their actions to cross a shared road section. In these experiments, communication is assumed to be reliable, or the unreliability is dealt with by sending a message several times. However, this will not always be sufficient to ensure message delivery in wireless networks, in particular for applications with strong real-time requirements such as autonomous vehicles. In safety-critical applications, the failure of these assumptions might lead to catastrophes.

A communication architecture for the cooperation of autonomous vehicles is presented in [10] and detailed in [11], however this is limited to infrastructure-based networks, and real-time issues are not examined. Similarly, existing coordination models for mobile autonomous entities, such as Linda In Mobile Environments (LIME) [12] do not provide real-time guarantees, and because of their best-effort nature, are not suitable for safety-critical applications. Therefore, due to to the unreliability of communication in real-time networks, none of the existing coordination models can be applied for safety-critical applications. Our coordination model builds on a real-time communication model which provides feedback about the state of communication to entities, to allow them to adapt their behaviour accordingly.

III. Space-Elastic Model

Communication in wireless networks is inherently less reliable than in wired networks because of the higher rate of link failures due to node mobility and varying signal strength [5]. Furthermore, message collisions are particularly hard to avoid, in ad hoc networks in particular, and cause unpredictable latency. Therefore providing real-time communication in wireless and ad hoc networks is particularly challenging.

To allow entities of safety-critical applications to make progress despite unreliable communication, a real-time communication model for wireless networks, including ad hoc networks, has been designed [13]. In this model, feedback about the state of communication is provided to message senders. We summarise in this section the specifications of this model and the guarantees that it provides.

A. Specifications

This model exploits the rationale defined in [14], i.e., the relevance of context to a particular geographical area, to guarantee real-time communication only within a geographical proximity. This proximity can be defined either absolutely (via GPS coordinates), or relatively around the entity (using an anchor point and a size) [15]. An entity wishing to send messages specifies the proximity within which it wishes these messages to be delivered. This proximity is called the *desired coverage*, and is used to bound message propagation. The entity also specifies the maximum latency $msgLatency$ within which the messages must be delivered, and the desired period $period$ for these messages.

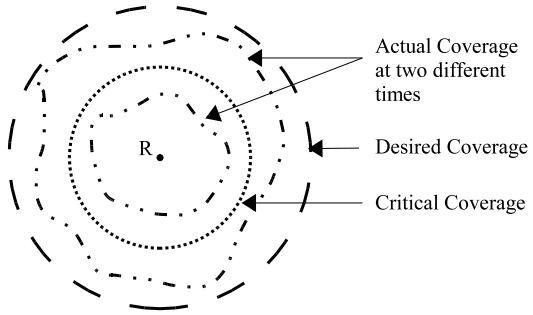

Fig. 1. Different coverages of the Space Elastic Model

Depending on the topology of the network (i.e., the distribution of the nodes and the quality of the wireless links), it might not be possible over some period of time to deliver a message in time to all interested entities within the desired coverage. Therefore, the size of the area in which timely delivery of messages is provided, called the *actual coverage*, changes over time. In the worst case, no communication is possible; this corresponds to an actual coverage of null. The sender is notified in real-time of changes in the actual coverage, within a bounded time, $adaptNotif$. Therefore, an entity knows within $msgLatency + adaptNotif$ after sending a message the area in which it has been delivered, and can adapt its behaviour accordingly. If the actual coverage becomes smaller than one or more thresholds, called *critical coverage(s)*, the sender might need to take into account that it cannot communicate in a area wide-enough to maintain safe operation, and might need to adapt its behaviour. Variations of the actual coverage around the desired and the critical coverages are shown in Figure 1.

B. Guarantees

In this communication model, guarantees about real-time communication can be provided to both message senders and entities interested in the type of messages it sends. Senders are guaranteed to be able to communicate with a given latency in the actual coverage, and to be notified within a given time delay if this coverage changes. Senders can therefore adapt their behaviour depending on the value of the actual coverage. On the other hand, entities present within the actual coverage at the delivery time of a message of a type in which they have expressed interest, are guaranteed to receive it. We define an entity as present within the actual coverage once its is able to receive messages after arriving in the communication coverage. This will take an implementation-dependent time, $present$, which might be necessary to include the entity in the real-time route for example. We will see in the next section that these guarantees are easily exploitable to ensure system-wide constraints while allowing progress of entities.

IV. Coordination Model

Using the Space-Elastic Model, entities can be notified in real-time about the proximity in which they can communicate, and can adapt their behaviour depending on its size. For

example, a vehicle intending to cross an unsignalised junction needs to communicate in an area wide-enough to ensure that other vehicles intending to cross this junction will receive its messages. It is often not sufficient for a single entity to adapt its behaviour depending on the state of communication to ensure that some safety constraints are not violated. In this case, entities need to coordinate their actions. For example, once a vehicle announced that it will cross an unsignalised junction, other cars should not cross.

In this section, we present a real-time coordination model for mobile autonomous entities based on the notion of distributed responsibility. We first define a formalism to express high-level, implementation-independent, system-wide, safety constraints. We then use the notion of responsibility to translate these safety constraints into constraints on individual entities.

A. Specifying the Safety Constraints

Safety constraints typically include constraints on the actions of entities and on their state, as well as their distance to each another. This exploits the rationale that entities need to coordinate their behaviour when they are in the same vicinity, the definition of which is application-specific. For example, cars need to coordinate their behaviour when they are close. In this section, we introduce a formalism to express these concepts and their interactions.

1) Scenario, Modes and States: A *scenario* encompasses a set of *entities* $E_1, E_2, .., E_n$, a *goal*, and some *safety constraints*. The behaviour of an entity is composed of a set of modes of operation (*modes*) that describe the actions it can take, and the transition rules between these modes. Modes should be defined so that an entity is always in one of its modes, i.e., transitions between modes are assumed to be instantaneous. For example, the modes of a car can be `stopped`, `accelerating`, `braking` or `going_at_maximum_speed`. We use M_i to denote the set of modes of entity E_i.

The situation of an entity at a given time is described by its *state* which encompasses its mode and some additional application-specific information, for example, the position of the entity. We denote the set of states of entity E_i, as S_i. The information contained in the state of an entity should be sufficient to characterise the state of the entity for the purpose of the application. For example, the state of a car could encompass its mode, position, speed, and direction. We define the function $\mathcal{M} : S \mapsto M$, from the set of states S to the set of modes M, that returns the mode of a given state.

2) Compatibility: We say that a set of states $(s_1, s_2, ..., s_n) \in S_1 \times S_2 ... \times S_n$ is *compatible,* noted $\mathcal{C}_\mathrm{s}(s_1, s_2, ..., s_n)$, if the safety constraints are not violated when some entities are simultaneously in these states. For example, the states of two cars are compatible if their positions are far enough away.

Similarly, a set of modes $(m_1, m_2, ..., m_n) \in M_1 \times ... \times M_n$ is *compatible,* noted $\mathcal{C}_\mathrm{m}(m_1, m_2, ..., m_n)$ if, when some entities are simultaneously in these modes, their states are compatible. So if we define, for $m \in M_i$, $S_{i,m}$ as the set of states of the entity E_i, in which it is in mode m, i.e.,

$$S_{i,m} = \{s \in S_i | \mathcal{M}(s) = m\} \ ,$$

mode compatibility can be defined as:

$$\mathcal{C}_\mathrm{m}(m_1, m_2, ..., m_n) \text{ iff}$$
$$\forall (s_1, s_2, ..., s_n) \in S_{1,m_1} \times ... \times S_{n,m_n}, \mathcal{C}_\mathrm{s}(s_1, s_2, ..., s_n) \, .$$

While the notion of state compatibility captures whether the safety constraints are violated at a given time, mode compatibility enables us to make predictions that no incompatibility will happen (while entities are in these given modes). It must be noted that if the modes of a set of entities are not compatible, it does not imply that the safety constraints will be violated. For example, the modes `stopped` of one car and `going_at_maximum_speed` of another are not compatible, as entities might collide into each other when they are in these modes, but if they are far enough apart, the safety constraints will not be violated, hence their states are compatible when they are far enough apart.

3) Expressing the Safety Constraints: The safety constraints can be expressed as a set of incompatibilities between states, including constraints on the relative distance of entities (noted distance$(position1, position2)$). For example, the fact that two cars should not collide into each other could be expressed as:

$$\mathcal{C}_\mathrm{s}(s_{car1}, s_{car2}) \text{ iff}$$
$$\neg[(\text{distance}(s_{car1}.position, s_{car2}.position) < d)] \, ,$$

where $\neg$ is a notation for logical negation. This example is simple, but illustrates that the formalism is high-level and implementation-independent. This formalism captures all the salient details of the safety constraints, and allows safety requirements for mobile autonomous entities to be expressed simply. Note that this expression is emphasising when the states of a set of vehicles are not compatible (as opposed to when they are), as the aim of the coordination model is to prevent incompatibilities from happening.

B. Translating the Safety Constraints

High-level system-wide safety constraints, while being simple and quite intuitive to state, are not easily exploitable as such. In our experience, it is non-trivial to deduce the necessary and sufficient constraints on individual entity's behaviour from such safety constraints, or even to check that some specification of the entity's behaviours ensures that these safety constraints will not be violated. To ease this process, we introduce a number of concepts which can be used to derive constraints on entities.

1) Responsibility: For every possible incompatibility between the states of two entities, i.e., possible violation of one of the safety constraints involving these two entities, at least one of them needs to ensure that it will not occur. We say that this entity is *responsible* for the incompatibility. The responsibility can be attributed to entities of a certain type or

TABLE I
COORDINATION MECHANISMS

Mechanism	Meaning
Adapt	Perform another action than the one planned
Delay	Perform a planned action later than initially planned
Transfer responsibility	Communicate with other entities

to entities in a certain role. For example, traffic light entities might be responsible to ensure that cars do not go through a crossing when the light is red. Similarly every car might be responsible for ensuring that it does not collide into cars in front of it, so following cars are responsible for possible state incompatibilities with cars in front of them. Responsibility might be attributed a priori or in real-time, and might be transferred. However, at any time, at least one entity must be responsible for each possible incompatibility.

This notion of responsibility is the first step in the translation of system-wide safety constraints. It allows the distribution of the duty of ensuring safety constraints over entities. Being responsible for an incompatibility implies constraints on the entity's behaviour, as it should ensure at any time that the incompatibility will not happen. To this effect, the responsible entity can use three different mechanisms: it can adapt its behaviour, delay its own actions, or communicate with other entities. These mechanisms are detailed below and summarised in Table I.

2) Coordination Mechanisms:

Adapting its behaviour: A responsible entity can have information about the modes that other entities can be in, both a priori (by previous knowledge) and in real-time, by messages or sensor information. Using this information, a responsible entity can adapt its behaviour to always be in a mode which ensures that the safety constraints will not be violated.

Delaying actions: A responsible entity can ensure that the incompatibility for which it is responsible will not happen by delaying an action that can trigger this incompatibility (i.e., delay switching to a mode in which an incompatibility might occur). It can delay its action until it gets information that it is safe to undertake it, or it can delay its action until it has warned all entities that it will undertake it.

When an entity needs to warn other entities about a mode switch that it is intending to undertake, it must do so at least a predefined delay Δ in advance. Using the Space-Elastic Model presented in the Section III, the constraints on Δ can be derived. This delay must ensure that all incoming entities have been informed of the planned mode switch (this takes $msgLatency$), and after that, that those that will not have time to adapt their behaviour are gone (we call this time $leavingTime$), and that those that have time to adapt their behaviour have done so (this duration is denoted $O\_reaction$). This requires:

$$\Delta \geq msgLatency + \max(leavingTime, O\_reaction)\,.$$

The responsible entity must also ensure that after its message has been delivered, it will have time to be notified of the proximity on which they were delivered (this duration is bounded by $adaptNotif$), and will have time to cancel its mode switch if the delivery zone is not big enough (the required duration is called $R\_reaction$). This requires:

$$\Delta \geq msgLatency + adaptNotif + R\_reaction\,.$$

So, the value of Δ can be derived:

$$\Delta = msgLatency + \max(leavingTime, O\_reaction, adaptNotif + R\_reaction). \quad (1)$$

The responsible entity also needs to be able to communicate on a proximity big enough so that mobile entities will receive its message early enough to react to it.

Transferring responsibility: Another means for responsible entities to ensure that the incompatibility they are responsible for will not occur, is to warn other entities that the incompatibility might occur. Other entities are then expected to change their behaviour to prevent the incompatibility. The responsible entity might include its state and mode in the message. They can then be used by entities receiving the message to avoid the incompatibility. Messages need to be sent periodically over a proximity big enough, to ensure that entities approaching will receive a message early enough to be able to react to its contents if necessary.

An entity sending a message is notified about the delivery area, but not whether there was any entity within this area, so it does not know whether any entity actually received the message. Therefore, entities that receive the message become responsible to ensure that no incompatibility arises with the entity that sent it, which corresponds to a *transfer of responsibility*. This transfer is however only partial (as the responsible entity remains responsible for the incompatibility in relation to other entities).

3) Contracts between Entities: A responsible entity can use a combination of the three mechanisms mentioned above to ensure that the incompatibility for which it is responsible will not occur. This must be decided a priori, and can be seen as an implicit contract between the responsible entity and other entities. We have identified three types of contracts:

Contract without transfer: In this case, the responsible entity will not transfer its responsibility, and must always ensure, by adapting its behaviour if necessary, that the safety constraints are not violated. Other entities do not need to be aware of the contract, or even of the existence of the responsible entity.

Contract without feedback: The responsible entity must warn other entities at least a predefined $t_{warning}$ duration in advance when the safety constraints are liable to be violated. Other entities must be able, at any time, to react (i.e., change their behaviour to ensure that no incompatibility will happen) within $t_{warning}$ to a message from a responsible entity.

Contract with feedback: The responsible entity must also warn other entities at least $t_{warning}$ in advance when the safety constraints are liable to be violated. In this contract, however,

TABLE II

USE OF THE MECHANISMS BY THE CONTRACTS

Contract	Adapt	Delay	Transfer Responsibility
without transfer	R	R	-
without feedback	R, O	R	R
with feedback	R, O	R, O	R, O

entities can provide feedback to the responsible entity, when they cannot adapt their behaviour so that the safety constraints will not be violated. Therefore, the responsible entity must also be able to react, to ensure that no incompatibility will happen, to the feedback from another entity arising from its previous message, within $t_{warning} - t_{feedback}$. Other entities must be able at any time either to react within $t_{warning}$ to a message from a responsible entity, or to communicate within $t_{feedback}$ to this entity. This contract might include the exchange of further messages, but after the initial exchange the entities have discovered the presence of each other, and if necessary, the delay to exchange more messages can be included in the definition of $t_{warning}$.

The use of the three mechanisms by both responsible entities (R) and others (O) in the three contracts is described in Table II.

4) Zones: These contracts can be translated into geographical zones. We define the meaning of zones in this subsection, and derive their value in the following one.

Safety zone: The states of all the entities of a scenario must be compatible at all times. But the safety constraints actually impose constraints only on specific states, typically when two entities are "close" according to some application-specific definition. For this reason, we define the *safety zone,* denoted SZ, as the set of positions of entities where their states are liable to be incompatible with that of the responsible entity.

Consistency zone: If a responsible entity foresees that an entity could be in a state that is not compatible with its own state when that entity enters its safety zone, the responsible entity can choose to transfer its responsibility, by sending a message. In this case, it must do so early enough, so that the incoming entity will have time to adapt its behaviour (either by not entering the safety zone, or by changing its mode) to prevent the incompatibility. The zone in which this must be achieved is called the *consistency zone* of the mode M that the responsible entity is in, and denoted $CZ(M)$.

Critical coverage: If the responsible entity chooses to transfer its responsibility, to ensure that all incoming entities will have an accurate view of the state of the responsible entity when entering $CZ(M)$, timely communication must be guaranteed in a zone $CC(M)$ around $CZ(M)$. This corresponds to the *critical coverage* associated with mode M of the responsible entity. Upon failure of communication (i.e., when the critical coverage of its current mode is not covered), a responsible entity needs to *adapt* its behaviour, by entering a mode whose critical coverage is covered. The different zones

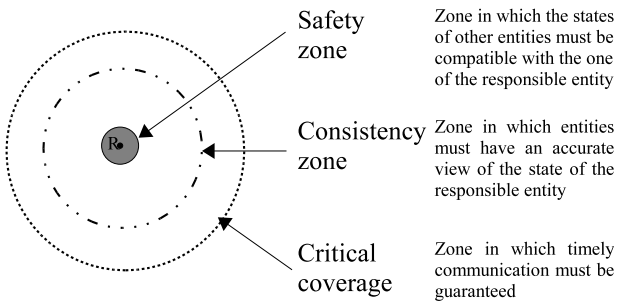

Fig. 2. Definitions of the different zones within the critical coverage

and their definitions are summarised in Figure 2.

5) Constraints on Entity's Behaviours: The different contracts imply different constraints on entity behaviour. We use the zones defined above to characterise these constraints.

Contract without transfer: In this case, the responsible entity has to ensure that it adapts its behaviour so that no entity will enter its (application-specific) safety zone or that when they do, their states will be compatible. This can be achieved by using either a priori information or information obtained in real-time, by messages or sensors.

Contract without feedback: In the case where entities obey a contract without feedback, other entities must be warned on time for them to adapt to the message before the possible incompatibility. This requires entities to be warned at least $t_{warning} = O\_reaction(M)$ in advance of an incompatibility. Furthermore, to ensure that incoming entities will have time to adapt their behaviour before arriving to the safety zone, a consistency zone of size

$$CZ(M) = SZ + t_{warning} \cdot v_{max}(M) \tag{2}$$

is needed, where M is the mode of the responsible entity, and $v_{max}(M)$ the maximal speed at which entities might approach an entity which is in mode M.

Messages must be sent in a zone wide enough to allow an incoming entity to receive them before entering the consistency zone. Furthermore, in the case where the coverage is not big enough, the responsible entity must have time to switch to another mode M' whose critical coverage $CC(M')$ is covered before the incoming entity enters it. So, to cater for the worst case, i.e., if an entity arrives at the maximum speed just after a message has been delivered, the following is needed:

$$CC(M) = (present + period) \cdot v_{max} + \max\Big(CZ(M), \big(adaptNotif + R\_reaction(M)\big) \cdot v_{max} + CC(M')\Big), \tag{3}$$

where M' is the mode, among all modes to which the responsible entity might switch to from M when $CC(M)$ is not covered, whose critical coverage is the biggest.

Contract with feedback: If, the entities are obeying a contract with feedback, other entities must be warned of a possible inconsistency by a responsible entity early enough

for them to have the time to either adapt their behaviour or send a message before the incompatibility arises. If we denote $O\_feedback(M)$ the time required for an entity receiving the message to send some feedback and this feedback to be delivered to the initially responsible entity, we need:

$$t_{warning} = \max(O\_reaction(M), \\ O\_feedback(M) + R\_reaction(M)) .$$

The expression for the consistency zone and the critical coverage as a function of $t_{warning}$ are the same than in the contract without feedback (see (2) and (3)).

6) Summary: Using all these concepts, the constraints on individual entity's behaviours can be deduced from the safety constraints expressed using the formalism presented in Section III. First, a responsible entity must be assigned for each possible incompatibility, and a contract type must be assigned to its interaction with other entities. The parameters of this contract depend on the characteristics of entities, and a mutual understanding of how they will act, with regard to others.

Responsible entities must at any time obey the contract(s) and either adapt their behaviour to ensure that incompatibilities will not happen or warn entities when incompatibilities are liable to happen. To warn entities they must send periodic messages in a coverage whose size depends on the value of the period and their contract type. When timely communication is not available in this coverage, responsible entities cannot transfer their responsibility and have to alter their behaviour to ensure that incompatibilities will not happen.

It must be noted that given a set of safety constraints, and some characteristics of entities, not every scenario is solvable when trivial non-progress making solutions (e.g., all entities idle) are not considered. The resolvability of a scenario can be assessed, but this is outside the scope of this paper.

V. Example: Unsignalised Junction

In this section, we demonstrate that the coordination model can be used to coordinate autonomous vehicles around an unsignalised junction and ensure that they will cross the junction safely. The coordination model allows vehicles to adapt their behaviour depending on the state of communication. If the communication is sufficient, vehicles will coordinate their behaviour in a similar fashion to existing solutions. However, if a vehicle cannot communicate within a given latency in a wide enough zone, it will cancel its crossing and wait until the communication is sufficient. Therefore, the coordination model will ensure that safety constraints are satisfied at all times, independently of the state of communication.

A. Specifying Safety Constraints

The first step in using the coordination model in this scenario is to formalise the safety constraint. This scenario encompasses a single type of entity: autonomous vehicles. The modes of these vehicles can be described as:

- `waiting` while the vehicle is approaching or waiting to cross the junction,
- `crossing` when the vehicle is actually crossing, and
- `not_interested` when the vehicle is in the vicinity of the junction but not interested in crossing it, for example because it is leaving the junction.

It can be noted that these modes are different from the ones presented for an autonomous car in the previous section, as the focus of this scenario is different (and the problem of safe driving on a safe lane is not considered). The state of a vehicle can be described by its mode, its position, and the duration for which it has been waiting to cross the junction (if relevant).

With these definitions, the safety constraint that there should at any time be only a single vehicle on the junction can be expressed as:

$$\mathcal{C}_{\mathrm{s}}(\mathrm{s}_{car_1}, \mathrm{s}_{car_2}, ..., \mathrm{s}_{car_n}) \text{ iff} \\ \neg(|\{\mathrm{s}_{car}/\mathrm{s}_{car}.\text{mode} = \texttt{crossing}\}| > 1) ,$$

where $|G|$ denotes the cardinality of the group G, i.e., the number of elements in G.

B. Translating Safety Constraints

Once the safety constraints have been formalised, the constraints on entity's behaviours to ensure that the safety constraints are not violated can be derived. This requires ensuring that no state incompatibility occurs and is achieved via the notions of responsibility and contract.

1) Responsibility: The safety constraint implies constraints on the states of all vehicles. The notion of responsibility is used to distribute the enforcement of constraints over entities. As long as no vehicle enters the junction, the states of the vehicles will remain compatible. Therefore, to ensure that the safety constraint is not violated, we make any vehicle entering the junction responsible to prevent incompatibilities. When about to enter the junction, a vehicle needs to ensure that the states of all vehicles will remain compatible. For this purpose, a vehicle will transfer the responsibility to other vehicles before it starts crossing, by sending them a message:

<`ID`, t_0, `position`, will cross at t_1>,

where `ID` is an identifier for the vehicle (for example, its registration number), t_0 is the time at which the message was sent, and `position` is the GPS position of the vehicle at this time. Other vehicles are then warned that this vehicle is intending to cross, and should defer crossing the junction until the other vehicle has finished crossing. This arrangement between entities can be seen as a contract.

2) Contract: Using the terminology of the coordination model, the contract between the vehicles is a contract without feedback: every vehicle should warn other vehicles that an incompatibility could happen at least $t_{warning} = O\_reaction$ in advance. Vehicles that receive a message from another vehicle that it will cross, should ensure that the safety constraints will not be violated by not entering the junction until the other vehicle has crossed (except if they have requested to cross first, as detailed below).

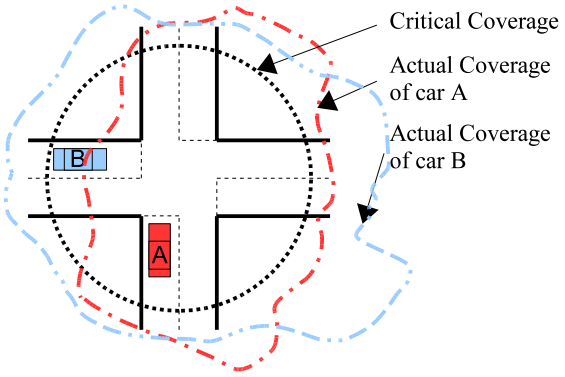

Fig. 3. Cars arriving at a crossing. Car A sent a message that was not delivered in a zone that covers the critical coverage, it cannot cross the junction. Car B sent a message that was delivered in a zone that covers the critical coverage, so car B can cross the junction safely.

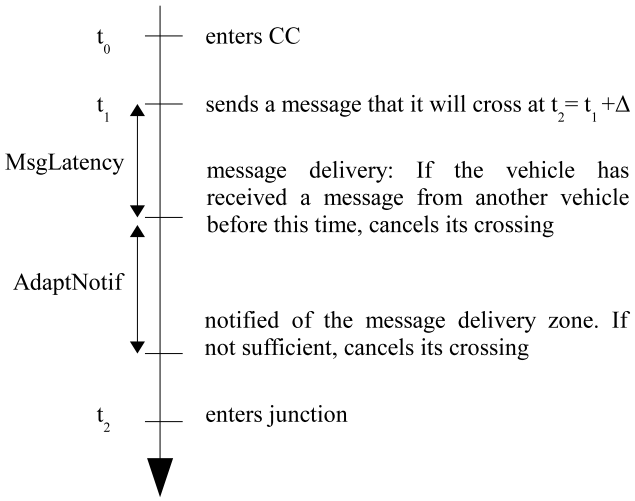

Fig. 4. Decisions sequence for crossing the junction, when the coverage is sufficient

3) Constraints on Entity's Behaviours: Respecting this contract implies a number of constraints on the behaviour of vehicles.

a) Request to cross: When a vehicle intends to cross, it needs to warn other vehicles early enough so that they will have time to stop before entering the junction. This requires them to be warned at least $O\_reaction = ST(v_{max})$ in advance, where $ST(v)$ is the time required for a vehicle to stop when travelling at speed v, and v_{max} the maximum vehicle speed when approaching this junction.

As defined in (1), this requires that a vehicle travelling at speed v intending to cross the junction, sends a message at least $\Delta(v) = t_1 - t_0$, in advance before starting to cross, with

$$\Delta(v) = msgLatency + \max(O\_reaction, \\ adaptNotif + R\_reaction(v)) .$$

In this example, $leavingTime = 0$ and $R\_reaction(v) = ST(v)$, as the responsible entity will need to stop before the junction if its message is not delivered over the critical coverage. This ensures that the message will be delivered in time for vehicles receiving it to stop before the junction, or if the communication is not sufficient, for the sending vehicle to be notified, and stop before the junction.

The request message needs to be sent from and be delivered in a zone of size CC, as defined in (3):

$$CC = (present + period) \cdot v_{max} \\ + \max\big(SZ + O\_reaction \cdot v_{max}, \\ (adaptNotif + R\_reaction) \cdot v_{max}\big) . \quad (4)$$

In this example, the safety zone SZ corresponds to the size of the junction. It must be noted that the definitions of SZ and CC in this case are absolute, i.e., not relative to the position of the vehicle.

The value in (4) ensures that a vehicle has enough time between the time it enters the critical coverage and the time that it arrives to the junction to send a message and be notified where it was delivered. If this delivery zone does not cover CC, the vehicle will not be able to cross. It has to cancel its crossing and start again (see Figure 3). Cars outside CC at the delivery time for the message will arrive at the junction after the vehicle has started crossing, so they will be able to sense whether it is still on the junction.

b) Simultaneous requests: To send a message indicating that it intends to cross, a vehicle needs to be in the critical coverage. Therefore, before any one of them can start crossing, all the other vehicles will have received its message. A simple algorithm to ensure that at most one vehicle crosses at a time can be devised as follows: if a vehicle receives a message that another vehicle intends cross before it sends a request to cross, it will not send its request. If a vehicle receives a request to cross from another vehicle between the sending of its request, and the delivery of this request, it will cancel its crossing. If a vehicle receives a request to cross from another vehicle after the delivery of its own request, it will ignore this other request. As any vehicle whose request to cross has not been delivered in CC will not cross, and that the delivery order of all messages is the same at all vehicles, this ensures that only a single vehicle crosses at a time.

c) Cancellation of crossing: When the request to cross of a vehicle has not been delivered in CC, or the vehicle has received a request to cross before the delivery time of its own request to cross, a vehicle should cancel its crossing. It might send a message to this effect to other vehicles in the critical coverage, but as it might not be able to communicate over all of the critical coverage, vehicles should not rely on it. Such messages can be used as an optimisation, so that the vehicles who receive them are informed that they can initiate a request to cross, without having to wait for the time the other vehicle would have finished its crossing.

d) Summary: The behaviour of a vehicle intending to cross is summarised on Figure 4. In a queue of vehicles, the vehicle in front is the only one that can cross.

C. Evaluation

We have shown how the coordination model can be applied to autonomous vehicles to allow them to coordinate their behaviour around an unsignalised junction. This solution ensures that as long as they can communicate, vehicles will make progress, and that they will stop before crossing the junction should the communication quality fail below a threshold (in terms of coverage for the given latency) which implies that the safety constraints cannot be guaranteed. Meanwhile, vehicles which are still able to communicate will be able to make progress. The solution does not cater for fairness, but this could be included, by changing the algorithm to handle multiple requests.

It must be noted that the assumption that vehicles can sense whether the junction is empty has been adopted for simplicity, but might be removed by imposing on vehicles to listen for messages for a certain time before sending any. This would require either a bigger critical coverage, or that vehicles stop or slow down before crossing the junction. In this case, however, vehicles have to be able to cross the junction within a bounded time.

The coordination model has been applied to other scenarios from the Intelligent Transportation Systems domain, including early ambulance arrival warning and a pedestrian traffic light for autonomous vehicles [13].

VI. Conclusion

In this paper, we complement existing research on autonomous vehicles by presenting a real-time coordination model for autonomous mobile entities. This coordination model is built over a real-time communication model for wireless networks, in which entities are informed about the area in which the messages they have sent have been delivered. Depending on the size of the coverage, vehicles can adapt their behaviour, for example by waiting before crossing an unsignalised junction. We have shown that this coordination model can be used to derive constraints on the behaviour of autonomous vehicles around an unsignalised junction to ensure that they will cross the junction safely. Our future work include developing a set of tools to help the automatic derivation of such constraints.

Acknowledgement

The authors are grateful to Science Foundation Ireland for their support of the work described in this paper under Investigator award 02/IN1/I250 between 2003 and 2007.

References

[1] M. Parent, "From driver assistance to full automation for improved efficiency and better safety," in *Proceedings of the Vehicular Technology Conference (VTC 2004-Spring)*, vol. 5. IEEE Computer Society, May 2004, pp. 2931–2934.

[2] M. Parents and G. Gallais, "Intelligent transportation in cities with cts," in *Proceedings of the IEEE Conference on Intelligent Transportation Systems (ITSC'02)*. IEEE Computer Society, Sept. 2002, pp. 826–830.

[3] J. Baber, J. Kolodko, T. Noel, M. Parent, and L. Vlacic, "Cooperative autonomous driving: Intelligent vehicles sharing city roads," *IEEE Robotics & Automation Magazine*, vol. 12, no. 1, pp. 44–49, Mar. 2005.

[4] S. Kato, S. Tsugawa, K. Tokuda, T. Matsui, and H. Fujii, "Vehicle control algorithms for cooperative driving with automated vehicles and intervehicle communications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, no. 3, pp. 155–161, Sept. 2002.

[5] G. Gaertner and V. Cahill, "Understanding link quality in 802.11 mobile ad hoc networks," *IEEE Internet Computing*, vol. 8, no. 1, pp. 55–60, Jan./Feb. 2004.

[6] R. Aufrere, J. Gowdy, C. Mertz, C. Thorpe, C.-C. Wang, and T. Yata, "Perception for collision avoidance and autonomous driving," *Mechatronics*, vol. 13, no. 10, pp. 1149–1161, Dec. 2003.

[7] J. Kolodko and L. Vlacic, "Cooperative autonomous driving at the intelligent control systems laboratory," *IEEE Intelligent Systems*, vol. 18, no. 4, pp. 8–11, July/Aug. 2003.

[8] E. O'Gorman, "Using group communication to support inter-vehicle coordination," Master's thesis, Dept. of Computer Science, Trinity College Dublin, Sept. 2002.

[9] M. Mock, *On the Real-Time Cooperation of Autonomous Systems*, ser. Fraunhofer Series in Information and Communicatoin Technology. Aachen: Shaker Verlag, June 2004.

[10] E. Nett, M. Gergeleit, and M. Mock, "Mechanisms for a reliable cooperation of vehicles," in *Proceedings of the IEEE International Symposium on High-Assurance Systems Engineering (HASE'01)*. IEEE Computer Society, Oct. 2001, pp. 75–81.

[11] S. Schemmer, E. Nett, and M. Mock, "Reliable real-time cooperation of mobile autonomous systems," in *Proceedings of the Symposium on Reliable Distributed Systems (SRDS 2001)*. IEEE Computer Society, Oct. 2001, pp. 238–246.

[12] A. L. Murphy, G. P. Picco, and G.-C. Roman, "Lime: A middleware for physical and logical mobility," in *Proceedings of the International Conference on Distributed Computing Systems (ICDCS'01)*. IEEE Computer Society, Apr. 2001, pp. 524–533.

[13] M. Bouroche, B. Hughes, and V. Cahill, "Building reliable mobile applications with space-elastic adaptation," in *Proceedings of the Mobile Distributed Computing workshop (MDC 2006)*. IEEE Computer Society, June 2006, pp. 627–632.

[14] H. Hartenstein, B. Bochow, A. Ebner, M. Lott, M. Radimirsch, and D. Vollmer, "Position-aware ad hoc wireless networks for inter-vehicle communications: the fleetnet project," in *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*. ACM, Oct. 2001, pp. 259–262.

[15] M.-O. Killijian, R. Cunningham, R. Meier, L. Mazare, and V. Cahill, "Towards group communication for mobile participants," in *Proceedings of ACM Workshop on Principles of Mobile Computing (POMC'2001)*, Aug. 2001, pp. 75–82. [Online]. Available: http://www.cs.tcd.ie/publications/tech-reports/reports.01/TCD-CS-2001-27.ps