Mobility Models and Behavioural Equivalence for Wireless Networks

Jens Chr. Godskesen[1,⋆] and Sebastian Nanz[2]

[1] IT University of Copenhagen
jcg@itu.dk
[2] Technical University of Denmark
nanz@imm.dtu.dk

Abstract. In protocol development for wireless systems, the choice of appropriate mobility models describing the movement patterns of devices has long been recognised as a crucial factor for the successful evaluation of protocols. More recently, wireless protocols have also come into the focus of formal approaches to the modelling and verification of concurrent systems. While in these approaches mobility is also given a central role, the actual mobility modelling remains simplistic since arbitrary node movements are allowed. This leads to a huge behavioural overapproximation that might prevent a successful reasoning about protocol properties. In this paper we describe how to extend a process calculus by realistic mobility models in an orthogonal way. The semantics of our calculus incorporates a notion of global time passing that allows us to express a wide range of mobility models currently used in protocol development practice. Using the behavioural equivalence and pre-order of our calculus, we are furthermore able to compare the strength of these models in our approach.

1 Introduction

As a result of the availability and popularity of mobile devices with networking capabilities, the use of wireless communication has seen a tremendous increase in recent years. The applications of this technology are broad and include wireless local area networks, cellular and ad-hoc networks, and have a further growth potential in the area of ubiquitous computing.

Naturally, the interest in modelling and formal reasoning about wireless networks has risen as well, for example using process algebra as a specification formalism. Process algebra itself has proved to be a versatile formalism for modelling various kinds of concurrent systems. This versatility is needed to model wireless networks, since there are a number of key differences to other network behaviour typically modelled in process calculi. A number of works [10,7,4,11,15,8] have stressed two of the main differences: the prevalent mode of communication

---

[⋆] Supported by grant no. 272-05-0258 from the Danish Natural Science Research Council and by the VKR Centre of Excellence MT-LAB.

in wireless networks is broadcast, and the network topology can change spontaneously. The latter point implies that there is a strict separation between process actions and the mobility modelling, since changes in connectivity are influenced by environment conditions (such as node movement), but not by the actions of a protocol process.

In the works mentioned above, a simplistic view of mobility is taken by assuming that connectivity develops completely arbitrarily over time. Hence, erratic behaviour with respect to connectivity breaking and establishing is part of the model. This contrasts with the approach taken by protocol developers, where more realistic mobility models (the survey paper [2] provides a fairly comprehensive overview) are seen as a key ingredient for producing meaningful protocol evaluations using simulation. Realistic mobility models should however play an important role for formal reasoning as well because, by excluding all erratic behaviour, they can limit the size of the state space to be reasoned about, which is an important prerequisite for verification. For the same reason, stronger properties should be provable for the system.

In this paper, we provide a general model of mobility to parametrise a simple calculus with broadcast capabilities. The calculus is equipped with a notion of global time passing, such that movement trajectories of nodes can be determined explicitly via a mobility model. The general model is shown to instantiate to widely used concrete mobility models for network simulation. We develop a behavioural equivalence and pre-order that allow us to compare the strength of these mobility models in our formal setting.

The remainder of the paper is structured as follows. In Section 2 we give an introduction to a number of mobility models typically used in network simulators, and describe our general model of mobility and broadcast. We evaluate related work on modelling wireless mobility and time passing in Section 3. The syntax and semantics of our calculus are presented in Section 4. In Section 5 we develop a behavioural equivalence and apply it in order to evaluate the strength of the mobility models we have instantiated our calculus to. We give an outlook on the development of a discrete version of the semantics in Section 6 and conclude in Section 7.

2 Mobility Models for Wireless Networks

When evaluating protocols for wireless networks with respect to performance or functional correctness, a variety of assumptions has to be decided upon. Such assumptions may for example include the size and shape of the area used by the wireless devices, their transmission ranges, and their movement patterns including allowed speeds and directional changes [2]. In every case, careful choices have to be made by the analyst to ensure that the evaluation results apply in practice.

As a specific protocol may be targeted to various environment conditions, different choices of assumptions may be in order, and therefore it is important that simulation tools (such as the network simulator ns-2 [12]) or indeed veri-

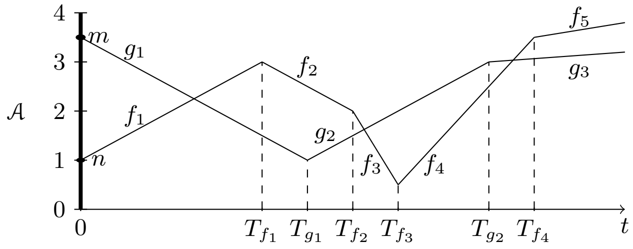

Fig. 1. Node movements described by mobility functions with timeouts

fication tools are parametric in these assumptions. In the following we develop such a parametric framework for use with a process algebraic approach.

We thus assume that wireless devices ("nodes") move in a *global area* $\mathcal{A}$ that is bounded and convex (this ensures that the definitions of the most commonly used mobility models are well-defined). The *boundary* of $\mathcal{A}$ is denoted by $bd(\mathcal{A})$. We assume to have a *global clock* $t \in \mathbb{R}_0^+$, such that the location $\vec{l} \in \mathcal{A}$ of a node n at time t can be determined by its associated *mobility function* $f : \mathbb{R}_0^+ \to \mathcal{A}$ as $f(t) = \vec{l}$.

We furthermore assume that each mobility function has a *timeout* $T \in \mathbb{R}_0^+ \cup \{\infty, \diamond\}$. If $T \in \mathbb{R}_0^+$, this means that f describes the trajectory of n for $t \leq T$, and has to be replaced by a new mobility function f' at time $t = T$. If $T = \infty$ the timeout never occurs and the current mobility function is always valid. We use the special symbol $\diamond$ to express that no timeout is set, i.e. the mobility function may be replaced at any time. We write a node n with mobility function f and timeout T as n_f^T.

The choice of a new mobility function is determined by a *mobility model* $\mathcal{M}$, a function that takes a pair $(\vec{l_0}, t_0)$ of a current location and a current time as input and returns, for these parameters, a set of pairs (f', T') of admissible mobility functions and their timeouts. For example, at time $t_0 = T$, node n_f^T may become $n_{f'}^{T'}$ where $(f', T') \in \mathcal{M}(f(t_0), t_0)$. It is important to note at this point that this *allows* that two nodes who happen to be at the same location at same time may take different movement trajectories: the set $\mathcal{M}(f(t_0), t_0)$ has more than one element in general, and elements (f', T') are nondeterministically chosen for each node.

Example 1. Consider the illustration in Figure 1, which describes the movement of two nodes m and n within the one-dimensional area $\mathcal{A} = [0, 4]$. We use a mobility model **Mob** defined as follows:

$$\mathbf{Mob}(\vec{l_0}, t_0) = \{(g, T) \mid \begin{array}{l} g(t) = \vec{l_0} + v \cdot (t - t_0) \cdot \alpha \wedge T \in [t_0, t_0 + 2], \text{ where} \\ v \in \ [0, 2],\ \alpha \in \{1, -1\},\ g(t) \neq bd(\mathcal{A}) \text{ for } t \in [t_0, T]\} \end{array}$$

In this model, node movement is continuous in the sense that $f(T) = f'(T)$, where T is the timeout of f, and f' replaces f. Furthermore, movement takes

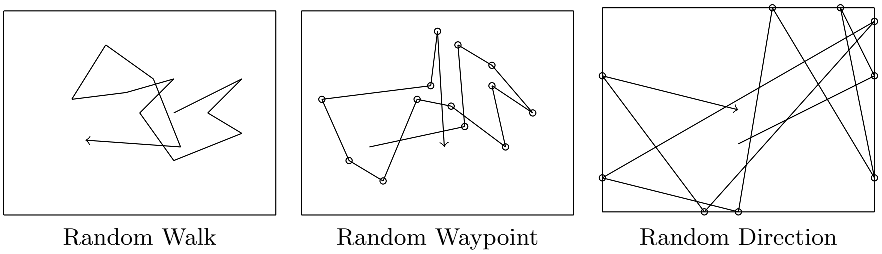
Random Walk Random Waypoint Random Direction

Fig. 2. Movement patterns of a node according to three mobility models

place at constant speed $v \in [0, 2]$ and both in forward and backward direction $\alpha \in \{1, -1\}$, a timeout occurs at least every 2 time units, and nodes never leave the area $\mathcal{A}$.

In Figure 1, node m moves initially according to the mobility function g_1 with timeout T_{g_1}, hence $m = m_{g_1}^{T_{g_1}}$. At time $t = 0$ node $m_{g_1}^{T_{g_1}}$ is at location $g_1(0) = 3.5 \in \mathcal{A}$ and then moves with constant speed to location $g_1(T_{g_1}) = 1$ where its mobility function g_1 is timing out, $t = T_{g_1}$. Because of the timeout, a new mobility function g_2 with new timeout T_{g_2} is chosen from $\mathbf{Mob}(g_1(T_{g_1}), T_{g_1})$ and the node thus becomes $m_{g_2}^{T_{g_2}}$. The further movements of m and those of n can be described similarly.

The proposed framework allows to describe a variety of concrete mobility models which we outline in the following.

Stationary Nodes Nodes are assumed to be stationary.

$$\mathbf{Stat}(\vec{l_0}, t_0) = \{(g, \infty) \mid g(t) = \vec{l_0}\}$$

Arbitrary Movement Nodes can change their location arbitrarily and instantaneously.

$$\mathbf{Arb}(\vec{l_0}, t_0) = \{(g, \diamond) \mid \ g(t) = \vec{l} \text{ where } \vec{l} \in \mathcal{A}\}$$

Random Walk Nodes choose randomly a speed $v \in \mathit{Vel}$ and a direction $\vec{\alpha} \in \mathit{Dir}$ and travel in direction $\vec{\alpha}$ at the chosen speed for a fixed time interval Δ.

$$\begin{aligned}\mathbf{RWalk}_{\Delta}(\vec{l_0}, t_0) = \{(g, t_0 + \Delta) \mid \ & g(t) = \vec{l_0} + v \cdot (t - t_0) \cdot \vec{\alpha} \text{ where } v \in \mathit{Vel}, \\ & \vec{\alpha} \in \mathit{Dir},\ g(t) \neq bd(\mathcal{A}) \text{ for } t \in [t_0, t_0 + \Delta]\}\end{aligned}$$

(Note that in the classical model a node's trajectory is "reflected" according to $\vec{\alpha}$ when reaching the boundary of $\mathcal{A}$. While this can be expressed in our model, we modify the definition for simplicity to say that a trajectory that would reach the boundary is never chosen.)

Random Waypoint Nodes choose randomly a speed $v \in \mathit{Vel}$ and a destination $\vec{d} \in \mathcal{A}$ and travel to $\vec{d}$ at the chosen speed, where they pause for a fixed time interval p. In the following definition, t_1 is the time when the destination will be reached and $\vec{\alpha}$ is the direction in which to travel (both easily calculated from $\vec{l_0}, \vec{d}, v$).

$$\mathbf{RWay}_p(\vec{l_0}, t_0) = \{(g, t_1 + p) \mid \ g(t) = \begin{cases} \vec{l_0} + v \cdot (t - t_0) \cdot \vec{\alpha} & \text{if } t_0 \leq t \leq t_1 \\ \vec{d} & \text{if } t_1 < t \leq t_1 + p \end{cases} \\ \text{where } v \in \mathit{Vel} \text{ and } \vec{d} \in \mathcal{A}\}$$

Random Direction Nodes choose randomly a speed $v \in \mathit{Vel}$ and a direction $\vec{\alpha} \in \mathit{Dir}$ and travel in direction $\vec{\alpha}$ at the chosen speed until they reach the boundary, where they pause for a fixed time interval p. In the following definition, t_1 is the time when the boundary will be reached and $\vec{d}$ is the point reached on the boundary (both easily calculated from $\vec{l_0}, \vec{\alpha}, v$).

$$\mathbf{RDir}_p(\vec{l_0}, t_0) = \{(g, t_1 + p) \mid \ g(t) = \begin{cases} \vec{l_0} + v \cdot (t - t_0) \cdot \vec{\alpha} & \text{if } t_0 \leq t \leq t_1 \\ \vec{d} & \text{if } t_1 < t \leq t_1 + p \end{cases} \\ \text{where } v \in \mathit{Vel},\ \vec{\alpha} \in \mathit{Dir},\ g(t_1) = \vec{d} \in bd(\mathcal{A})\}$$

The Random Walk, Random Waypoint, and Random Direction models are classical models used to realistically represent the movement of mobile nodes [2], e.g. in network simulation tools. Figure 2 depicts possible movements of a node in a two-dimensional area according to these three models, where for the Random Waypoint and Random Direction models pause times are expressed using small circles on trajectories. The Random Walk and Random Waypoint models also count as the two most commonly [2] used mobility models in research on wireless networks, and all models shown can be classified as so-called entity mobility models (where nodes move completely independent of each other). Group mobility models (where multiple nodes move together) are beyond the scope of this paper.

In contrast to the classical simulation models, the stationary model and the model of arbitrary movement have predominantly been used in formal approaches to the evaluation of wireless networks. In the next section we review these approaches.

3 Mobility and Time in Process Calculi

An important feature of the wireless medium is that all message transmissions are broadcasts, and may thus be received by several nodes. A suitable model for wireless networks must thus also determine which nodes may receive a broadcast message. In our approach, this is straightforwardly modelled by providing a function $\mathit{area}_n(\vec{l})$ that defines the *transmission area* of a node n at location $\vec{l}$. Hence, if a node n_f^T transmits at a time t, then nodes $m_g^{T'}$ with $g(t) \in \mathit{area}_n(f(t))$ may receive the transmission. Note however that transmissions may fail in reality even though nodes are within radio range, e.g. due to hidden terminal effects; this leads us later in Section 4 to an operational semantics that incorporates message loss. In all our examples we assume that area_n describes a circular

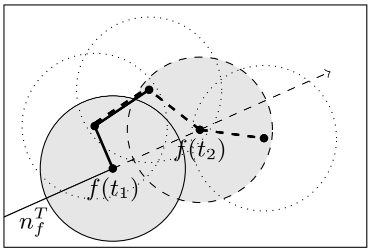

Fig. 3. Changes in the network topology due to node movement

transmission area, but differently shaped areas or even a realistic modelling of radio attenuation may be handled using a similar choice of representation.

The set of all nodes together with the transmission links between senders and receivers as edges then determines a graph, called the *network topology*, which may change over time depending on node movements. The network topology is in general given by a directed graph: unidirectional links can occur if there are different transmission radii (in our examples we assume bidirectional links to simplify presentation). An example for the change of network topology is given in Figure 3, where the moving node n_f^T with circular transmission area first establishes one connection with one of the stationary nodes at time t_1, and then breaks the connection again and has established two new links at time t_2. In the figure, the bold solid lines correspond to the node connectivity at time t_1, and the bold dashed lines to the node connectivity at time t_2.

In the following we describe related approaches to studying mobile wireless behaviour in a process algebraic setting. Process calculi with broadcast behaviour were first studied by Prasad [13] in the *Calculus of Broadcasting Systems (CBS)* and Ene and Muntean [3] in the *bπ-calculus*. These approaches are not directly suitable to describe wireless communication, since in their model all nodes in the network receive a broadcast message (*global broadcast*), whereas, as discussed above, the transmission ranges of nodes naturally introduce a kind of locality for broadcast actions (*local broadcast*). A number of process calculi that take this consideration into account have been proposed recently [10,7,4,11,15,8]. The approaches differ in the way the network topology is specified. Nanz and Hankin [10,9] have introduced *CBS*$^\sharp$ where the topology modelling takes place at the semantic level: every sending step in the semantics is parametrised by a directed graph that determines which nodes are connected at that moment. These connectivity graphs are nondeterministically chosen from a predefined set of graphs, hence modelling spontaneous topological changes. As node movements are otherwise unrestricted, this corresponds in our classification to the model of arbitrary movement. Based on the same principle Nanz, Nielson, and Nielson [11] have defined *bKlaim* which uses a coordination model where nodes communicate by placing message tuples into tuple spaces and retrieving them again; the same mobility model is used as in *CBS*$^\sharp$.

In contrast to [10,11], other approaches attempt connectivity modelling at the syntactic level. The *Calculus for Mobile Ad Hoc Networks (CMAN)* by Godskesen [4] focuses on behavioural equivalences for the purpose of security modelling. The connectivity of nodes is determined by examining the set of neighbouring nodes, associated with each node. Since these sets can change arbitrarily, node movement has no restrictions. In a more recent work [5] CMAN is equipped with a static location binding operator that limits the arbitrary mobility to happen only within the scope of the binder. Singh, Ramakrishnan, and Smolka [15] define the *ω-calculus* and use it for protocol analysis. Their connectivity modelling is based on connected groups in the connectivity graph. Each process is associated with the set of groups it belongs to, and change of group membership may be forced to comply with graph invariants. While this approach is able to restrict arbitrary movement, which we also strive to do, our model emphasizes more the importance of *realistic* transitions between connectivity graphs.

It is interesting to note that the approaches to connectivity modelling described above are just variations on traditional ways to describe ordinary graphs: using node and edge sets as in [10], adjacency lists as in [4], and maximal cliques as in [15]. One could imagine an alternative approach where connectivity and movements are coded into a distinguished process which acts as network medium. Our modelling using the *area* function for computing transmission areas strives instead to be a close model of real networks. This type of modelling can also be understood as a generalisation of the approach of Mezzetti and Sangiorgi [8], who equip each node with a location and a sending radius. Assuming a distance function, possible receivers can thus be determined from this data, together with the receivers' locations. In [8] only stationary nodes are considered, and the calculus aims at modelling MAC layer protocols (medium access control), where wireless interferences play a central role, while the other calculi mentioned here abstract from these physical complications and are suited for modelling Network layer protocols (routing). Merro [7] has used the location/radius approach in the *Calculus of Mobile Ad Hoc Networks (CMN)* with a special focus on defining appropriate behavioural equivalences in the wireless setting (again abstracting from MAC-problems). In terms of mobility modelling, CMN is restricted to the model of arbitrary movement, but introduces a switch to make selected nodes stationary.

The overview shows that current approaches to modelling mobile wireless networks have focused on stationary nodes and nodes with arbitrary movement patterns. While considering arbitrary movement seems compelling for formal verification ("all behaviour is included"), it is also limiting in many respects. For example, in security analysis one might want to establish a robustness property of a network against the influence of a single attacker. The attacker might only be able to influence nodes it can connect to, and as arbitrary movement essentially enables the attacker to be in all locations of the network simultaneously, it may be impossible to establish this property under this model. Also for properties of functional correctness the model of arbitrary movement seems inadequate, for example one would not expect a routing protocol to be able to handle completely different network topologies at every sending step. Our approach addresses this issue by offering the usage of realistic mobility models.

Lastly, we consider time passing in conjunction with wireless behaviour. While timing aspects are well-known to be essential in evaluating wireless protocols, they have not been considered for wireless calculi so far. There is however a variety of classical calculi that have been extended with time, for example TCCS [16], Timed ACP [1], TCSP [14], to mention only a few (and omitting stochastic calculi [6] where timing is determined by random variables). In [16,14], a delay action is considered to let processes idle until they can perform an action, and the maximal progress assumption is used to ensure that τ-actions cannot be delayed. In [1], time is absolute and each action is associated with a time stamp at which it should be performed. In our approach, time is also global as it is used to determine the positions of the nodes, and we consider a minimal calculus around this idea. However, our calculus is not a genuine timed calculus in that the process part does not contain operators dealing with time, only the mobility part is time dependent. Extensions of the calculus that also tag actions with delays may be considered in future work.

4 A Simple Calculus for Wireless Networks

In this section we introduce the syntax and the operational semantics of a calculus for wireless networks. The process part is deliberately chosen to be very simple and more or less standard for wireless process calculi and is not our contribution, but for the network part we add mobility functions and their timeout for individual nodes as a new feature. While we use a minimal operator set to make for a concise presentation, the process part can be extended with conditional expressions and name creation as expected, and we have proved our results for this larger set.

The labelled transition system semantics for our calculus is equipped with a notion of global time such that the mobility of nodes can be determined explicitly via a mobility function that is chosen from a mobility model parametrising the operational semantics.

4.1 Syntax

We assume to have infinite sets of *names* $\mathcal{N}$, *variables* $\mathcal{X}$, *identities* $\mathcal{I}$, as well as *process constants* (ranged over by A). The sets of *processes* and *networks* is defined as:

$$\begin{aligned} u, v \quad &::= n \mid x \quad \text{where } n \in \mathcal{N},\ x \in \mathcal{X} \\ P, Q \quad &::= \mathbf{0} \mid \langle v \rangle.P \mid (x).P \mid A(\tilde{v}) \\ N, M \ &::= a[P]^{T}_{f} \mid M \parallel N \quad \text{where } a \in \mathcal{I} \end{aligned}$$

The terminated process is represented by $\mathbf{0}$. Output of a name v is described by the process $\langle v \rangle.P$, input and binding of the received name to the variable

x by $(x).P$. Note that the output is on a wireless channel, i.e. it has a broadcast semantics. Furthermore we are not distinguishing between multiple wireless channels (the number of these is fixed in real systems) which would be a straightforward extension if needed. We write $A(\tilde{v})$ to denote a process defined via a (possibly recursive) process definition $A(\tilde{x}) \triangleq P$. Processes are assumed to be sequential only.

Networks can be composed in parallel via $M \parallel N$, and consist of nodes of the form $a[P]^T_f$. A node consists of a process P that runs at an identified container a, where the identity a is unique in a network and represents an address of some sort. Furthermore, nodes move in a global area $\mathcal{A}$ according to their *mobility function* f, which yields a location $f(t)$ when applied to a point in time t. The *timeout* T determines until when the mobility function will be valid as spelled out in Section 2.

4.2 Operational Semantics

We equip the calculus with a parametrised operational semantics where transition rules are of the form:

$$\mathcal{M} \vdash (t, M) \xrightarrow{\lambda} (t', N) \quad \text{where} \quad \lambda ::= \tau \mid \varDelta \mid \langle m \rangle @A! \mid (m)@A?$$

i.e. the parameter is $\mathcal{M}$ and configurations are pairs (t, M) of global time t and network M. Transition labels can either refer to change of mobility function (τ) or a time delay ($\varDelta \in \mathbb{R}^+_0$), or it may refer to output of a message in some area ($\langle m \rangle @A!$) or to input of a message ($(m)@A?$) send out in some area. We always assume a configuration (t, M) to be *well-formed*, i.e. for all timeouts $T \in \mathbb{R}^+_0$ in M it must be that $t \leq T$.

The overall intuition of our semantics is that we let time proceed globally for a network and furthermore network execution not only depends on the global time but also on the mobility model $\mathcal{M}$.

We change the movement trajectories of nodes using the mobility rule:

$$\text{(mobility)} \quad \frac{(f', T') \in \mathcal{M}(f(t), t)}{\mathcal{M} \vdash (t, a[P]^T_f) \xrightarrow{\tau} (t, a[P]^{T'}_{f'})} \quad \text{if } T = \diamond \text{ or } T = t$$

The rule of time passing tells when a network may let time progress:

$$\text{(time)} \quad \mathcal{M} \vdash (t, N) \xrightarrow{\varDelta} (t + \varDelta, N) \quad \text{if } t + \varDelta \leq T \text{ for all timeouts } T \in \mathbb{R}^+_0 \text{ in } N$$

Sending and reception is described by the following rules

$$\text{(send)} \qquad \mathcal{M} \vdash (t, a[\langle m \rangle.P]^T_f) \xrightarrow{\langle m \rangle @A!} (t, a[P]^T_f) \qquad \mathit{area}_a(f(t)) = A$$

$$\text{(receive}_1\text{)} \quad \mathcal{M} \vdash (t, a[(x).P]^T_f) \xrightarrow{(m)@A?} (t, a[P\{m/x\}]^T_f) \quad f(t) \in A$$

$$\text{(receive}_2\text{)} \qquad \mathcal{M} \vdash (t, a[\mathbf{0}]^T_f) \xrightarrow{(m)@A?} (t, a[\mathbf{0}]^T_f) \qquad f(t) \in A$$

where the operator $area_a(\vec{l})$ computes the sending area of node a at location $\vec{l}$ as described in Section 3. Note that we assume that the time for transmission of messages is negligible compared to the time it takes for a node to move (for example, transmitting 100 kB with 11 Mbit/s bandwidth takes less than 0.1 s, in which a node moving at 50 km/h will move about 1 meter); therefore transmissions have no duration in our semantics.

Synchronisation of sending actions is done via the following two rules.

$$\text{(synch}_1\text{)}\quad \frac{\mathcal{M} \vdash (t, N) \xrightarrow{\langle m \rangle @A!} (t, N') \quad \mathcal{M} \vdash (t, M) \xrightarrow{(m)@A?} (t, M')}{\mathcal{M} \vdash (t, N \parallel M) \xrightarrow{\langle m \rangle @A!} (t, N' \parallel M')}$$

$$\text{(synch}_2\text{)}\quad \frac{\mathcal{M} \vdash (t, N) \xrightarrow{(m)@A?} (t, N') \quad \mathcal{M} \vdash (t, M) \xrightarrow{(m)@A?} (t, M')}{\mathcal{M} \vdash (t, N \parallel M) \xrightarrow{(m)@A?} (t, N' \parallel M')}$$

The rule (synch$_1$) and (synch$_2$) are similar to the rules in [13] for dealing with broadcast: a broadcast message continues to be distributed, and several nodes may agree on simultaneous reception of a message. Here we note that there are two symmetric rules to the two before, where the order of the parallel networks is switched (we do not have a structural congruence).

It is important to note that we do *not* intend that local broadcasts are actually received by *all* nodes that are located in the sending area. This is because in reality, as mentioned earlier in Section 3, message loss may occur even though nodes are within radio range. The following rule for (par) expresses this situation by allowing to be concerned with only a subset of the nodes in the network. We assume again that there is a symmetric rule available.

$$\text{(par)}\quad \frac{\mathcal{M} \vdash (t, N) \xrightarrow{\lambda} (t, N')}{\mathcal{M} \vdash (t, N \parallel M) \xrightarrow{\lambda} (t, N' \parallel M)} \quad \lambda \neq \varDelta$$

The rule for recursion is standard.

$$\text{(rec)}\quad \frac{\mathcal{M} \vdash (t, a[P\{\tilde{v}/\tilde{x}\}]^T_f) \xrightarrow{\lambda} (t, a[P']^{T'}_{f'})}{\mathcal{M} \vdash (t, a[A(\tilde{v})]^T_f) \xrightarrow{\lambda} (t, a[P']^{T'}_{f'})} \quad A(\tilde{x}) \triangleq P$$

4.3 Semantic Characteristics

As pointed out in Section 3, the labelled transition semantics of our calculus differs much from the semantics of classical timed calculi: process actions are not time dependent, only the mobility part is. In the following we describe some of the semantic characteristics of change of mobility function and progress of global time, which can however be related to concepts known from timed calculi.

Like many timed calculi our calculus also gives priority to τ-actions over time delays. However, since only the mobility rule gives rise to τ-actions, this does *not* force transmitters to send instantaneously (as the "maximal progress" assumption of some timed calculi does). Instead, this condition translates to our setting in that we may only let time progress if no mobility function has timed out:

Proposition 1. *If* $\mathcal{M} \vdash (t, M) \xrightarrow{\tau} (t, M')$ *for some* M' *then* $\mathcal{M} \vdash (t, M) \overset{\Delta}{\not\longrightarrow}$.

Another important timed property possessed by timed calculi is *time determinism*, meaning that when time progresses the same configuration is always reached. Actually, since we have no explicit delay operators in our calculus it holds that a network does not change (syntactically) after a delay.

Proposition 2 (Time Determinism). *If* $\mathcal{M} \vdash (t, M) \xrightarrow{\Delta} (t', M')$ *and* $\mathcal{M} \vdash (t, M) \xrightarrow{\Delta} (t'', M'')$ *then* $t' = t'' = t + \Delta$ *and* $M = M' = M''$.

As corollaries we obtain that all nodes in a network must agree to let time progress (and hence all nodes must move on according to their mobility function)

Corollary 1 (Time Synchronisation). $\mathcal{M} \vdash (t, M \parallel N) \xrightarrow{\Delta} (t', M \parallel N)$ *iff* $\mathcal{M} \vdash (t, M) \xrightarrow{\Delta} (t', M)$ *and* $\mathcal{M} \vdash (t, N) \xrightarrow{\Delta} (t', N)$.

and that any delay can be divided into sub-delays:

Corollary 2 (Time Additivity). *If* $\mathcal{M} \vdash (t, M) \xrightarrow{\Delta+\Delta'} (t + \Delta + \Delta', M)$ *then* $\mathcal{M} \vdash (t, M) \xrightarrow{\Delta} (t + \Delta, M)$ *and* $\mathcal{M} \vdash (t + \Delta, M) \xrightarrow{\Delta'} (t + \Delta + \Delta', M)$

Observe that our calculus as a natural consequence does not support the property of *time persistency* i.e.

$$\text{if } \mathcal{M} \vdash (t, M) \xrightarrow{\lambda} \text{ and } \mathcal{M} \vdash (t, M) \xrightarrow{\Delta} \text{ then } \mathcal{M} \vdash (t + \Delta, M) \xrightarrow{\lambda}$$

simply because it would be unreasonable to expect due to the interplay between time and mobility that a node after some time progress will continue broadcasting in the same area.

5 A Framework for Comparing Mobility Models

In order to be able to compare the strengths of the various mobility models we provide standard behavioural pre-order and equivalences. Since we do not want to take the actual shift of mobility functions as an observable into account, these turn out to be weak simulations and bisimulations in our case.

5.1 Behavioural Pre-order and Equivalence

We consider relations $\mathcal{R}$ containing pairs $((\mathcal{M}, t, M), (\mathcal{M}', t, M'))$ consisting of two networks with identical timing information but where the networks may choose mobility functions from (in principle) different mobility models; $\mathcal{R}$ is said to be *well-formed* whenever for all such pairs (t, M) and (t, M') are well-formed.

Moreover, for ease of notation we write $(\mathcal{M}, M)\,\mathcal{R}_t(\mathcal{M}', M')$ if $(\mathcal{M}, t, M)\,\mathcal{R}\,(\mathcal{M}', t, M')$ and $M\,\mathcal{R}_t^{\mathcal{M}}M'$ if $(\mathcal{M}, M)\,\mathcal{R}_t(\mathcal{M}, M')$.

Next we define weak simulation and bisimulation. In order to define weak labelled transitions, for each label λ, we shall write $P \overset{\lambda}{\Rightarrow} Q$ iff either $\lambda \neq \tau$ and there exist P' and Q' such that $P(\xrightarrow{\tau})^*P' \xrightarrow{\lambda} Q'(\xrightarrow{\tau})^*Q$ or $\lambda = \tau$ and $P(\xrightarrow{\tau})^*Q$.

Definition 1. *A well-formed relation $\mathcal{R}$ is a* weak simulation *if $(\mathcal{M}, N)\,\mathcal{R}_t(\mathcal{M}', M)$ implies*

1. $\mathcal{M} \vdash (t, N) \xrightarrow{\langle m\rangle @A!} (t, N')$ *implies* $\mathcal{M}' \vdash (t, M) \xRightarrow{\langle m\rangle @A'!} (t, M')$ *for some M' and A' such that $A \subseteq A'$ and $(\mathcal{M}, N')\,\mathcal{R}_t(\mathcal{M}', M')$.*
2. $\mathcal{M} \vdash (t, N) \xrightarrow{(m)@A?} (t, N')$ *implies* $\mathcal{M}' \vdash (t, M) \xRightarrow{(m)@A'?} (t, M')$ *for some M' and A' such that $A' \subseteq A$ such that $(\mathcal{M}, N')\,\mathcal{R}_t(\mathcal{M}', M')$.*
3. $\mathcal{M} \vdash (t, N) \xrightarrow{\tau} (t, N')$ *implies* $\mathcal{M}' \vdash (t, M) \overset{\tau}{\Rightarrow} (t, M')$ *for some N' such that $(\mathcal{M}, N')\,\mathcal{R}_t(\mathcal{M}', M')$.*
4. $\mathcal{M} \vdash (t, M) \xrightarrow{\Delta} (t', M)$ *implies* $\mathcal{M}' \vdash (t, N) \overset{\Delta}{\Rightarrow} (t', N)$ *and* $(\mathcal{M}, N)\,\mathcal{R}_{t'}(\mathcal{M}', M)$.

$\mathcal{R}$ is a weak bisimulation if $\mathcal{R}$ and $\mathcal{R}^{-1}$ are weak simulations.

We let $\sqsubseteq$ denote the largest weak simulation and $\approx$ the largest weak bisimulation. It is immediate that $\approx$ is an equivalence relation.

Notice the asymmetry in our definition of weak simulation. A network where a node may broadcast a message m within some area A is simulated by a network at least as powerful that broadcasts m in an area containing A. Dually, a network where nodes receive within some area A must be simulated by a network where nodes are capable of receiving in an area A' contained in A. Notice that a network receiving in some area can always receive in a larger area but not the other way around. The two last clauses in Definition 1 are standard.

It turns out that our bisimulations are closed by parallel composition of networks that are well-formed with respect to the current time:

Theorem 1. *$N \approx_t^{\mathcal{M}} N'$ implies $N \parallel M \approx_t^{\mathcal{M}} N' \parallel M$ for all (t, M).*

As we would expect our behavioural theory satisfies the following commutativity and associativity laws:

Proposition 3. *$M \parallel N \approx_t^{\mathcal{M}} N \parallel M$*

Proposition 4. *$(M \parallel M') \parallel M'' \approx_t^{\mathcal{M}} M \parallel (M' \parallel M'')$.*

Comparing a network across mobility models we can infer that

Proposition 5. *$(\mathcal{M}, M) \sqsubseteq_t (\mathcal{M} \cup \mathcal{M}', M) \sqsubseteq_t (\mathbf{Arb}, M)$*

because whatever M can do choosing from a mobility model $\mathcal{M}$ clearly it can do choosing from a larger set of mobility functions (and from the most powerful set of mobility functions **Arb**), but not necessarily the other way around.

When comparing networks within the same mobility model, we can infer that bisimulation equivalence gets stronger if more mobility functions are available to choose from:

Proposition 6. $\approx_t^{\mathbf{Arb}} \subseteq \approx_t^{\mathcal{M}\cup\mathcal{M}'} \subseteq \approx_t^{\mathcal{M}}$

This means, if $M \approx_t^{\mathcal{M}\cup\mathcal{M}'} N$ then also $M \approx_t^{\mathcal{M}} N$ because in the former case all possible behaviours for M can be matched by N (and vice versa) when choosing mobility functions from both $\mathcal{M}$ and $\mathcal{M}'$, but then clearly with a limited choice of mobility functions M and N are still equivalent. A similar argument holds when comparing $\approx_t^{\mathbf{Arb}}$ and $\approx_t^{\mathcal{M}\cup\mathcal{M}'}$ because whatever a mobility function from $\mathcal{M}\cup\mathcal{M}'$ causes can be matched by mobility functions in **Arb**.

Finally, we observe that if a pair of networks with all timeouts being infinite belongs to $\approx_t^{\mathcal{M}}$ then it also belongs to $\approx_t^{\mathcal{M}'}$ for any $\mathcal{M}'$ because the possible choices of new mobility functions are irrelevant.

5.2 Comparing Mobility Models

Our main motivation for introducing the weak behavioural simulation and equivalence is to be able to compare mobility models, and to demonstrate formally that the choice of mobility function matters for reasoning. The latter point can be shown with the following simple example, where two networks are distinguished under the Random Walk model, but found equivalent using arbitrary movement.

Example 2. Consider the three mobility functions f, g_1, and g_2, and assume that $g_1(t_0) \in \mathit{area}_a(f(t_0))$ and $g_2(t_0) \notin \mathit{area}_a(f(t_0))$. Furthermore, let networks N_1 and N_2 be defined as follows:

$$N_i = a[A]_f^T \parallel b[B]_{g_i}^{\diamond} \qquad \text{where } A \triangleq \langle m\rangle.(x).A \text{ and } B \triangleq (x).\langle x\rangle.B$$

Then the following results hold:

$$N_1 \not\approx_{t_0}^{\mathbf{RWalk}_\Delta} N_2 \quad \text{ and } \quad N_1 \approx_{t_0}^{\mathbf{Arb}} N_2$$

Proof. To see that the inequation holds, we have for network N_1 that

$$\mathbf{RWalk}_\Delta \vdash (t_0, N_1) \xrightarrow{\langle m\rangle @A!} (t_0, a[(x).A]_f^T \parallel b[\langle m\rangle.B]_{g_1}^{\diamond}) \xrightarrow{\langle m\rangle @A'!} (t_0, N_1),$$

however for network N_2, since b is not in range of a,

$$\mathbf{RWalk}_\Delta \vdash (t_0, N_2) \xRightarrow{\langle m\rangle @A!} (t_0, a[(x).A]_f^T \parallel b[(x).\langle x\rangle.B]_{g_2}^{\diamond}) \not\xRightarrow{\langle m\rangle @A'!} .$$

In particular τ-transitions using (mobility) cannot change this situation, because in the model $\mathbf{RWalk}_\Delta$ an amount of time $\delta > 0$ would have to pass to allow for node b to move into the range of a.

To see that the equation holds, note that the two networks differ only in the initial mobility function of node b. Furthermore the model **Arb** allows every node to move instantaneously to any location $\vec{l}$:

$$\mathbf{Arb} \vdash (t_0, b[B]_{g_2}^{\diamond}) \xrightarrow{\tau} (t_0, b[B]_{\lambda x.\vec{l}}^{\diamond})$$

Thus it is easy to establish $b[B]_{g_1}^{\diamond} \approx_t^{\mathbf{Arb}} b[B]_{g_2}^{\diamond}$ and therefore the equation holds by Theorem 1.

The next proposition is about comparing the mobility models of Section 2. As expected, the model of arbitrary movement can simulate all other models, Random Waypoint can simulate Random Direction, and all other pairs of models are incomparable.

Proposition 7. *Assume that all nodes in the network N initially have timeout t_0. Then the following results hold for $\Delta > 0$ and $p > 0$:*

1. $(\mathcal{M}, N) \sqsubseteq_t (\mathbf{Arb}, N)$ *for all* $\mathcal{M} \in \{\mathbf{Stat}, \mathbf{Arb}, \mathbf{RWalk}_\Delta, \mathbf{RWay}_p, \mathbf{RDir}_p\}$
2. $(\mathbf{RDir}_p, N) \sqsubseteq_t (\mathbf{RWay}_p, N)$
3. *All other model pairs in* $\{\mathbf{Stat}, \mathbf{Arb}, \mathbf{RWalk}_\Delta, \mathbf{RWay}_p, \mathbf{RDir}_p\}$ *are incomparable with respect to weak simulation.*

Proof (sketch). The first part of the proposition follows directly from Proposition 5. Conversely, **Arb** cannot be simulated by any other model, because it is the only model where movement does not take time. $\mathbf{RWay}_p$ can simulate $\mathbf{RDir}_p$ as it can choose the point on the boundary of $\mathcal{A}$ that a node under $\mathbf{RDir}_p$ would reach as a waypoint, and has pause times. **Stat** cannot compare with any model model except **Arb**, because it can always let time pass without movement. $\mathbf{RWalk}_\Delta$ cannot compare with either $\mathbf{RWay}_p$ or $\mathbf{RDir}_p$ because it has no notion of pause times. The remaining incomparability results can be established similarly.

6 Discretising Time Delay

The operational semantics defined in Section 4.2 is infinite state because of its real-time delay transitions, and thus there will be no hope for the immediate algorithm underlying our weak bisimulation behavioural equivalence (or weak simulation pre-order) to be decidable.

In this section we hint at future work where a finite state operational semantics and a decidable behavioural equivalence that is a sound and complete characterisation of weak bisimulation should be defined. Clearly such a decidability result relies on the process language of our calculus, but as our exposition in this paper is focused on the semantics of mobility and not on the process part we leave the discussion of an appropriate process fragment for future work.

A Discrete Time Semantics. One may observe that a time delay Δ may be safely carried out without side effects on the capability of a network configuration (t, M) if the following holds:

- No mobility function in M must *timeout* in the interval $[t, t + \Delta)$, and
- No potential receiver of a broadcast message m must *leave* the broadcasting area of the node in M broadcasting m.

To formalise these requirements let (t, M) be a well-formed network configuration, let a_i with $i \in I$ be all identifiers in M and let f_i and T_i be the mobility function and its timeout for a_i.

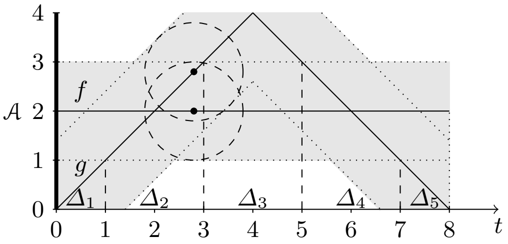

Fig. 4. Node movements and time discretisation in Example 3

Then let $\Delta^t_{timeout}$ be the maximal delay Δ such that $t + \Delta \leq T_i$ for all $i \in I$ with $T_i \neq \diamond$, or if $T_i = \diamond$ for all i then $\Delta^t_{timeout} = \infty$.

Next, for all node identities a_i we need to keep track of all node identities that at time t lie within the transmission range of a_i, i.e. let

$$I^t_{a_i} = \{j \in I \mid f_j(t) \in area_{a_i}(f_i(t))\}$$

Then let Δ^t_{leave} be the smallest delay Δ such that $I^t_{a_i} \neq I^{t+\Delta}_{a_i}$, or if $I^t_{a_i} = I^{t+\Delta}_{a_i}$ for all i then $\Delta^t_{leave} = 0$.

We may now define a new time rule based on the definitions above such that:

$$\text{(time)} \quad \mathcal{M} \vdash (t, N) \xrightarrow{\Delta} (t + \Delta, N) \quad \text{if} \quad \Delta = min\{\Delta^t_{timeout}, \Delta^t_{leave}\}$$

Observe that if always $\Delta = 0$ when applying the rule above then time can be completely abstracted away in the semantics.

Example 3. To illustrate our ideas, assume for simplicity the same one-dimensional area, $\mathcal{A} = [0, 4]$, as in Example 1, and consider a network M consisting of two nodes with identities a and b respectively.

Also, suppose a mobility model $\mathcal{M}$ with just two mobility functions f and g defined by $f(t) = 2$ and by $g(t) = t$ if $t \leq 4$, and $g(t) = 8 - t$ otherwise (see Figure 4). Suppose that the timeout for both f and g is 8.

If the two nodes initially possess the very same mobility function the network may from its initial state $(0, M)$ delay 8 time units, i.e. until the mobility functions time out.

Suppose instead that the two nodes initially contain different mobility functions, and assume for simplicity that both a and b have the same transmission radius 1. In that setting we obtain that the Δ in rule (time) defined above must be chosen according to the following sequence: $1, 2, 2, 2, 1$. That is, first delay is one time unit, then follow three delays of two time units, and finally a delay of one time unit. This situation is depicted in Figure 4.

In future research we want to identify under what restrictions the sampling points for time delays may always be precompiled from a mobility model and the broadcast area of any node. Furthermore, we aim to show that our discretised model may be preferable to models currently used as input to automatic verification tools: as our more realistic handling of connectivity change considers *fewer* possible connections, it may thus set more effective limits on the size of the state space to be considered.

7 Conclusion

In this paper we have described how to extend a simple process calculus with realistic mobility models. The semantics of our calculus incorporates a notion of global time passing that allows us to express a wide range of mobility models currently used in protocol development practice. Using the behavioural equivalence and pre-order of our calculus, we have been able to compare the strength of these models. Finally, we have briefly touched upon the issue of making the real-time semantics symbolic and finite, thereby giving hope for verification of wireless protocols modelled in our approach. In addition, our approach is a step towards bridging the gap between protocol development efforts and formal verification, since it allows to use identical mobility models for both simulation and formal reasoning.

Acknowledgements. We would like to thank the anonymous reviewers for their valuable comments on an earlier version of the paper.

References

1. Jos C. M. Baeten and Jan A. Bergstra. Real time process algebra. *Formal Aspects of Computing*, 3(2):142–188, 1991.
2. Tracy Camp, Jeff Boleng, and Vanessa Davies. A survey of mobility models for ad hoc network research. *Wireless Communications and Mobile Computing*, 2(5):483–502, 2002.
3. Christian Ene and Traian Muntean. A broadcast-based calculus for communicating systems. In *Proc. FMPPTA'03*, 2001.
4. Jens Chr. Godskesen. A calculus for mobile ad hoc networks. In *Proc. COORDINATION'07*, volume 4467 of *LNCS*, pages 132–150. Springer, 2007.
5. Jens Chr. Godskesen. A calculus for mobile ad hoc networks with static location binding. In *Proc. EXPRESS'08*, 2008.
6. Holger Hermanns, Ulrich Herzog, and Joost-Pieter Katoen. Process algebra for performance evaluation. *Theoretical Computer Science*, 274(1-2):43–87, 2002.
7. Massimo Merro. An observational theory for mobile ad hoc networks. In *Proc. MFPS'07*, volume 173 of *ENTCS*, pages 275–293, 2007.
8. Nicola Mezzetti and Davide Sangiorgi. Towards a calculus for wireless systems. In *Proc. MFPS'06*, volume 158 of *ENTCS*, pages 331–353, 2006.
9. Sebastian Nanz. *Specification and Security Analysis of Mobile Ad-Hoc Networks.* PhD thesis, Imperial College London, 2006.
10. Sebastian Nanz and Chris Hankin. A framework for security analysis of mobile wireless networks. *Theoretical Computer Science*, 367(1-2):203–227, 2006.
11. Sebastian Nanz, Flemming Nielson, and Hanne Riis Nielson. Topology-dependent abstractions of broadcast networks. In *Proc. CONCUR'07*, volume 4703 of *LNCS*, pages 226–240. Springer, 2007.

12. The Network Simulator ns-2. `http://www.isi.edu/nsnam/ns/`.
13. K. V. S. Prasad. A calculus of broadcasting systems. *Science of Computer Programming*, 25(2-3):285–327, 1995.
14. George M. Reed and A. W. Roscoe. A timed model for communicating sequential processes. *Theoretical Computer Science*, 58:249–261, 1988.
15. Anu Singh, C. R. Ramakrishnan, and Scott A. Smolka. A process calculus for mobile ad hoc networks. In *Proc. COORDINATION'08*, volume 5052 of *LNCS*, pages 296–314. Springer, 2008.
16. Wang Yi. Real-time behaviour of asynchronous agents. In *Proc. CONCUR'90*, volume 458 of *LNCS*, pages 502–520. Springer, 1990.