Simulation and Modeling of Wireless, Mobile and Ad Hoc Networks

Azzedine Boukerche[*]
Department of Computer Sciences
University of North of Texas, USA
Email : boukerche@cs.unt.edu

Luciano Bononi[†]
Department of Computer Science
University of Bologna, Italy
Email : bononi@cs.unibo.it

1 Introduction

Mobile ad hoc networking technologies and wireless communication systems are growing at an ever faster rate, and this is likely to continue in the foreseeable future. Higher reliability, better coverage and services, higher capacity, mobility management, and wireless multimedia are all parts of the potpourri. The evolution of new systems and improved designs will always depend on the ability to predict mobile, wireless and ad hoc networks' performance using analytical or simulation methods. Modeling and simulation are traditional methods to evaluate wireless network designs. To date, mathematical modeling and analysis have brought some insights into the design of such systems. However, analytical methods are often not general or detailed enough for evaluation and comparison of various proposed wireless and mobile systems and their services. Thus, simulation can significantly help system engineers to obtain crucial performance characteristics.

However, detailed simulations of these systems may require excessive amounts of CPU time, and their execution on sequential machines has long been known to have computational requirements which far exceed the computing capabilities of the fastest available machines. For instance, it is not unusual for simulations of large wired and wireless networks to require hundreds hours or even days of machine time. As a consequence, the development of methods to speed up simulations has recently received a great deal of interest [6, 7, 8, 9, 22, 38, 65].

With the ever increasing use of simulation for designing large and complex systems, wireless mobile and ad hoc networks have brought several challenges to the parallel discrete-event simulation (PDES) community. The challenges require not only extension and advances in current parallel and distributed simulation methodologies, but also the discovery of innovative approaches and techniques to deal with the rapidly expanding expectations of wireless network designers [6].

In this chapter, we shall present some guidelines related to Mobile Ad Hoc Networks (MANETs) modeling and simulation, several sequential network simulation testbeds, and distributed simulation testbeds for

---

[*] This work was supported by the Texas Advanced Research Program ARP/ATP Grant 003594-0092-2001.

[†] This work was supported by the Italian MIUR FIRB-PERF "Performance Evaluation of Complex Systems: Techniques, Methodologies and Tools" project funds.

wireless and mobile networks. We shall also address the challenges PDES community has to face in order to design high performance simulators.

2 Design and modeling of wireless and mobile ad hoc networks

In this section, we shall introduce the basic characteristics and major issues pertaining to Mobile Ad Hoc Networks (MANETs) modeling and simulation. A complete definition of all aspects of interests and a fit-all solution for simulation is not possible and it is out of the scope in this chapter, because it depends on the objectives sought by the designers and the assumptions they have made. Thus, we will point out only some of the most promising and interesting challenges and solutions to the modeling of mobile ad hoc networking and communications, and we will present the state of the art of simulation of wireless, mobile and ad hoc systems. The main purpose of a simulation-based study for a MANET system is to obtain detailed information about performance figures, behavior, overheads, quality of service, and many other metrics regarding the system, protocols and policies adopted at many levels of the ISO/OSI protocol stack [12, 18, 23, 29, 56]. Among all of the model parameters, one defined as "factor" is selected as the varying parameter whose effect on the performance indices is to be evaluated [29]. Evaluating system performances via modeling and simulation consists of two preliminary steps: *i)* defining the system model, and *ii)* adopting the appropriate simulation technique to estimate the metrics needed to evaluate the performamce of the system. In what follows, we will first talk about MANETs modeling: some concepts can be considered general for every wireless and mobile system (e.g., wireless PCS, cellular networks), whereas others can be considered specific to MANETs.

2.1 Mobile ad hoc networks modeling

As stated before, it is not convenient to talk about MANET models without defining the set of objectives and questions the simulation experiments should answer to. Every system model is tailored depending on the goal of the simulation project. Any unrequired additional detail will introduce overheads, possible errors and a slow-down of the simulation process [26]. Any missing detail, relevant for the performance evaluation of the system, will also introduce errors, approximated results, and the need for additional model updates [15, 26]. General purpose models are known to be very complex and hard to adapt to specific system models. Today, many simulation tools provide a library of simulation models written by professional modelers and researchers [20, 45, 46, 47, 55]. Many times, when incremental updates are performed by different people, the model validation becomes a time-consuming and difficult task to overcome [15]. Most of the times, models are supplied or exchanged without any comments and/or their documentation, requiring a great effort to the designer to interpret and validate them [15]. Today, most of the models can be defined by using *object-oriented paradigms* and *languages*, such as Java, C/C++ and OTcl/Tk, just to mention a few. This makes it possible and more practical to extend, adopt, exchange and re-use existing models in new simulation projects. Inheritance allows to create module hierarchies, and instances of complex objects, with a simple management of model libraries. Widespread adoption of object oriented languages works in favor of model

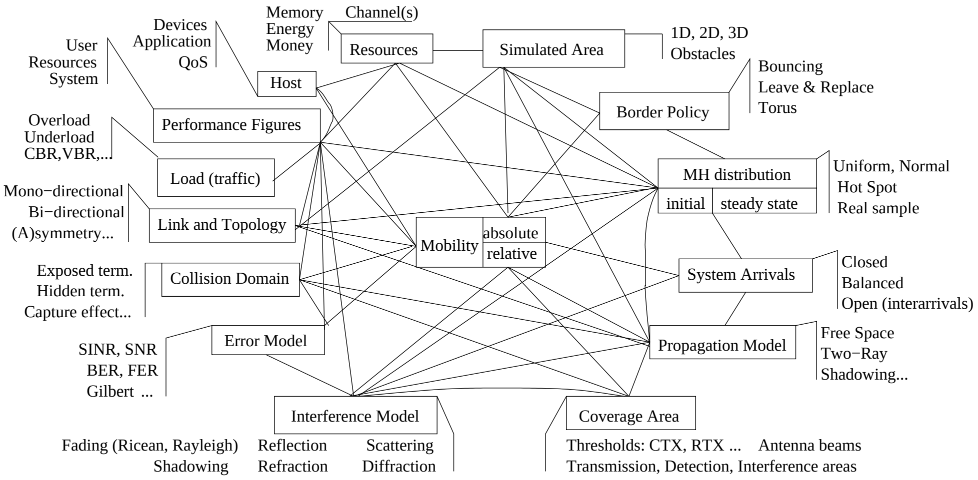


Figure 1: **The modeling roadmap**

distribution among researchers. C++ models and tools are usually adopted and preferred to Java-based tools for simulation-performance purposes. Modeling component-based units can be performed with the adoption of high-level compositional languages and a set of application tools [45, 46, 47, 55].

In the performance evaluation of a wireless, mobile ad hoc system, every simulation experiment should be done under a variety of modeling conditions and factors, in order to capture detailed and "realistic" effects of the real system. These conditions should be well-defined and may have wide and clear interactions at various levels in the model. As an example, correct model design should a-priori evaluate any possible relationship among physical, topology and mobility levels, up to the protocol layers such as Medium Access Control, Logical Link Control, Network, Transport, and Application. Examples of such conditions include transmission ranges, power consumption, detection thresholds, data-traffic sources and loads, buffering storage, user mobility and topology restrictions, signal propagation and obstacles, interference and bit errors, just to mention a few.

Figure 1 shows a roadmap of some modeling issues that will be considered in this chapter. The edge-based representation of the multiple effects, relationships and conditions among the modeling issues was shown just to put in some evidence the non-trivial work of the modeler. Also, figure 1 emphasizes that mobility plays a central role in the model design. All of these conditions need to be represented and managed within the system model, by means of well-defined and efficiently manageable data structures. Many solutions to model testing conditions, in many simulation studies, have been proposed in the literature for wireless and mobile systems' models, and will be discussed in this chapter. Specifically, simulation models for wireless and mobile systems in general, and MANETs in particular, have to deal with at least two innovative concepts with respect to wired networks' models: the *user-mobility* and *open-broadcast nature of the wireless medium*. In this chapter we will discuss the model definition issues related to these innovative concepts for wireless mobile systems, with a special attention to MANETs. Our presentation starts from the bottom to the upper levels, i.e., physical, topological, mobility models, up to protocol layers. The model implementation would depend on the model-definition languages and simulation techniques and tools adopted, thereby requiring additional validation and verification efforts. Hence, we will not discuss in detail the model implementation, verification and validation in this chapter.

2.1.1 Simulated area and boundary policy

In this section, we shall discuss the simulated area and the boundary policy concepts, as well as a relevant set of assumptions related to the design and modeling of the simulated area, which may have many effects on the simulation of the target system [3, 12, 13, 32, 55]. This area is the virtual theater of execution of the simulation, and the area size is not important in this discussion. The area size becomes important when coupled with other parameters governing the mobility, propagation and node-density models considered in the proposed scenario.

First of all, the *simulated area* of interest is limited (i.e., it is bounded by limit borders), it can be mono- or multi-dimensional, and can be represented by Cartesian coordinates[1] as follows:

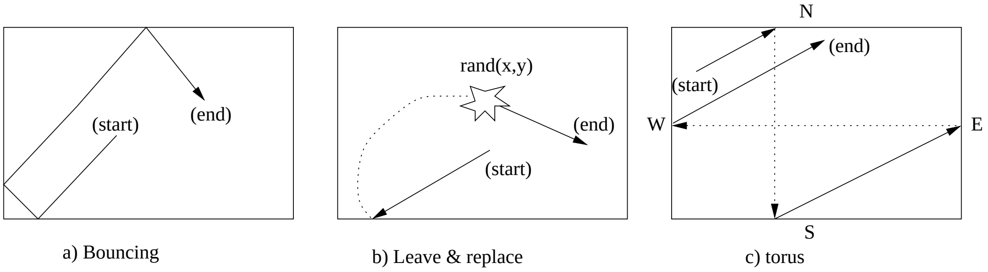
- *Mono-dimensional (1-D)* area is a simple linear path for a set of MHs (e.g., a simple highway simulation model). The relevant parameter is one single x coordinate along the linear path, varying in the limited range $[minX, maxX]$. Such model can be used in cellular systems, assuming a linear path between a set of adjacent cells, and it is unfrequent in MANET systems' simulation.
- *Bi-dimensional (2-D)* areas are the most used models, because they allow to embed and map any possible user-path in a real (flat) geographical area. Every portion of a real geographical area can be mapped on a 2-D grid with (x, y) coordinates varying in the limiting ranges $[minX, maxX]$ and $[minY, maxY]$. Definition of sub-gridding cells can be exploited to manage and to sample object distributions. As an example, hierarchies of grid cells can simplify the management of "neighbors" objects in adjacent cells, and can support object distribution policies (e.g., n objects per grid-cell). When object density evolution is required to be evaluated, grids allow a consistent, snapshot-based sampling and runtime calculation of the objects' distribution.
- *3-D models*: sometimes, 2-D models can be extended to three-dimensional space models (x, y, z), e.g., when modeling user mobility inside buildings with many floors, the user mobility can be described including vertical movements as in staircases and elevators [34].

The simulated area may also be enriched with obstacles, affecting user mobility and propagation of signals. A brief discussion will be presented in the following sections about propagation and mobility models. Obstacles can be modeled and realized as additional data structures.

The *boundary policy* is another relevant characteristic of the simulated area that one might consider in the model design. This policy defines the behavior of the Mobile Hosts (MHs) when, due to the motion process,

---

[1] Note that polar coordinates can also be used.



a) Bouncing b) Leave & replace c) torus

Figure 2: **Three boundary policies**

they reach the boundaries of the simulated area. Many possible solutions have been proposed depending on the simulation and model requirements. The modeler should be careful about the effect that the boundary behavior, composed with a mobility model, may cause on the resulting spatial node-distribution [3, 4, 5, 13]. In what follows, we shall distinguish between three boundary policies, widely used by the wireless and MANET networks' modelers (see figure 2).

- The *"bouncing boundary"* solution requires to bounce back MHs towards the simulated area when they are going to move outside [3, 5, 13] (see figure 2.a). This simple solution can be used if the number of MHs in the simulation is required to be constant (e.g., a closed system). Different "bouncing" rules can be adopted, e.g., *mirror reflection* (e.g., preserving the angle of incidence θ in a bouncing angle $-\theta$ or $\pi - \theta$, respectively) and *random reflection* (i.e., a random reflection angle θ is generated). This boundary policy can be considered as quite unrealistic for large areas, and quite approximated for simulation of indoor mobility (e.g., people moving in a room). A modeler has the choice to preserve the state of a bouncing MH or to create a new instance of MH when it virtually "leaves" the area (i.e. the MH hits the area boundaries). This choice may be useful if the state information of the MH is relevant for the simulation process, or for the protocol to be tested (e.g., protocols based on MH's history and evolution state).

- The *"leave & replace"* variant of the bouncing boundary policy is to delete a "bouncing" (leaving) MH and clone it in a randomly chosen position within the simulated area, following any node-position distribution (see figure 2.b). As in the "bouncing boundary" solution, the cloning of the leaving MH can preserve state and history information on the new instance or not, depending on the modeler choice. This policy may result in a non-uniform steady-state spatial node distribution, with a node concentration around the center of the simulated area [3, 5, 13]. Intuitively, this is due to the fact that MHs leave from the boundaries and re-enter by choosing a randomly distributed position "inside" the area. This also means a biased (i.e. reduced) probability to find MHs moving from the borders towards the middle of the simulated area. This solution is rarely considered useful in MANET's simulation.

- The *"torus boundary"* solution is another policy widely adopted by many researchers [3, 13]. In this policy, when a mobile host reaches the $North, West, South, East$ boundaries of any rectangular area, it simply leaves the area and re-enter with the same direction and speed from the $South, East, North, West$ boundary, respectively (see figure 2.c). Intuitively, the rectangular area is wrapped-around itself, North with South, and East with West, like a rubber-ring. The reason why the torus policy is widely used is given by its simple implementation, and because it simplifies the management of uniformly distributed host densities and directions (accordingly with the implemented motion models). There is a full correlation effect balancing leaving and re-entering hosts' directions and velocities. Again, leaving MHs can preserve state and history information when they re-enter the area, depending on the modeler choice (excepted velocity and direction that should be maintained anyway).

2.1.2 Host sources and position-distributions

Host mobility is the main cause in determining the "arrival" and physical presence of a set of MHs within a fixed simulated area. The physical presence of hosts does not necessarily represent the relevant factor to be modeled for the simulation goals. This mainly depends on the MHs' roles to be modeled and on the performance indices required. As an example, a switched-off MH in the simulated area may not be considered as relevant to determine transmission-based performance indices. We denote as "active" the role of a MH that can be considered effective in determining the value of a performance index whose evaluation is the goal of a simulation run. Every performance analysis for a wireless mobile system can be performed by assuming a (fixed or variable) number of "active" mobile hosts (MHs) implemented in a limited area of interest. In what follows, we focus upon active MHs, and consider only the motion-related "presence" of these MHs. The modeler should evaluate the opportune and realistic definition of the average density of "active" MHs, with attention to the policies for the creation and position-allocation of new MH-instances, both at the simulation start, and at runtime [54].

Dealing with the sources and creation policies, accordingly with the factors influencing the performance metrics of interest, one possible choice is to require the number of active objects in the simulated area to be constant. This choice may be useful, for example, when performance metrics to be obtained are related to the number of MHs, or to the MHs' density (e.g. existence of route path, network partition, average degree of MHs). In the following we denote MHs' presence in the simulated area as "active presence", whatever meaning this would imply.

- in *closed systems*, the initial number of "active" simulated MHs is constant, and every MH lives (i.e. it maintains its functionalities) for the whole simulation run.
- *balanced systems* realize a simple hybrid solution that can be useful for some simulation analysis. Every time a MH leaves the simulated world (e.g., moving outside the simulation area, or switching off the network interface) a new instance of the MH is causally introduced in the simulated area, following the selected position-distribution. Bouncing, torus and "leave & replace" boundaries allow a natural implementation of a closed or balanced system under the MHs' mobility viewpoint (if MHs' sources and sinks are missing).

- In *open systems*, one or more sources of active simulated MHs are defined (e.g., inter-arrival or activation processes for MHs): in such scenarios the modeler should evaluate and select inter-arrival time distributions for the sources (e.g., Exponential or Poisson distribution), sink policies (e.g., MHs with no battery-energy, or moving outside the simulated area, are discarded) and the initial-position distribution for incoming MHs (e.g. at random uniformly-distributed coordinates in the area). If the arrival process is too fast, the system is *unstable*, i.e. the asymptotical number of MHs in the simulated area is not upper bounded. This can give a biasing problem if we are interested in the evaluation of performance metrics that may be related to the average MH density (e.g., multi-hop link reliability in routing protocols, network connectivity and partitions, average next-hop distance, average transmission power, etc.) [3, 12, 13, 23]. To obtain an open, stable system, the rate of MHs leaving the area should be statistically balanced by the summatory of arrival rates of the MH sources. This means that the number of MHs in the area is not a constant value (like in closed and balanced systems), but it converges asymptotically.

One possible choice for *initial allocation* of a new MH's position is a random selection of its position coordinates within the simulated area. Uniform or Normal distributions are widely used in the literature, depending on the host density to be modeled (e.g., uniform vs. hot-spot density, respectively) [13, 55]. This choice does not provide any best-effort guarantee about network partitions. One possible solution to this problem is to divide the simulation area in a grid of square-cells, with a size that could be determined by the minimum range of connectivity among the MHs, and distribute MHs' positions such that at least one MH is given in every cell of the grid.

On the other hand, a real, sampled distribution-snapshot of MH-positions can be used, if available. This is a common way to model hot-spot MHs' distribution [55]. In many scenarios, depending on the mobility models and their respective parameters adopted for MHs, the initial distribution of MHs is less or more relevant to determine the steady-state MHs' distribution [3, 13]. When the "memory effect" of the initial distribution is not preserved by the motion model characteristics, every possible transient effect of the initial distribution should be evaluated and eliminated to collect unbiased steady-state results.

The choice for *runtime position-allocation* policies for newly generated MHs is a little bit more subtle. Random allocation of MHs may result in non-uniform distribution and biased node-density, e.g. if hosts leave the system only from the boundaries [3]. A possible solution for this scenario would be to "delete" hosts selected randomly in the simulation area, and to adopt distribution-balancing boundary policies, like bouncing and torus borders [3, 5, 13].

2.1.3 Coverage areas, physical propagation, error and interference models

Usually, in MANETs, every host can be considered as a potentially mobile host. As a consequence, hybrid MANETs under analysis today may include fixed, static Base Stations (BS), with their respectively managed coverage areas, like in cellular and PCS systems [18]. A detailed physical model for wireless transmission, including propagation, mobility, error and interference issues, is one of the most difficult and computationally expensive tasks to do, and usually strong approximation and assumptions are introduced [2, 3, 12, 35, 42, 62, 66]. Many models and solutions have been proposed, at different levels of detail [26, 59]. We will skip most of the details, for space reasons, and we will just point out some of the modeling issues related to MANETs.

The physical wireless transmission is based on the emission of electro-magnetic waves coding information with many possible modulation and coding techniques. The natural decay of transmitted signals can be modeled following simple analytical approximations. If the residual signal power of the receiving network interface is above the *detection threshold* then a communication is possible. Otherwise, to allow a communication (link establishment) between the intended sender and receiver it would be necessary to increase the transmission power of the sender, and/or to reduce their relative distance d.

One of the most used propagation models, adopted in MANET's simulation, is the simple *Free Space Propagation Model* [12, 53]: if Pt is the transmission power (i.e. $energy/time$) used for the signal transmission, then the receiving power Pr is proportional to $1/d^2$, where d is the distance between sender and receiver in open space (see figure 3).

The Free Space Propagation Model can be extended to better describe the effects over near and far receivers, with the *Two-Ray Ground Reflection Model* [12, 53]: the model is the same as Free Space, excepted when the distance d is greater than a crossover point, called *reference distance* (around 100 meters). For such long distances the receiving power Pr is modeled as proportional to $1/d^{\beta}$, $\beta > 2$.

Free Space and Two-ray propagation models assume ideal propagations over a circular area around the transmitter. To model irregular coverage areas, the *Shadowing propagation model* [45, 53] is defined with two components: a component similar to the Free Space model, and a random component to make randomly variable (and statistically controlled) the edge of the communication range. For a complete discussion of the *Free Space Propagation model* and other models, see [26, 53].

A modeling choice to define if a transmission can be detected by a tagged receiver, is to define a *receiving threshold, RTX* and a *carrier-sense threshold, CTX* for every device [12, 59, 53]. For every simulated transmission, it would be required to scan every MH in the system and to apply the propagation model to the transmitted signal. This requires to evaluate, for each receiver, if the receiving power perceived for the ongoing transmission is sufficient for reception (i.e. greater than RTX), if it is sufficient for detection and carrier-sensing (i.e. greater than CTX), or if it is simple interference. Reception and carrier sensing events can be passed to the model components devoted to manage events at the upper layers of the model, e.g. Medium Access Control policy implementation. This scan-based computation may require long time, if performed for a large set of MHs.

The system model can be extended with *coverage areas*, in order to reduce the transmission-detection overhead, and to model much more complex propagation models, depending on the modeling and simulation requirements. Transmission coverage areas definition can be directly associated to every transmitter, but the area size and shape is relative to the receiver thresholds. For ease of management, the area-size definition

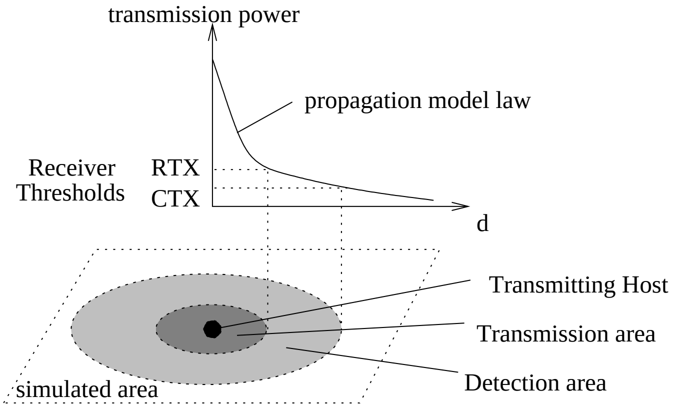


Figure 3: **Transmission power, propagation and coverage areas**

would require to assume common threshold levels (i.e. common CTX and RTX values) for every MH in the system (see figure 3).

The *transmission (coverage) area* of a wireless transmitter can be defined as the area where the transmitted wireless signal propagates and can be correctly detected and decoded (i.e. transmission is possible with few/no errors due to interference). This area depends on the transmission power of the transmitter, on the propagation model, on the reception threshold (sensitivity) of the receiving network interface (RTX), and on the amount of interference (noise) caused by many possible factors (described in the following). The transmission area should be defined and managed in the model, for each MH, in order to dynamically evaluate a communication capability (i.e. a direct link) between every candidate transmitter and receiver.

The *detection area* (see figure 3) of a wireless transmission-device is the area where the signal propagates, and where it can be detected by a carrier sensing mechanism, without being necessarily decoded (i.e. $CTX \leq Received\_Signal\_Power \leq RTX$). This means that a mobile host can sense the wireless medium as busy, without being able to decode received signals. The definition of this area in the model, for each MH, may be relevant for the evaluation of detailed carrier sensing and MAC level effects, such as exposed terminals, hidden terminals, capture effects (described in the following) [24, 52].

The *interference area* of a wireless transmission-device is defined as the area where the transmitted wireless signal propagates, without being detected or decoded by any receiver, adding interference and noise to any possible ongoing transmission for intended receivers. The cumulative effect of noise (i.e. interference) might add errors to the transmission of bits of information. The definition and management of this area in the model, for each MH, may be relevant for the evaluation of detailed interference and error models. Transmission areas are included in detection areas, and detection areas are included in interference areas, given the propagation properties of wireless signals in open spaces. Possible choices to model coverage and interference areas in open spaces are given by regular polygons centered in the transmitter's position. Circular coverage areas can be defined for open space propagation models, and are a simple common choice for MANET's modeling and simulation purposes. The circle radius, centered on the transmitter position, can be made proportional to the transmission power in adaptive way, mapping to real power control management and policies implemented in simulated MHs. If fixed (static) transmitters are present, with a constant transmission power, like in cellular and PCS networks, a common choice is to approximate the coverage areas by hexagons, squares, and Voronoi diagrams. This choice can simplify the link management, because connectivity between a MH and a fixed transmitter can be evaluated with no ambiguity (i.e. only one reference transmitter is defined in every point). This choice is simple to define and to manage for the simulation purpose, but it realizes a strong approximation of the real behavior of wireless transmissions. The circular coverage model is quite realistic in open spaces, but it would require some changes if obstacles are present to interfere with signal propagation.

When *obstacles* are present in the simulation area, the coverage and interference areas may be severely affected, in almost unpredictable ways [35, 47, 53]. This is even worst if mobility of wireless sources (or, equivalently, wireless receivers) is present. Models to deal with obstacles have been defined e.g., in [32, 47, 53, 55]. Moreover, if the antennas are not omnidirectional and the transmission beam is not *isotropic*, the regular polygon choice for coverage areas is even more approximated (e.g., with smart antennas, the transmission energy is not uniformly propagated in all directions, but has a directional effect [53]). Modeling of asymmetric beams and obstacles in these scenarios might be quite complex, and computationally expensive too.

To model *interference effects* many additional factors need to be defined, and a realistic model is quite hard if not impossible to obtain, without paying for a high computation overhead. As a simple description of the wide set of problems to deal with in accurate interference-modeling of the physical wireless transmission, we present a short list of system details that should be considered in theory, and that aren't usually considered in many models, given the complexity and extensive computation required to simulate their effects. Most of the described problems are given by continuous physical phenomena, whose approximate modeling in the discrete-event simulation field would be really hard and expensive. Mobility is an additional source of problems. In MANET scenarios, the model would be even more complex than in cellular and PCS networks, because usually both the transmitter and the receiver can be mobile, and a distributed, *relative-mobility* parameter should be evaluated, instead of a local, absolute-mobility parameter; Physical problems to be modeled in wireless transmission include the following phenomena [53]:

- *fading*: a physical phenomenon, frequency dependent, inducing delay and phase variations between the main transmitted signal (following a dominant path) and many secondary signals (following alternative paths) caused by obstacles and mobility. This causes long term and short term variations of the resulting reception-power of transmitted signals. The *Additive White Gaussian Noise* model is used to represent ideal channel conditions under the signal fading viewpoint. *Rayleigh* and *Ricean* Fading are widely accepted frameworks [2, 35, 59, 53], to model fading prone scenarios. They can be applied to highly mobile scenarios, with No Line of Sight (NLOS) and with Line of Sight (LOS) paths, respectively [59]. The K parameter of the Ricean Fading model can be used to control the composed effect of LOS and NLOS signal powers [59]. A *Coherence time* parameter is adopted to control the time frequency and duration of fading effects on the channel. As an example of the modeling complexity for fading effects, let us assume that there are M base stations and N mobile hosts in the simulation scenario, and there are roughly L paths determined by obstacles in each propagation direction, then we would need approximatively up to $2M * N * L$ instances of Rayleigh fading generators. Efficient implementation of fading models is still an ongoing research activity [2, 35, 53].

- *shadowing*: attenuation of signal power propagation caused by physical obstacles. This effect is mainly responsible for irregular coverage areas. The Shadowing propagation model defined in the previous section gives a statistical approximation of this effect [53].
- *reflection*: signal reflection caused by large obstacles, and by indoor walls. This effect is quite important to be modeled for indoor scenarios;
- *refraction*: marginal signal change and reflection caused by variation in the medium density;
- *scattering*: signal diffusion caused by sharp obstacles;
- *diffraction*: signal deviation caused by large edges and corners;

Each one of the above mentioned phenomena may have different characteristics, given any different physical implementation and any different coding technique adopted for wireless transmissions. The whole effect of such a collection of complex phenomena is usually modeled as a simple error probability for a given amount of information received (e.g., a bit) on the physical channel. The idea is to close all this stuff in a black box describing the whole effect as the *probability to obtain a bit error*. Obviously this may be a strong, unacceptable approximation, depending on the aim of the simulation. In many models, in order to approximate the real behavior of the wireless medium, the physical medium (or its high-level abstraction: the channel) behavior can be described with more accurate error models. Signal to Interference and Noise Ratio (SINR) and Signal to Noise Ratio (SNR) are the key parameters adopted to model the signal composition of interference effects described above. The generalized SINR and SNR values, together with RTX and CTX thresholds, can be adopted to model with some detail the high-level effect of interference resulting in Bit Error Rates (BER) and Frame Error Rates (FER). BER and FER values model the generalized probabilities that a transmitted bit or frame is received with errors, respectively. FER is a function of BER and the frame length (in bits). In order to capture the bursty effect of wireless transmission errors, the *Gilbert-Elliott error model* has been used to define the status of the wireless medium as a function of time [62]. This model defines a random Markovian process between the following two states: *Good* and *Bad*. Good status is characterized by low probability of bit error (low bit error rate, BER), while bad status is characterized by high bit error rate. The time in bad and good status is usually sampled from a exponential (or its discrete counterpart: geometric) time distribution, with respective parameter and average values. It is a good choice to implement it in time-slotted models.

The coding techniques adopted for the wireless transmission, and the frequency spectrum allocated for adjacent channels, are additional parameters to be evaluated in order to define opportune error models for the wireless communication. New coding techniques allow for interference effects' cancellation, and interference models should be defined for adjacent channel interference and co-channel interference [53]. We skip all the details for space reasons.

2.1.4 Link definition and network topology

The modeling of coverage areas is really important because it is related to the link definition between any couple of MHs, or between a MH and a BS. This is really important in MANETs, because a simple star-topology given by a set of MHs around a BS is not a concrete and dynamic vision of the system. Moreover, in modeling and simulation of MANETs, the "link-established" property between a couple of MHs is more complicated than in most wireless cellular networks, mainly due to the management of relative mobility (as opposite to absolute mobility) of these mobile hosts [66]. For any couple of MHs and/or fixed hosts (e.g., Base Stations) covering the area of interest for the simulation (call them host X and Y), we have three possible expected scenarios related to reception and carrier sensing thresholds, coverage areas and link definitions [24]:

- X is out-of the transmission area of Y, and vice-versa: this means that X and Y are partitioned (e.g. see A and C in figure 4). The network topology does not assume to have any direct communication link between X and Y. Maybe communication between X and Y is possible, at the upper routing layer, if supported by an intermediate-hosts chain of mutually reachable hosts (e.g. like host B and C in figure 4).
- X is within the transmission area of Y, and Y is out of the transmission area of X: this means that Y can communicate to X, but not vice-versa. In this scenario, a mono-directional link exists from Y to X (e.g. see hosts A and D in figure 4). Mono-directional links exist in many real scenarios, mainly due to the different transmission power and propagation characteristics of MHs (e.g. see host D in figure 4). The obtained network topology is a direct graph based upon the mono-directional links.
- X is within the transmission area of Y, and vice-versa: X and Y are mutually reachable via a wireless bi-directional link (e.g. see hosts B and C in figure 4). Depending on the coding techniques and channel bandwidth allocated for the physical channel, it may happen that the bi-directional link is not a symmetric link. A bi-directional link is *symmetric* if the physical channel capacity (i.e., the maximum bit rate obtained for wireless transmission) is the same for both link directions (otherwise it is asymmetric). Many simulation models usually assume bi-directional and symmetric links, for ease of implementation. The assumptions about these scenarios may severely influence the modeling and performance results in the evaluation of network protocols, e.g., in routing protocols and multi-hop communication.

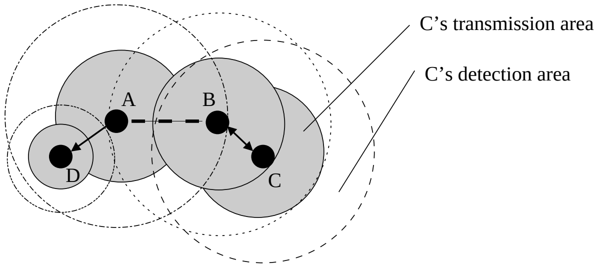


Figure 4: **Example of a collision domain**

Dealing with discrete, event-based simulation, the critical question related to MANET topology management in the simulation process is: *what is the simulated time of next link-state-change event that will be expected, given the current relative mobility pattern and coverage areas of mobile hosts?* The answer to this question would require little or more computation, based on the model and data structures defined to implement the simulation. Every link-state-change event can be calculated, based on current coverage and mobility conditions, and its execution scheduled in an ordered event-list. Any intermediate change in the speed, direction, transmission power, of anyone of the involved hosts would require to delete the causally-dependent, scheduled link-state-events and substitute them with the updated ones. Moreover, this event-list management is at the basis of any discrete event simulation. This indicates clearly how complex and computationally hard the mobility and link management in MANETs could be.

Once the link existence is established, many other conditions should be managed regarding the high-level link properties. As an example, in wireless physical channels it is not possible to receive a communication on the channel, while a simultaneous transmission is performed on the same physical channel [24]. A bidirectional (Full Duplex) link can be obtained by adopting Time Division Duplex (TDD), or Frequency Division Duplex (FDD). TDD consists in splitting the transmission and reception phases over adjacent, non-overlapped time intervals on the same physical channel. FDD consists in adopting two physical channels: one for transmission and one for reception. In MANETs modeling and simulation, Time Division Duplex is commonly adopted. All data transmissions and receptions have to be in the same frequency band, since there are no ”bridge” nodes (maybe excepted Base Stations) to translate the transmissions from one physical channel to another one. This usually requires strict time-synchronization in the system, and Medium Access Control (MAC) protocols’ definition [24]. Frequency Division Duplex may be adopted (together with TDD) in centralized networks (like cellular) characterized by up-link and down-link channels [24].

The coverage area management in the model can be used to simulate additional MAC level details relevant for MANETs, like *collision domains.* A collision domain can be defined as the coverage area shared by a set of MHs, mutually connected by a single shared communication channel (i.e. a single logical channel). Collision of concurrent signals transmitted on the same collision domain would cause a destructive interference for detected signals on the receiver. The main task of a MAC protocol policy is to avoid such collisions, mainly avoiding new transmissions to start while another transmission is being detected. A detection-based policy for a MAC implementation may result in some problems whose investigation would require an accurate coverage-model definition. As an example, let us suppose A is within the *detection area* of B and vice-versa, B is within the *transmission area* of C and vice-versa, and C is outside the *detection area* of A (see figure 4). In this scenario A senses the transmission of B, i.e., it senses the channel as busy, but it cannot decode the transmission. This condition is often modeled in order to obtain a real performance investigation about *exposed terminals* [24]. Exposed terminals are terminals, (e.g., A) whose transmission is not allowed, (e.g., by a MAC policy over a collision domain) due to exposure to unrelevant transmissions, (e.g., B to C). A similar problem is given by *hidden terminals*: due to shadowing effects and limited transmission ranges, a given terminal C could start a transmission towards another terminal B (C and B are within each other's transmission area, see figure 4), while B is receiving signals from a hidden (with respect to C) terminal A. This means that B cannot complete any reception, due to the destructive collision of signals from A and C. It may happen also that B can detect and isolates one of the colliding transmissions, (e.g., from C to B): in this case, we modeled a *capture effect* of transmission from C to B despite A's interference and collision. A discussion of details for hidden and exposed terminals and modeling of capture effect can be found in [24, 52]. A rough modeling, based on a shared, global boolean variable $Channel = Busy$ or $Idle$ would not describe with required accuracy the previous scenarios.

This discussion was given to illustrate how, in the simulation plan, the definition and the structure of the coverage-area, topology and interference models may include the information required to perform a realistic simulation. Anyway, detailed models are quite complex and simplifying techniques and assumptions are widely adopted. In the following we are going to describe another relevant characteristic of MANET and wireless networks' models, adding additional complexity to the model definition and management: host mobility.

2.1.5 Mobility models

User mobility is the main added value of wireless networks. Accurate simulation results would require accurate details to be modeled, and many fine-grained, low-level causal effects to be kept into account in the simulation process. Mobility has a central role, and is a relevant background effect to be modeled in almost every simulation analysis of wireless systems (see figure 1). The effect of mobility on the system policies and protocols is relevant at many layers: dynamic topologies, due to simulated hosts mobility, map on causality effects in the areas of influence of each mobile device, resulting in dynamically shaped causality-domains [6, 24]. The effect of mobility introduces adaptive behaviors of users, protocols and applications. Moreover it may happen that mobility models are related to the physical scenario under consideration [55, 60]. Mobility patterns may be, sometimes, application dependent [13, 55]. Most of Medium Access Control, Routing and Transport protocols proposed for MANETs scenarios are customized and designed for specified mobility models, and behave better than a general purpose protocol for that given scenario. The evaluation of user-positions can be a computationally relevant task in a wireless mobile system's simulation, due to the mobility and high number of events related to the user position. In two or more neighbor-hosts simply sharing the wireless medium (without any end-to-end communication session on) the causal effect of signal interference, due to mobility, could result in a chain of local-state events from Medium Access Control (MAC) up to the Transport and Application layers [23, 59]. In MANETs, given the infrastructureless architecture, some of the mobility models adopted for cellular systems are not appealing. As an example, *Markovian models* (random walks) described by cell-to-cell migration probabilities, or *Fluid Flow models* whose characteristic is to describe host mobility in terms of "the mean number of users crossing the boundary of a given area" are not considered as relevant as Random and Restricted mobility models, Gravity models or Group Mobility models.

In general, two types of mobility models can be adopted in the simulation of wireless mobile networks, and specifically for MANETs: *motion traces* and *synthetic models* [13].

Motion traces provide accurate and realistic information about the user mobility patterns and behavior, in particular when the user mobility is related to real users in a bounded scenario, (e.g., downtown streets, highways) [3, 13, 55, 60]. Unfortunately, traces require large log files depending on the number of tracked hosts and the time granularity of samples. Traces are significant descriptions about the steady-state mobility of a user only if the motion samples are collected for significant time intervals. If the sample frequency is low, approximated solutions (e.g., interpolation and dead-reckoning) can be used, but this requires additional computation, and may result in strange behavior (like users walking through obstacles instead of turning around them). Moreover, traces can be collected only for existing systems, and MANET traces are still hard to find mainly because large MANETs scenarios have still to be implemented and user applications have to be defined. Motion traces solve the problem to define the initial and runtime position-distribution of MHs, in a deterministic way. Another interesting characteristic of motion traces is the ability to capture the real correlation effect between user mobility and real application/user needs. It may happen that users move driven by the application needs, e.g., in order to reach good coverage areas. Also, it may happen that users move showing a group behavior [13, 28, 61]. Synthetic models trying to define a similar correlation effect, for single MHs and MHs' groups, will be defined in the following.

Synthetic models are defined to represent the mobility of users in a realistic way, without using traces. Many synthetic models have been defined in the past to be adopted as analytical models [3, 13]. The main requirement for such analytical models was mathematical tractability, in the place of realism. Such models survived also in many simulation studies, mainly due to validation possibilities they offer with respect to the simulation counterpart [5]. In other scenarios, random motion models far from reality can be adopted in order to stress a given mechanism or protocol, emphasizing worst-case scenario results [13, 27]. Random motion models have been recently extended introducing correlation effects, restrictions and group behaviors, in order to meet the requirements of mobile systems' modeling and simulation [3, 13]. We can distinguish between three degrees of "randomness" in the classification of Random models [3, 5]: i) models that allow users to move anywhere on the simulation area, following pseudo-random processes to select speed and direction, ii) models that bound the movement of users (like streets, walls, etc.) but allow for pseudo-random selection of direction and speed at crossings (like City Section and Manhattan Model [42]), and iii) models based on predefined paths (deterministic paths).

In the following, we define and discuss a list of synthetic models used for MANETs' simulation.

- *Random Mobility Model*: it is a discrete interpretation of the Brownian motion model [3, 13]. It is completely unpredictable and it has the memoryless property for speed and direction. This means that current speed and direction is not related to the speed and direction history, and speed and direction are completely uncorrelated (i.e., two independent stochastic processes) [3]. Many mobile users adopting such motion model result in completely uncorrelated mobility, and the mobility pattern is quite unrealistic. This is a typical worst-case modeling assumption, e.g., when the analysis should demonstrate that adaptive protocols' performance does not rely on any motion-correlation and/or predictable-position assumptions. This model can be used for vehicular and large scale environments, and can be implemented in many ways. Assuming a 2-D motion, one possible implementation is the following: every mobile host (MH) moves from a current location to its new location defined by a uniformly distributed pseudo-random choice of a new direction θ in the polar angle interval $[0, 2\pi[$, and by a uniformly distributed pseudo-random choice of a speed value s in $[minSpeed, maxSpeed]$. The speed and direction are maintained for constant time values ts and td, respectively (this is a good implementation for simulations with slotted-time management). If time management is not slotted, an equivalent choice is to uniformly generate direction θ and speed s to be maintained for a constant distance ds and dd, respectively. This can be a good choice when the system to model has underlying grid topologies, e.g., cells, and every single cell-migration is a relevant event. Given a similar choice, the "next move" events are not synchronous, in the simulation. This can give additional problems and computation needs: if it is required to obtain the position of neighbor-MHs, before every move, then every neighbor-MHs' position would need to be interpolated. Dealing with the simulated area boundaries, if the area limit is "bouncing" off the MHs, e.g., with a direction proportional to the angle of incidence, in 1-D and 2-D there is an interesting property: every MH will randomly move around its initial position [13]. This represents also a pitfall for this model: if the initial allocation of MHs is not uniform, and the average speed is low, then the "memory effect" of the initial distribution would be persistent, and clusters of MHs could be maintained, despite the randomness of the motion model. This behavior may affect the assumptions about the average degree (i.e., number of neighbors) of a mobile host during the simulation. If the time values ts and td are long, given the bouncing behavior of area boundaries, the average distribution of MHs would be concentrated in the middle of the simulated area (because when a mobile host is near the boundaries it has a high probability to be reflected, or to choose a new direction, towards the center of the area) [3, 13]. If ts or the average speed is great, the node density could be made quite uniform with a torus border area policy [3]. Many additional choices or assumptions can be made for this model. The main factors to define for analysis based on this model are the speed ranges and average speed values. The speed factor defines how far a mobile host can roam away from its initial position, and should be dependent on what is intended to be simulated (e.g., micro-mobility within one room, macro-mobility between cells, etc.) and on time-management as well, such as slot duration for instance. Any biased distribution for speed and direction might lead to a different implementation of the model. Special attention is required regarding the MHs' density assumption, and initial distribution of MHs' positions [13, 54].

- *Restricted Random Model*: the restricted random model usually introduces some kind of auto-correlation or biasing in the "randomness" of uniform distribution of random model parameters [13]. An example can be given if the speed selection s or the direction θ can be updated up to a limited amount based on current values, e.g., $s \in [s-k, s+k]$, $\theta \in [\theta - \pi/4, \theta + \pi/4]$. This model defines a preferred direction and a preferred speed range for all users (or for every single user), and smoothed curves and accelerations. The main factors to define for analysis based on this model are the direction and speed ranges, and the admitted tolerance for variations.

- *Smooth Random Mobility Model*: it was proposed in [3, 5], and it can be seen as an extended Random Mobility Model. It is defined with two stochastic processes for correlated speed and direction management. In [28], a criticism to the Random Models used for MANETs' simulation was based on the unrealistic movement behavior caused by sudden and uncorrelated speed and direction changes. Restricted Random Models introduced auto-correlation. In the Smooth Random Model, correlation is introduced, together with a set of tunable parameters concerning "node-classes" characterized by acceleration and deceleration parameters, target speed, and smoothed direction changes. The proposed model is able to implement realistic behavior of nodes in many scenarios, from urban (Manhattan-like) to large-scale, with acceptable additional computation required [3].

- *Random Waypoint Model*: it is similar to Random Mobility Model, but it adds the *epoch* and *pause* concepts, to make the Random model a little bit more similar to realistic user mobility [4, 12]. A mobile host (MH) executes a sequence of epochs, each one defined as a motion interval followed by a pause interval. At the beginning of a motion interval, the MH selects the new destination coordinates (x, y) (not the direction as in the Random model) uniformly distributed in the simulated area. Any border policy is equivalent with this motion model, since MHs can only touch, and never hit, the area boundaries. Speed is uniformly distributed in $[minSpeed, maxSpeed]$. At the end of a motion interval, a given "pause time" pt is defined, uniformly distributed in $]0, maxPauseTime]$. If $pt = 0$ something really similar to the Random Mobility model is obtained. Intuitively, this motion model is the "after dinner" behavior of the "dining philosophers" (walk,think). This model is of widespread use in many simulations of wireless mobile systems [4, 12, 64]. All of the considerations regarding the Random mobility model are still valid, e.g., uncorrelated and unpredictable mobility, memoryless property for speed and direction [4]. The pitfall of MHs' concentration in the center of the simulated area is still present: this means that MHs are often moving towards the high-density center of the simulated area, and sometimes moves temporarily to the sparse boundary areas [5, 54]. Any initial distribution of MHs is not relevant for the steady-state distribution of MHs, because the next position is always a random point in the simulated area. Transient effects from the initial distribution of nodes can be fastly eliminated, in order to avoid biasing in the steady-state simulation results. Some problems can be given by the model factors: speed and pause time. Currently, Random Waypoint is subject to criticism [64], mainly for the speed distribution of nodes and for the risk of density concentration of MHs in the center of the simulated area [54]. Non trivial relationship between average speed and average pause time has been reported in many scenarios, depending on the objective of the simulation [13]. If the network stability and link reliability is under analysis, it would result that the average pause-time sometimes has a prevailing effect with respect to average speed of MHs [13].

- *Random Direction Model*: it is a small variation of Random Waypoint epochs, defined in order to avoid the MHs concentration in the center of the simulated area. Anyway, to obtain uniform number of neighbors (i.e. degree) for each MH, the modeler should be careful about the model parameters. The model is similar to Random Waypoint: before a motion period, a speed and a direction (like in Random model) is uniformly selected, to be maintained up to the area boundaries will be reached [54]. Once on the boundary, a pause time is selected, then a new epoch starts. Given the reduced density distribution, network partitions are more probable in this model [13]. Another variation is the *Modified Random Direction Model*, where the selected direction is followed up to a given distance d, without necessarily reaching the area boundaries. This model would be quite similar to a hybrid Random Mobility model with pause times, and to Random Waypoint.
- *Boundless Simulation Model*: is similar to a vector-based implementation of the Restricted Random Mobility model, implemented over a torus-like simulation area [13].
- *Gauss-Markov Mobility Model*: this model uses a tuning parameter $\alpha \in [0, 1]$ to vary the degree of "randomness" and self-correlation of speed and direction in a Random Mobility Model [13]. $\alpha = 0$ returns a Random Mobility Model, while $\alpha = 1$ returns a linear motion in the initial direction and speed [13].
- *Mobility Vector Model*: this model uses a Base Vector, a Deviation Vector, and an acceleration parameter α to define the Mobility Vector for every MH. Given the Mobility vector definition, the extension to 3-D space model is straight forward, and the vector model defined can be considered as a framework for many models' implementation [27].
- *City Section Mobility Model*: it is a hybrid model merging Random Waypoint Model and Manhattan-like scenarios. The urban constraints are defined as usual (streets, one-ways, crossing, walls, etc). Every MH randomly selects a destination, then it travels towards the destination following the most linear way. Once arrived to its destination, the MHs pauses for a random time, then it chooses another

destination [13]. The model may introduce some additional issues to be managed, like speed limits, traffic lights, and traffic laws. This may require a significant computation.

- *Graph-based Mobility Model*: this model has some similarities with the City Section model. Every MH moves following the edges of a graph defining the infrastructure of the area. The target destination is one of the graph vertex, randomly selected, and the way to follow is always the shortest path [60].
- *Random (Manhattan) Drunk Mobility Model*: is similar to the City Section Mobility Model, but it does not define a target point to reach. Every time a new crossing is reached, a new direction is selected between the available ones, according to any distribution probability. Speed can be changed as a separated stochastic process, or according to scenario constraints [3].

Among the synthetic mobility models, the *Group Mobility Models* belong to a new class of models which can be used for MANETs modeling and simulation purposes [13, 28, 61]. The main difference for such models is given by the idea that MHs' decisions about their movements would mainly depend upon other MHs in their group, or common factors in the scenario. This introduces a motion-correlation effect among MHs belonging to the same logical group. This effect should be evaluated as unacceptable if the assumption for the analysis requires uncorrelated mobility. It may happen, for example, that the relative mobility of MHs within the same group is really low, thereby favoring intra-group communication and routing. The analysis of a given routing protocol under this mobility model should not be considered as a generalized result for general scenarios, because it is biased in a significant way by the adopted mobility model correlation. Group partition and definition is out of the scope of this presentation: it may be related to position, host speed (walk, car, train), and scenario characteristics (e.g., highway lane). The Group Mobility Models can be roughly classified in Gravity Models, Location Dependent Models, Targeting Models and Random Group Mobility Models [13]:

- *Gravity Model*: this model can be used in scenarios where MHs may tend to move towards some destinations (e.g., signal sources), named *attraction points.* Intuitively, every MH is assigned a positive charge, while attraction points are assigned a negative charge. Opposite charges attract each other, while same charges repel each other. MHs with no charge have no gravity effects [13, 27].
- *Reference Point Group Mobility (RPGM) model*: this is the most general group mobility model. Specifically, Column model, Nomadic Community model and Pursue model can be implemented as special cases of the RPGM model [28]. A logical center for the group is defined, and each MH defines a reference point fixed with respect to the group's logical center. The logical center moves accordingly to a group's motion vector (GMV), randomly chosen or predefined, and every MH adds a random motion vector (RMV) to its reference point [28].
- *Reference Velocity Group Mobility (RVGM) model*: this model can be used when the group shares velocity and direction characteristics, rather than proximity [61]. A Group Velocity vector defines

the dominant velocity characteristic of the group, and a random Local Velocity deviation vector is composed with the Group Velocity to determine the single Host Velocity vector. This can be thought as the time derivative of the position-based group representation in the RPGM model [61].

- *Exponential Correlated Random Mobility (ECRM) model*: this model introduces a quite complex motion function that can be used to define the MH movements, where a parameter τ defines the mobility factor, and a random Gaussian variable with parameter σ is included in the formula [28]. The main problem with this model is to find appropriate values for the model parameters.
- *Column model*: this model defines a mobility pattern similar to a column of not well-trained soldiers marching in line. Every MH has a reference point in the column and moves randomly around that point. All reference points (i.e., the column) move together based on the common advance vector definition [13].
- *Nomadic Community model*: this model defines a mobility pattern similar to a group of students following a guided visit to a museum. It is a hybrid random/targeting group mobility model. The whole group of MHs (students) has a common single reference point (the guide) which is moving according to a given random mobility model (e.g., similar to Random Waypoint). Every single MH is free to move around the group's reference-point, according to a random mobility model [13].
- *Pursue model*: another targeting group mobility model. This model is based on the definition of a single target moving accordingly to a random mobility model. The tracking MHs define their group mobility based on the straight direction from their position to the target, biased by a random offset-vector.

Other complex and mathematically untractable motion models can be defined to capture more realistic user mobility patterns, to be used in simulation. This is an ongoing research activity. One of the challenges for the research is to find efficient techniques for the implementations of the proposed models. Additional efforts should be given in the study of models whose implementation can be supported efficiently in the adoption of the distributed simulation paradigm.

Many commercial and freely distributed simulation tools support mobility models and complex scenarios. Recently, some application tools have been proposed for known simulation tools and models. CAD-HOC [55] is a tool to generate mobility benchmarks and ad hoc scenarios to feed the network simulator ns-2 [45]. Bonn-Motion is a mobility, scenario-generation and analysis tool, written in Java, that can be used to define Tcl scripts feeding ns-2. FraSiMo [20] is a research project to model mobile ad hoc networks with Omnet++ [46]. A commercial tool, OPNET [47], defines a complete set of facilities to model complex mobility, scenarios and propagation models for ad hoc networks.

2.1.6 Traffic workload

The workload characterization for MANETs, i.e., the amount of data to be transmitted between MHs, is another relevant point for the modeling definition. Workload is relevant for the evaluation of the supported

Quality of Service (QoS) and service reliability for the application and user needs. The network traffic characterization is a problem that has been analyzed for years, dealing with self-similarity, bursty nature and correlation of packets arrival processes, etc. [56].

Trace-based workload models are widely used in many simulations, data and video transmission for instance. In MANETs, currently nobody knows what would be the killer application, so we can only speculate about the workload characterization of such systems. Usually, as a worst case scenario, the simulation analysis can be performed in *asymptotic workload conditions*: this means that the assumption for the system is that the sources of traffic in the network always have full transmission buffers. This hypothesis is good to test the stability and congestion reaction of a given network, or to evaluate the scalable behavior and asymptotical throughput metrics for the system[2].

Underload conditions can be defined by adopting other commonly used, parametrized models. Another widely adopted traffic-model for MANET simulation-analysis is the *Constant Bit Rate (CBR) traffic* model, where every source (sender) of traffic generates a constant flow of packets. This can be obtained simply by assuming that a given amount of data is generated at constant time intervals. This model is commonly used to approximate the workload generated by voice-based applications. This model can also be extended in many ways, in order to make it much more realistic.

The *Variable Bit Rate* (VBR) traffic model can be adopted to approximate the workload generated by data and video applications [41]. It is defined by traffic sources generating a variable amount of data, as a function of time, depending on many packet-interarrival distribution parameters.

2.2 Mobile ad hoc networks simulation

The computer-based discrete-event simulation is one of the most flexible methods for the performance evaluation of complex systems, such as MANETs. The goal of a simulation study is the construction of a simulator that mimics the system state transitions, and, (by collecting and analyzing data during simulation runs) estimate the performance metrics of the systems under analysis. An orthodox simulation study is based on several steps whose characteristics and number can vary with respect to the nature of the system analyzed and to the objectives of the study. The key steps establishing the kernel of any simulation study are: *i) problem formulation; ii) workload characterization; iii) model definition and validation; iv) construction and verification of the simulator; v) design of experiments; and vi) analysis of the simulation results or output analysis* [29].

In this section, we discuss the main system characteristics and performance figures of interest for mobile ad hoc network simulations. Regardless of the applications and the protocol layers considered for the analysis, many critical features contribute to determine the efficiency, reliability and effectiveness of MANETs. MANET networks are characterized by dynamic topologies, requiring adaptive, multihop routing protocols, dealing with bi-directional and uni-directional links [18]. Links are bandwidth constrained with respect to

---

[2] note that this is a worst-case scenario, and maybe an unrealistic condition.

typical wired networks, and they offer variable capacity and delay, due to the effect of highly variable scenario conditions. Mobile hosts are energy constrained, so MANETs privilege protocols dealing with energy reduction approaches, e.g., sleep-period management and adaptive power-reduction. Due to host mobility, MANETs' scalability is a major issue to solve. This problem is even more complicated by the distributed management and distributed protocols' implementation which are commonly adopted [24]. This makes it difficult to guarantee network behavior, reliability, fairness and efficiency under every condition. Reduction of overheads to maintain proper network functionality is a common problem: use of critical resources, like battery energy, buffer memory, local CPU computation and bandwidth for the transmission of control packets should be minimized.

2.2.1 Performance metrics

A large set of *performance metrics* could be defined to evaluate MANETs, in order to understand the critical features of the considered system. Some metrics can be considered relevant or significant only for a given protocol layer. Other metrics can be general, even if they may be affected by a chain of inter-layer implications. Performance metrics can be roughly divided into the following three categories: user performance metrics, resource utilization metrics, and system metrics. User performance metrics include, but are not limited to: latency, delay, quality of service and priorities, average and peak performance, reliability and cost-efficiency metrics. Resource utilization metrics include overheads, utilization, fairness and efficiency, just to mention a few. System metrics include scenario, stability, scalability and context metrics (e.g., topology changes, network partitions, cluster-life, mobility, density, load, path length, etc.). As an example, given a task-process evaluation, interesting metrics can be defined as: average power consumed and communication overheads (which are both considered as resource utilization metrics) and task completion time (i.e. a user-satisfaction parameter). Given a routing protocol evaluation, interesting metrics can be defined as: average end-to-end throughput, average end-to-end delay, average link utilization, average packet-loss probability, energy efficiency, protocol overheads, among other indices.

Now we will present a short list of generalized metrics that can be evaluated and adopted in the analysis of Medium Access Control, Routing and Transport protocols for MANETs [24, 18, 23]. In the analysis of the following metrics, mean values should be investigated together with variances or confidence intervals, and distribution percentiles.

- *Access delay*: the time spent by a frame (or a packet) in the MAC (routing or transport level) queue. It is defined from the instant the frame is enqueued till its transmission is successfully completed. Since delay depends on protocol definition and also on system load and traffic model, every comparison should be performed under the same conditions.

- *Channel Capacity*: is the maximum amount of data (e.g., bit rate C) that can be transmitted over a single channel. The nominal bit-rate can be reduced in presence of noise and interference. Coding techniques scale in the number of bit/symbol in order to contrast the noise, resulting in lower and lower bit-rates.

- *Throughput and Utilization*: the scope of any transmission protocol is to maximize the number of transmitted bits while minimizing the average access delay. Throughput T is defined as the average size S of a given frame (packet), divided by the corresponding average access delay D, i.e. $T = S/D$. This index is related to the Utilization index U, which can be defined as the fraction of channel capacity C used for successful data transmission.
- *Overheads*: every resource in the system that would not be strictly necessary to transmit the payload of the communication can be considered as an overhead and should be minimized (e.g., time, bandwidth, capacity, cpu-time, energy, money).
- *Fairness*: this is a concept related to service and resource sharing, rather than a performance index. A transmission protocol is fair if it does not show any preference for any single MH contending or waiting for resources or services. Fairness is the opposite of prioritized access and scheduling policies, adopted to support QoS and multimedia applications.
- *Stability*: a stable system should not have any fluctuating behavior resulting in a reduction of the average Throughput and Utilization. Adaptive protocols should be evaluated under the stability viewpoint. Many factors contribute to make the system unstable.
- *Reliability*: this concept defines a measure of the system reliability with respect to many failures that can be expected, e.g., network partition and broken paths. The reliability can be evaluated as a probability measure of failures, and as a measure of the failure-recovery delay.
- *Scalability*: a scalable system is obtained when protocols and management react and adapt in the opportune way to changes in the system factors like load and number of MHs. A scalable system is a system in which performance scales with no collapse. If a collapse is given, it would be interesting to find information about the *saturation point*, i.e. the limit the system can sustain, and the recovery time from saturation conditions. A typical example is given by congestion problems.
- *Power consumption*: most MHs are battery powered, and a maximum energy efficiency is required for every task performed, including system maintenance, transmission and reception of data.

3 Simulation techniques

In this section, we shall introduce the basic terminology and major issues pertaining to simulation techniques. Before, we proceed further, we must draw distinctions between different types of simulations: *continuous, discrete,* and *hybrid.*

Continuous simulation models the situation in which changes in state occur smoothly and continuously in time, e.g., the flow of liquid through a pipeline, weather modeling, and circuit level simulation of electronic components. Continuous simulation models often involve difference or differential equations that represent

```
While Not Empty (EventQueue) Do
        dequeue (m)                    /* earliest event from EventQueue */
        update (clock)
        simulate (m)
        enqueue()                      /* enqueue any events produced */
EndWhile
```

Figure 5: **Basic Sequential Discrete Event Simulation Algorithm**

certain aspects of the system. *Discrete* simulation refers to the modeling technique in which changes to the state of the model can occur only at countable points in time [21, 57]. For example, in logic simulation, the circuit is simulated by assuming that node voltages only take on values from a finite set (say, 0 and 1) and that transitions between values are instantaneous; in switch-level simulation, transistors are simulated as switches that can be either opened or closed. Digital computing systems, computer and communication systems, and queueing systems (such as bank teller and job shops) are other examples of discrete event systems. Many systems are *hybrid*, that is, combinations of discrete and continuous characteristics. An example of a hybrid system is an unloading dock where tankers queue up to unload their oil through a pipeline. The decision of whether to use a discrete or continuous model for a particular system depends on the specific objectives of the study. For example, a model of traffic flow on a freeway would be discrete if the characteristics and movement of individual cars were important. Alternatively, if the cars can be treated in the "aggregate", the flow of traffic can be described by differential equations in a continuous model.

In this chapter, we are interested into discrete systems which can be simulated by discrete-event simulations. In a discrete-event simulation the model evolution is defined by instantaneous *events*. Each event corresponds to a transition in a portion of the model state, composed of *state variables*, each describing a characteristic of the model. Each event also has a simulation time associated with it, called *timestamp*, which defines its occurrence time. Each event may in turn generate new future events.

The generation of new events and the dependency of their transitions on state variables that previous events may have updated, define a relation of *causal order* (a partial order) among events. Related events are said to be *causally dependent*, whereas unrelated ones are called *concurrent*. In order to guarantee the correctness of the simulation, concurrent events may be safely processed in any order in a simulation, whereas causally dependent events must be processed according to the causal order. Thus, to ensure the strict chronological order, events are processed one at a time, resulting in an (apparently) sequential program. A typical template for a sequential simulation is given in Figure 5.

Discrete systems can be simulated by discrete-event simulations. Many methods have been proposed in the literature for implementing discrete systems. They can be broadly classified into two groups, the *synchronous* and the *asynchronous* methods. In synchronous discrete event simulation, all objects in the simulation progress forward in simulation time together, in synchrony, with no object ahead in time of any other. The usual queue implementations for sequential simulation are all synchronous methods. In contrast, an asynchronous method permits some objects to simulate ahead in time while others lag behind. Of course, an asynchronous method must include some mechanism for ensuring that when an object that is "behind" schedules an event for execution by an object that is "ahead" it does not cause any events to be executed in the wrong order.

In this chapter, we are interested into modeling and simulation of wireless and mobile networks based upon asynchronous discrete event simulation tools.

3.1 Sequential network simulation testbeds

In this section, we shall review several network simulators that have been widely used by both academia and industry communities.

OPNET (Optimized Network Engineering Tool) [47] provides a comprehensive development environment for the specification, simulation and performance analysis of wired and wireless networks. It is based upon a sequential discrete event simulation paradigm using a hierarchical modeling structure, where each level of the hierarchy represent different aspects of the complete model that is being simulated. It provides powerful tools to assist users to build their simulation model, and for output analysis. OPNET has been used quite successfully within the wireless network community. Though, everybody agrees that OPNET does not scale quite well. To the best of our knowledge, scalability is a major problem in most of the commercial simulator tools. Our experiences with OPNET indicate that it does have the scalability problem, it takes too long to run the simulation, and one has to spend a large amount of time to understand how to use. Furthermore, it is a commercial product, and one has to pay a large amount of money to buy it.

Recently, a federated simulation approach has been investigated to enhance and parallelize OPNET. Each federate is basically a sequential simulator modeling a subnetwork of the simulated model. Although, recent results were quite encouraging, the work is still at an early stage [63].

INSANE, a network simulator, was designed to test various IP over-ATM algorithms with realistic traffic load derived from empirical traffic measurements. Although the simulator provides an easy approach to check the progress of multiple running simulation processes, we find it quite restrictive to ATM network simulations.

NetSim is another network simulator. It was designed to provide detailed simulation of the Ethernet, including realistic modeling of signal propagation, collision detection and handling process.

OMNeT++ [46] is a freely distributed, object-oriented, modular discrete event simulator written in C++. It is designed for general-purpose discrete event simulation, and provides some model libraries for communication protocols and network systems. NED, a network descriptor language, can be used to assist the modeler in the model definition based on system modules written in C++. OMNeT++ support for parallel execution and parallel discrete event simulation is an ongoing research activity.

The network simulator *ns-2* [45] is a discrete event simulator that provides substantial support for simulation of TCP, routing, and multicast protocols over wired, wireless (local and satellite) networks, and wireless multihop ad hoc networks. *ns-2* began as a variant of the REAL network simulator in 1989 and has evolved substantially over the past few years. Since then, it has included substantial contributions from other researchers, including wireless code from the UCB Daedelus and CMU Monarch projects and Sun Microsystems. *ns-2* is written in C++, and it uses OTcl, an object oriented version of tcl, as a command and configuration interface. The interface code to the OTcl interpreter is separate from the main simulator, and complex objects are decomposed into simpler components for greater flexibility and composability. Although, *ns-2* is widely in use within the wireless networking communities, it is not a fine tuned and finished product, and it is still a result of an on-going effort of research and development. In particular, bugs in the software are still being discovered and corrected. Users of *ns* are mainly responsible for verifying for themselves that their simulations are not invalidated by bugs.

Among all the existing network simulators, *ns-2* is most popular tool used by both wireless and wired communities. It has also been extended to mobile ad hoc networks as well. The major drawback of *ns-2* is the execution time of the simulation, mainly due to the sequential implementation of the discrete event simulator. While it is quite easy to use, run, or modify pre-existing models, it requires a large amount of time to study the inside of *ns-2* before a simulation modeler develops new models.

A few researchers have investigated ways to speed up the running time of the simulation using *ns-2*. We shall describe them in the next section.

3.2 Parallel and distributed simulation

Due to enormous computational requirements by a sequential simulator for complex wireless systems, parallel discrete event simulation techniques [48, 14, 39, 40, 65, 43] are often studied to reduce the execution time of the simulation models. Before, we proceed further, let us introduce the basic terminology and major issues pertaining to parallel and distributed simulation. A parallel or distributed simulation should provide the same solution to a problem as a sequential simulation.

3.2.1 Principles of parallel and distributed simulation

To ensure the strict chronological order in large scale simulation, events are processed one at a time, resulting in an (apparently) sequential program.

Only by eliminating the event list in its traditional form so as to capture the interdependence of the process being simulated can additional parallelism be obtained [16]. This is the objective of *parallel simulation*. Indeed, parallel simulation shows a great potential in terms of exploiting the inherent parallelism of the system, and the concurrency among events to achieve execution speedup. Good surveys of the literature may be found in [21].

A parallel simulator is composed of a set of *logical processes* (*LP*) which interact by means of messages, each carrying an event and its timestamp, thus called *event messages*. Each *LP* is responsible for managing a subset of the model state, called *local state.* Each event e received by an LP represents a transition in its local state. The events scheduled by the simulation of e are sent as event messages to neighboring LPs to be simulated accordingly. In a simulation, events must always be executed in increasing order. Anomalous behavior might then result if an event is incorrectly simulated earlier in real time and affects state variables used by subsequent events. In the physical model this would represent a situation in which future events could influence the present. This is referred as *causality error.* Several synchronization protocols have been proposed to deal with this problem. These techniques can be classified into two groups: *conservative* and *optimistic.* While conservative synchronization techniques rely on *blocking* to avoid violation of dependence constraints, *optimistic* methods rely on detecting synchronization errors at run-time and on recovery using a *rollback* mechanism.

3.2.2 Conservative simulation

Conservative approaches enforce event causality by requiring that each logical process (LP) elaborates an event only if it is certain that it will not receive an earlier event. Consequently, events are always executed in chronological order at any LP. Each logical process LP_i maintains an input queue (l_{ij}) for each of its neighbor LP_j. In the case that one or more (input) queues are empty, the LP is blocked because an event with a smaller time stamp than the time stamp of the waiting events might yet arrive at an empty queue. This mechanism implies that only unblocked LPs can execute in parallel. If all the LPs were blocked, the simulation would be deadlocked. Ensuring synchronization and avoiding deadlocks are the central problems in the conservative approach. Several schemes have been proposed to alleviate this problem. In [16], the authors employ null messages in order to avoid deadlocks and to increase the parallelism of the simulation. When an event is sent on an output link a null message bearing the same time stamp as the event message is sent on all other output links. As is well known, it is possible to generate an inordinate number of null messages under this scheme, nullifying any performance gain [21].

As a result, a number of attempts to optimize this basic scheme have appeared in the literature. For example, in [57], the authors refrain from sending null messages until such time as the LP becomes blocked. They refer to this approach as *eager* events, *lazy null* messages. They reported some success in using variations of Chandy-Misra approaches to speed up logic simulation.

Boukerche and Tropper [10] employed the following approach. In the event that a null message is queued at an LP and a subsequent message (either null or event) arrives on the same channel, they overwrite the (old) null message with the new message. A Single buffer is associated with each input channel at an LP to store null messages, thereby saving space as well as the time required to perform the queuing and de-queuing operations associated with null messages. Good surveys of conservative techniques might be found in [11, 21].

3.2.3 Optimistic approach

Time Warp is based on an optimistic approach and enforces the causal order among events as follows: events are greedily simulated in timestamp order until no event messages remain or until a message arrives in the "past" (a straggler). Upon receiving a straggler, the process execution is interrupted, and a *rollback* action takes place using *anti-messages.* Each message is given a sign; positive messages indicate ordinary events, whereas negative messages indicate the need to *retract* any corresponding event that was previously processed. Similar messages that have different signs are called anti-messages. If a negative message is received, the message and the corresponding anti-message are both *annihilated.* A rollback consists of the following three phases: *i*) *restoration*: the latest state (with respect to simulation time) valid before the straggler's timestamp replaces the current state, and successive states are discarded from the state queue; *ii*) *cancellation*: the negative copies of messages which were produced at simulation times successive to the straggler's timestamp are sent to the proper processes, to possibly activate rollbacks there; and *iii*) *coasting-forward*: the effective state which is valid at the straggler's timestamp is computed by starting from the restored state and by elaborating those messages with a timestamp up to the stragglers; during this phase no message is produced. Rollbacks are made possible by means of *state checkpointing.* The whole state of the process is checkpointed into the *state queue* according to some discipline [49].

To minimize the storage overhead required to perform rollbacks, and to detect the termination of LPs, optimistic synchronization mechanism uses a *local virtual time*, (LVT), and a *global virtual time*, (GVT). LVT represents the timestamp of the latest processed event at an *LP*; whereas GVT is defined as the minimum of all the local virtual times of all *LP*s, and of all the timestamps of messages in transit within the simulation model. GVT indicates the minimum simulation time at which a causal violation may occur. GVT computation is used to *commit* the safe portion of the simulation.

The optimistic scheme is preferred when a system to be simulated contains high predictability of events so that rollbacks are kept to a minimum. Thus, to improve the PCS network simulation, we use a hybrid approach whit both conservative and optimistic schemes.

3.3 Wireless network simulators based upon PDES

Several simulation techniques have been proposed in the literature [14, 38, 39, 43, 48, 65] to speedup the execution of simulation of large-scale wireless networks. In this section, we shall describe them, and discuss their main features.

ns-2 has long been considered to be a *de facto* standard simulator for wireless and wired networking protocols research. The networking community has long been resisting to rewrite the network simulator, or use different platforms. Therefore, some researchers have tried to parallelize the *ns-2* using established parallel and distributed simulation techniques, thereby providing a transparent parallel execution of *ns-2*. G. Riley and R. Fujimoto have proposed a distributed version of the popular network simulator, *ns-2*, which they refer to as parallel and distributed *ns*, or simply PDNS [50]. Their goal is to use the existing network simulator and minimize the changes to it while allowing their parallel simulator to take advantage of their proposed and new version of *ns-2*. They revised the *ns-2* syntax by adding a set of directives that are directly related to the parallelization of the simulation. Their idea is based upon the federated simulation approach where separate subnetworks of the simulated model are executed on different processors connected either via Myrinet network, or a standard Ethernet network using the TCP/IP protocol stack. A set of library, which they refer to as Georgia Tech RTI Kit [25], is used for synchronization purposes. Conservative methods for synchronizing *ns-2* processes have been implemented. The RTI Kit is a software implementation of the Run-Time Infrastructure of the Department of Defense's High-Level Architecture (HLA) for large-scale distributed simulations [19]. Much of the improvement obtained in their design was obtained from parallelization of the set up of the simulation and not the actual execution of the simulation.

Another project at the University of Cincinnati [33, 17] involved running *ns-2* in parallel. The main objectives of their work is to build a space-time parallel simulator to study how effectively ad hoc network simulations can be performed in parallel. At the present time, their testbed supports parallel execution of *ns-2* programs consisting of point-to-point links with static routing and UDP traffic. Conservative null-messages approach has been used for synchronization purposes. Although initial results are encouraging, this work is still at an early stage.

PDES community has also tried to design efficient simulator for wireless and mobile systems using PDES synchronization schemes without relying on pre-existing network simulators. Wireless propagation and protocol testbed (Wippet) [48] is a versatile simulator for wireless networks. It consists of basic set of modules implemented using the TeD, an objected oriented and telecommunication descriptive language for parallel simulation of telecommunications developed at Georgia Tech. [51]. Its propagation and interference modeling at the receiver MH made simulation suitable for studying dynamic channel allocation schemes. The partitioning of the model into multiple zones is done either geographical based or channel based. Channel based partitioning gives rise to better speedup due to rare synchronization of zones where a mobile changes the channel. However, how that is achieved in the implementation of Wippet is unclear. Selection of other channels requires interference measurement on the destination channel which should induce overall synchronization on all zones.

The *GloMoSim* (Global Mobile Information System Simulator) is a library-based sequential and parallel simulator for wireless networks, including multihop wireless ad hoc networking and traditional wired Internet connectivity [65, 58]. The *GloMoSim* is designed as a set of library modules, each of which simulates a specific wireless communication protocol within the protocol stack. Modules of the protocol can be developed at different levels of granularity. It has been developed using PARSEC (PARAllel Simulation Environment for Complex Systems), a C-Based parallel simulation language [1]. PARSEC basically adopts a message-based approach to discrete event simulation where physical processes are modeled by simulation objects referred to as entities, and events which represent the transmission of time-stamped messages among corresponding entities. Glomosim has been designed so that it can be easily extended and new protocols and modules can be added to this library using this PARSEC language. It has been implemented on both shared and distributed memory machines, and it supports conservative layered simulation in the context of wireless network simulation. The synchronization protocol makes use of Chandy-Misra Null-Messages scheme [16, 57]. Although the results reported in [65] show a significant reduction of the null-messages overhead, a speedup of only up to 3.5 was obtained using 8 processors [65]. Low speedups shadow the improvement due to unresolved causal dependencies. More recently, the authors have reported improvement on conservative simulation due to better lookahead computation [43].

Both GloMoSim/Parsec and TeD/GTW systems require the simulation modelers to learn new language extensions to describe their network models. Although, GloMoSim, as opposed to TeD/GTW, was designed to make the mechanics of parallel simulation transparent to protocol modelers by embedding them into the lowest (channel) layer. Further knowledge of PDES synchronizations is needed to understand how they work, in order to develop new models, unless users are interested to run or modify pre-existing models.

QualNet is basically a commercial product derived from GloMoSim, developed at UCLA. It is designed by the Scalable Network Technologies Inc. headed by R. Bagrodia from UCLA. Several extensions have been added to GloMoSim to facilitate the development of new protocols for wired, wireless, and ad hoc networks.

The following summarizes other related work on parallel simulation of wireless networks. An optimistic model based on Time Warp is proposed in [14], which uses logical processes (LPs) and uniform rectangular shaped cells to simulate large-scale PCS networks. The mobility of a MH is limited to four neighbors only, and low blocking probability is achieved with a fixed ratio of 50 MHs per cell. Better results were obtained with a low number of mobile hosts per cell. In [39], another optimistic parallel simulation is presented in which the PCS coverage area is modeled by fixed hexagonal shaped cells identifying the LP. The MHs are given constant speed and angle of movement. Although the obtained results are encouraging, yet the model is useful only for a low call traffic and reduced mobility.

As opposed to the preceding two cell-based partitioning, a channel-based partitioning is proposed in [40]. In this method, when a MH makes a hand-off, a set of messages is sent out to all channels available on the new BS. This scheme may generate an inordinate number of messages nullifying any performance gain. Mobility of MHs is limited to constant speed and four directions – North, East, West, South. The MH is disposed after the call is terminated. A good analysis of break-even points between cell based and channel based partitioning were reported.

Despite the fact that promising results were obtained in these approaches, most of them ignored real life patterns for mobility and PCS network deployment by restricting cell shapes to uniform geometric objects such as hexagons, rectangles or squares. These limitations and weak spatial modeling of cell characteristics often simplify the simulation model, and do not capture the accuracy and realism in PCS networks performance evaluation. Linear movements have been used in some of the existing works [14, 39, 40, 43]. In real transportation traffic flow, segmented movement patterns, occasional pauses and, most importantly, rush hour traffic and/or congested roads, trigger spikes in the call arrival rate. The results reported in [14, 39] assumed a (fixed) ratio of MHs to channels per cell which is unrealistic. Furthermore, channel-based partitioning [40] creates an MH at runtime and discards it after the call is terminated, thus losing mobility within calls and requiring unrealistic call arrival processes which may be unrelated to mobility patterns.

In [6], Bononi et. al. have recently defined a prototype General Adaptive Interaction Architecture (GAIA) middleware to be implemented over a conservative, HLA-based, distributed simulation of mobile systems. The aim of the proposed middleware is to provide adaptive, runtime allocation of model components over the set of federates executed on the available set of execution units. The adaptive allocation is performed in order to balance the need for parallel execution and the message-passing overhead of distributed simulation. The leading assumption for this work is that mobility inside the simulation model maps on dynamic changes in the area of influence of every simulated host. If a certain amount of time-locality is present in the communication with the neighbor hosts, then adaptive allocation can reduce the amount of inter-federate synchronization-message overheads. Preliminary results shown that speedup can be obtained in HLA-based, conservative simulation of mobile ad hoc networks, executed over networked clusters of personal computers.

Recently, Boukerche et. al. [7] have developed, *SWiMNET*, a high performance simulator for wireless and mobile networks. Their scheme uses a hybrid approach to simulating wireless and mobile networks, based on a combination of optimistic and conservative techniques. It exploits event precomputation due to a simple assumption: *mobility and call arrival events of MHs are independent from the state of the wireless PCS simulation for instance.* Thus, all events for each MH can be precomputed assuming all channel requests are satisfied, and the actual channel allocation simulation cancels events for blocked calls. An exception to this fact may be a *hot spot*, e.g., a congestion due to rush-hour traffic or at a traffic junction. In this situation the MHs in that region have very low, if any, mobility and tend to make more calls. This is tackled in the mobility design by introducing hot spot areas where speeds are reduced and call arrival rates are increased.

In this mechanism, all movement and call-related events for each MH are precomputed assuming all channel requests are satisfied. The small portion of events to be retracted due to blocked calls is later computed in the actual simulation. The low percentage of blocked calls desirable for wireless networks is exploited by the optimistic portion. Event cancellations are done only if a call is blocked or dropped. The precomputation can be pipelined to the channel allocation simulation, thus minimizing the overhead of generating events.

In Table 1, we summarize a comparison of model-related and simulation-related issues for SWiMNet, Wippet and GloMoSim.

In what follows, we shall describe the main features of SWiMNet, a scalable simulation testbed for wireless and mobile networks recently developed.

3.3.1 Description of *SWiMNet* model components

In SWiMNet, the entire simulation model is the result of the composition of four model components: (i) mobility models, (ii) call process, (iii) BS deployment, and (iv) channel management scheme. While the first

Table 1: **Comparison of Model and Simulation Issues**

Model-Related Issues			
Parameters	SWiMNet	Wippet	GloMoSim
Mobility	Segmented paths	Manhattan style	Unspecified
Call arrivals	Poisson process per MH	Model-wise poisson process	Node-wise poisson process
Coverage map	Irregular cells over Voronoi diagrams	Manhattan style urban environment	Uniform geometry (hexagons or squares)
Signal propagation	Not employed	Stochastic fading	Free-space model
Call admission	FCA	RSSI based DCA	Unspecified
Handoff mechanism	Cell crossing induced	RSSI based	Cell crossing induced
Model size	54 BSs, 10000 MHs	48 BSs	2000 nodes (short range)
Simulation-Related Issues			
Precomputation	Mobility, calls	NA	NA
Synchronization	Hybrid	Optimistic	Conservative
Partitioning	Cell Based	Zone based (channel/cell)	Static node based
Call traffic	4 calls/MH/hr	6 calls/sec to system	1 pkt/sec to each node
Speedup	11.8 on 16 processors	4 on 8 processors	6 on 16 processors

three model components are independent of each other, the channel management component is dependent of the compound result of the first three components. Mobilities and calls are represented by independent and stochastic processes[3]: locations of MHs are chosen pseudo-randomly, MHs trajectories across the map are sequences of pseudo-randomly generated segmented movements, and call inter-arrivals and durations are pseudo-randomly distributed.

As part of the mobility model, the population of mobile hosts (MHs) are classified into groups of *workers, wanderers, travelers* or *static users*, so as to represent behavior of different users' across the wireless coverage area. The number of MHs per class is arbitrary. Movements are modeled such that a complete path is composed of any number of straight segments. This allows almost any kind of movement, by approximating a curve line with as many segments as required by the resolution considerations. Every segment is then logically partitioned into unitary tracts of a given unitary resolution, which defines how finely the MH position is checked.

The call model is specified by means of a maximum call rate per hour per MH, and an average call duration. The entire simulation time interval can be partitioned into any number of sub-intervals, each with a different call rate. Thus, it is possible to represent call rate changes during the simulation; night hours may be represented with very low call rates, whereas office hours with high call rates.

By composing mobility, calls, and BS deployment, the precomputation stage (Stage 1) is able to generate one stream of possible events per MH. The destination of such events within Stage 2 is precomputed as well. The actual set of possible events, their correlations, and how they are simulated, depends on the channel management policy to simulate.

---

[3] Note that a discrete distribution with one element only corresponds to a deterministic behavior.

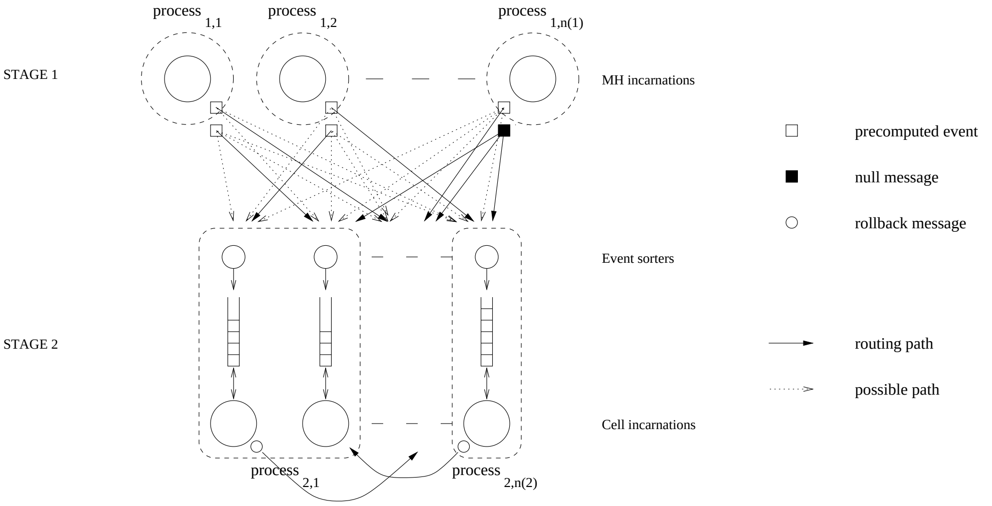


Figure 6: **Logical Interconnection between Levels of the Parallel Simulator**

The general structure of the simulator consists of three logical levels and two physical stages. Entities comprising the logical Level 1 are organized into the Stage 1 of n_1 (container) processes, where each process maintains a set of mobile host incarnations (all sets are disjoint). Stage 2 consists of n_2 cell container processes, and implements the Level 2 (event sorters) and Level 3 (cell incarnations) of the logical structure. The event sorter and the cell incarnation related to the same cell are managed by the same cell container process in Stage 2. Therefore, communications between Level 2 and Level 3 are easier and faster by means of direct memory access instead of message passing. Communications between Stage 1 and Stage 2 are implemented by means of Message Passing Interface (MPI) using the LAM 6.1 environment [44]. Since no feedback is necessary from Stage 2 to Stage 1, in principle the execution of the two stages may be performed at different times. However, that would require Stage 1 to store precomputed events on file and Stage 2 to read them from file afterwards, thus adding overhead.

The structure of *SWiMNet* simulator is depicted in Figure 6. For simplicity, every process is represented as composed of only one entity (i.e. MH/Cell objects) per level. Communications between Stage 1 and Stage 2 are based on a conservative scheme using null-messages paradigm, whereas communications within Stage 2 are based on an optimistic scheme [30].

The implementation is based on an object-oriented methodology using C++ as the programming language, which makes it easy to maintain and flexible to any changes. Every MH object is constructed as a base class with parameters that are generalized to all mobile hosts. The four classes of mobiles (workers, static users, wanderers, and travelers) are derived from the base MH class. MH objects are created in Stage 1 container processes while cell objects are created in Stage 2 container processes.

Three levels in the *SWiMNet* simulator have been defined, in which logical activities of objects are elaborated.

- Movement precomputation (Level 1) is composed of N_{mh} *mobile host incarnations.*
- Event sorting (Level 2) is composed of N_{cells} *event sorters.*
- Channel allocation simulation (Level 3) is composed of N_{cells} *cell incarnations.*

The communications within and between the two stages are shown in Fig. 6. Every MH incarnation process generates events for each MH it maintains. Then, those events are sent to the event sorter process (ES) for the cell where the event takes place. The MH incarnations are independent of each other. Thus, activities of Level 1 are completely parallelizable. Similarly, event sorters are independent of each other. However, cell incarnations, where channel management simulation takes place, are mutually dependent.

In SWiMNet, the optimism lies in the following: every time a move-in event is simulated at any cell incarnation, the latter optimistically assumes that the call is still on, unless it already received notification that the call was blocked by means of a *call blocked* message. In the case that a late notification is received, that is a call blocked message is received after the corresponding move-in event has been simulated, a rollback is performed that retracts and corrects the simulations which follow and include the move-in event. A call blocked message is sent by any cell incarnation whenever a move-out event is simulated, if the simulation of the corresponding channel request event in the event couple did not actually result in a channel to be allocated. This implies that from Level 1, it is necessary to keep track of at least that event which follows every move-out event in the sequence of events for the same mobile host. This information may then be used to construct a call blocked message.

Rolling back the computation might result in the need for *retracting* an incorrectly sent call blocked message by means of a *call allocated* message. For instance, let us assume that a call was notified as blocked because of channel unavailability after a channel was allocated to another call. If the allocated call turns out to be already blocked in a previous cell, then the call which was notified as blocked can indeed be allocated. A call blocked and a call allocated message relative to the same call correspond to a pair of anti-messages in *Time Warp* simulations [30, 31], hence a *sign* can be associated to these messages. In a Time Warp simulator, a call blocked message has a positive sign whereas a call allocated message has a negative sign.

Due to the optimistic assumption, the elaboration of a simulation message whose corresponding precomputed event was already elaborated always causes a rollback. However, the way in which data are stored in the simulator allows rollbacks to be optimized, i.e., only a portion of precomputed events need to be involved in the rollback, and such a portion is exactly computed without the need to inspect any additional event.

The simulator is automatically generated by the Master process which reads the description file of the simulation model, and partitions the model into two stages. The description file includes: the mobility model, the call arrival model, the cellular system map, the system architecture, the experiment seed and the simulation time of termination. The main tasks of the Master are: (i) generating parameters for every single logical entity, i.e., positions, speeds, movement times, etc. for each mobile host incarnation, according to the general description of the mobility model; and (ii) deciding the number of processes per stage, and mapping the processes to processors. Process mapping is an important factor in improving the efficiency of parallel simulation protocols. Currently, a simple static mapping discipline has been adopted: given a number of processors allocated to the simulation, half of the processors are used for Stage 1 and half for Stage 2.

Performance analysis of SWiMNet applied to wireless and mobile system can be found in [7, 9]. Further work is under study to evaluate its performance for mobile ad hoc networks.

4 Conclusion

This chapter focuses on several challenging design and modeling aspects of wireless, mobile and ad hoc networks. We presented a discussion of modeling issues related to physical transmission and interference, topology, mobility, workload and performance figures for mobile ad hoc networks simulation.

Modeling and simulation are traditional methods to evaluate large-scale wireless and multihop network designs. However, modeling is often intractable with today's large and complex mobility and traffic patterns in wireless and multihop systems. Thus, researchers have turned increasingly to use simulation studies of these systems. Though, detailed simulations of large scale, wireless, mobile and ad hoc networks require enormous execution time and large amount of memory due to the complexity involved in the simulation and mobility models. Even on high performance workstations, the execution time is in the order of days and memory requirements in the order of gigabytes which impose restrictions on the type of systems that can be simulated.

Parallel and Distributed Simulation (PDES) could be exploited to overcome these problems.

In this chapter, both sequential and parallel simulation tools for wireless mobile and ad hoc networks have been reviewed. We have also presented some recent examples of simulation methodologies to improve the simulation run-time of these networks using PDES techniques.

References

[1] R. Bagrodia, R. Meyer, et. al.,"PARSEC: a Parallel Simulation Environment for Complex Systems", UCLA Technical report, 1997.

[2] H. Bertoni, "Radio propagation for Modern Wireless Systems", Upper Sadle River, NJ, Prentice Hall, 2000

[3] C. Bettstetter, "Smooth is Better than Sharp: a Random Mobility Model for Simulation of Wireless Networks", Proc. of ACM Intern. Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM'01), Rome, Italy, July 2001.

[4] C. Bettstetter, H. Hartenstein, X. Prez-Costa, "Stochastic properties of the random waypoint mobility model: epoch length, direction distribution and cell-change rate", Proc. of the 5th ACM int. workshop MSWiM2002, Sept. 2002

[5] C. Bettstetter, "Mobility Modeling in Wireless Networks: Categorization, Smooth Movement, and Border Effects", Mobile Computing and Communications Review, Vol. 5, No. 3, July 2001

[6] L. Bononi, G. D'Angelo, L. Donatiello "HLA-based Adaptive Distributed Simulation of Wireless Mobile Systems", in Proceedings of IEEE/ACM International Workshop on Parallel and Distributed Systems (PADS'03), San Diego, CA, June 2003

[7] A. Boukerche, S. K. Das, and A. Fabbri "SWiMNet: A Scalable Parallel Simulation Testbed for Wireless and Mobile Networks", *ACM/Kluwer Wireless Networks*, Vol. 7, pp. 467-486, 2001.

[8] A. Boukerche, A. Fabbri "Partitioning PCS Networks for Distributed Simulation", *IEEE High Performance Computing (HiPC)*, 2000 LNCS 1970, Springer, pp. 449-458.

[9] A. Boukerche, S.K. Das, A. Fabbri, and O. Yildiz, " Exploiting Model Independence for PCS Network Simulation", *ACM/IEEE Parallel and Distributed Simulation (PADS'99)*, pp. 166-173, Atlanta, USA.

[10] A. Boukerche, C. Tropper,"Parallel Simulation on the Hypercube Multiprocessor", Distributed Computing, Spring Verlag, 1993.

[11] A. Boukerche,"Time Management in Parallel Simulation", High Performance Cluster Computing, Vo.2., by B. Rajkumar, Prentice Hall 1999.

[12] J. Broch, D.A. Maltz, D.B. Johnson, Y.-C. Hu, J. Jetcheva, "A performance comparison of multihop wireless ad hoc network routing protocols", Proc. of MobiCOM'99, Dallas TX, Oct. 1998, also in http://www.monarch.cs.cmu.edu

[13] T. Camp, J. Boleng, V. Davies, "A Survey of Mobility Models for Ad Hoc Network Research", Wireless Communications and Mobile Computing (WCMC), Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications, 2002

[14] C. Carothers, R. Fujimoto, Y.-B. Lin, P. England, "Distributed Simulation of Large-scale PCS Networks", *Proceedings of the 2nd International Workshop on Modeling, Analysis, and Simulation of Computer and Telecommunication systems*, (February 1994)

[15] D. Cavin, Y. Sasson, A. Schiper, "On the accuracy of MANET simulators", *Proceedings of POMC'02, Toulose, France*, (October 2002)

[16] K.M. Chandy, J. Misra, "Distributed Simulation: A Case Study in Design and Verification of Distributed Programs", *IEEE Transactions on Software Engineering*, Vol.SE-5 (September 1979), 440-452

[17] S. Das and K. Jones, "MASCOTS 2001"

[18] S. Corson, J. Macker, "Mobile Ad Hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations", RFC 2501, Jan. 1999

[19] DMSO: Defence Modeling and Simulation Office (1998), High Level Architecture RTI Interface Specification, Version 1.3, 1998

[20] FraSiMo, Framework for Simulation of Mobility in OMNET++, TKN Berlin, see http://www-tkn.ee.tu-berlin.de/research/research_texte/framework.html

[21] R.M. Fujimoto, "Parallel Discrete Event Simulation", *Communications of the ACM*, Vol.33, No.10 (October 1990), 30-53

[22] R.M. Fujimoto, Parallel and Distributed Simulation, Wiley&Sons, 2000.

[23] Gerla M., Tang K., Bagrodia R. (1999), "TCP Performance in Wireless Multi-hop Networks", Proceedings of IEEE WMCSA'99, New Orleans, LA, Feb. 1999.

[24] A.C. Chandra, V. Gummalla, J.O. Limb, "Wireless Medium Access Control Protocols", IEEE Communications Surveys and Tutorials, 2000

[25] Georgia Tech. RTI KIT, see http://www.cc.gatech.edu/computing/pads/fdk.html

[26] J. Heidemann, N. Bulusu, J. Elson, C. Intanagonwiwat, K.-C. Lan, Y. Xu, W. Ye, D. Estrin, R. Govindan, "Effects of Detail in Wireless Network Simulation", in Proc. of the SCS Multiconference on Distributed Simulation, Phoenix, AZ, Jan. 2001

[27] X. Hong, T.J. Kwon, M. Gerla, D.L. Gu, G. Pei, "A Mobility Framework for Ad Hoc Wireless Networks", Lecture Notes in Computer Science, 2001

[28] X. Hong, M. Gerla, G. Pei, C.-C. Chiang, "A Group Mobility Model for Ad Hoc Wireless Networks", Proc. ACM MSWiM'99, Seattle, 1999

[29] R. Jain, The Art of Computer Systems Performance Evaluation, Wiley, New York, 1991

[30] D.R. Jefferson, "Virtual Time", *ACM Transactions on Programming Languages and Systems*, Vol.7, No.3 (July 1985), 404-425.

[31] D.R. Jefferson, H. Sowizral, "Fast Concurrent Simulation Using the Time Warp Mechanism", *SCS Multiconference on Distributed Simulation* (1985), 63-69.

[32] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, M. Degermark, "Scenario-based Performance Analysis of Routing Protocols for Mobile Ad Hoc Networks", Proc. of MobiCOM'99, pp. 195-206, Seattle, 1999

[33] K. Jones, "Parallel and Distributed Simulation Techniques and Applications", PhD Proposal Nov. 2001, Univ. of Cincinnatti.

[34] T.S. Kim, J.K. Kwon, D.K. Sung, "Mobility and Traffic Analysis in three-dimensional high-rise building environments", IEEE Trans. on Vehicular Technology, 49(5):1633-1640, May 2000.

[35] Y.Y. Kim, S.Q. Li, “Modeling Multipath Fading Channel Dynamics for Packet Data Performance Analysis”, Wireless Networks, Vol.6, Dec. 2000

[36] C.Y. Lee William, Mobile Cellular Telecommunications: Analog and Digital Systems (McGraw-Hill, Inc., 1989).

[37] M.C. Little, D.L. McCue, “Construction and Use of a Simulation Package in C++”, Computing Science Technical Report, University of Newcastle upon Tyne, Number 437 (July 1993) (also appeared in the C User’s Journal Vol. 12 Number 3, March 1994).

[38] M. Liljenstam, R. Ayani, “A Model for Parallel Simulation of Mobile Telecommunication Systems”, *Proceedings of the 4th International Workshop on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*, San Jose, CA (1996)

[39] Y.-B. Lin, P. Fishwick, “Asynchronous Parallel Discrete Event Simulation”, *IEEE transactions on Systems and Cybernatics* (1995).

[40] M. Liljenstam, R. Ronngren, and R. Ayani, “Partitioning WCN Models for Parallel Simulation of Radio Ressource management”, pp. 307-324, ACM/Kluwer Wireless Networks, Vl. 7, N. 3, 2001.

[41] D.M. Lucantoni, M.F. Neuts, A.R. Reibman, “Method for Performance Evaluation of VBR Video Traffic Models”, IEEE/ACM Transactions on Networking, Vol.2, Issue 2, Apr. 1994

[42] J.G. Markoulidakis, G.L. Lyberopoulos, D.F. Tsirkas, E.D. Sykas, “Mobility Modeling in Third Generation Mobile Telecommunication Systems”, IEEE Personal Communications, pp.41-56, Aug. 1997

[43] R.A. Meyer, R.L. Bagrodia, “Improving Lookahead in Parallel Wireless Network Simulation”, *Proceedings of 6th International Workshop on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, Montreal, Canada (July 1998), 262-267.

[44] MPI Primer / Developing with LAM, Ohio Supercomputer Center (The Ohio State University, 1996).

[45] NS-2 simulation tool, see http://www.isi.edu/nsnam/ns/
NS-2 mobility extension from Rice Monarch, see http://www.monarch.cs.rice.edu/cmu-ns.html

[46] A. Varga, OMNET++, in the column ”Software Tools for Networking”, IEEE Network Interactive. July 2002, Vol.16 No.4, also in http://whale.hit.bme.hu/omnetpp/

[47] OPNET simulation tool, see http://www.mil3.com/home.html

[48] J. Panchal, O. Kelly, J. Lai, N. Mandayam, A. Ogielski, R. Yates, “Wippet, A Virtual Testbed for Parallel Simulations of Wireless Networks”, *PADS 98*, Banff, Canada (June 1998).

[49] A. C. Palaniswamy, P. A. Wilsey, “An Analytical Comparison of Periodic Checkpointing and Incremental State Saving”, *Proc. of the 1993 Workshop on Parallel and Distributed Simulation*, 127-134.

[50] Parallel and Distributed Network Simulator, PDNS, see http://www.cc.gatech.edu/computing/compass/pdns/

[51] K. Perumalla, R. Fujimoto, and A. Ogielski, “A TED: A Language for Modeling Telecommunication Networks”, ACM SIGMETRICS Performance Evaluation Review, 25, 4, March, 1998.

[52] B. Ramamurthi, D.J. Goodman, A. Saleh, “Perfect capture for local radio communications”, IEEE JSAC, vol. SAC-5, no.5, June 1987

[53] T.S. Rappaport, Wireless Communications: Principles and Practice, Second Edition, Prentice Hall, Upper Sadle River, NJ, 2002

[54] E.M. Royer, P.M. Melliar-Smith, L.E. Moser “An Analysis of the optimum node density for ad hoc mobile networks”, proc. IEEE International Conference on Communications (ICC), Helsinki, June 2001

[55] S. Shah, E. Hernandez, A. Helal, “CAD-HOC: A CAD Like Tool for Generating Mobility Benchmarks in Ad-Hoc Networks” Proceedings of SAINT’02, Nara, Japan, Feb. 2002

[56] W. Stallings, Wireless Communications and Networks, Prentice Hall, 2001

[57] Su, W. K., and Seitz, C. L., “Variants of the Chandy-Misra-Bryant Distributed Discrete Event Simulation Algorithm”, *Proc. of the SCS Multiconf. on Distributed Simulation*, Vol. 21, No. 2., 1989.

[58] M. Takai, R. Bagrodia, K. Tang, and M. Gerla, “Efficient Wireless Network Simulations with detailled Propagation Models”, ACM/Kluwer Wireles Network, pp. 283-306, Vol. 7, N. 3, 2001.

[59] M. Takai, J. Martin, R. Bagrodia, “Effects of Wireless Physical Layer Modeling in Mobile Ad Hoc Networks”, in Proc. of MobiHOC’01, Long Beach, CA, October 2001.

[60] J. Tian, J. Hahner, C. Becker, I. Stepanov, K. Rothermel, “Graph-based Mobility Model for Mobile Ad Hoc Network Simulation”, proc. of 35th Annual Simulation Symposium, San Diego, CA, April 2002.

[61] K.H. Wang, B. Li, “Group Mobility and Partition Prediction in Wireless Ad Hoc Networks”, proc. of IEEE Int. Conf. on Communications (ICC’02), NY, Apr. 2002

[62] A. Willig, “A New Class of Packet and Bit Level Models for Wireless Channels”, Proc. of 13th IEEE Intl. Symp. on Personal. Indoor and Mobile Radio Comm., Lisboa, Portugal, 2002

[63] Wu, H., and R. Fujimoto, “Experiences Parallelizing a Commercial Network Simulator”, Proc. of the 2001 Winter Simulation Conference, pp. 1353-1360.

[64] J. Yoon, M. Liu, B. Noble “Random Waypoint Considered Harmful”, to appear in Proc. InfoCom 2003.

[65] X. Zeng, R. Bagrodia, “GloMoSim: A Library for the Parallel Simulation of Large Wireless Networks”, *Proceedings of the 12th Workshop on Parallel and Distributed Simulation*, Calgary, Canada (June 1998).

[66] M.M. Zonoozi, P. Dassanayake, “User Mobility Modeling and Characterization of Mobility Patterns”, IEEE Journal on Selected Areas in Communication, Vol. 15 No. 7, Sept. 1997