

HLA-based Adaptive Distributed Simulation of Wireless Mobile Systems

Luciano Bononi, Gabriele D'Angelo, Lorenzo Donatiello
*Dipartimento di Scienze dell'Informazione, Università degli Studi di Bologna,
Via Mura Anteo Zamboni 7, 40126, Bologna, Italy
{bononi, gdangelo, donat}@cs.unibo.it*

Abstract

Wireless networks' models differ from wired ones at least in the innovative dynamic effects of host-mobility and open-broadcast nature of the wireless medium. Topology changes due to simulated hosts' mobility map on causality effects in the "areas of influence" of each mobile device. The analysis of wireless networks of interest today may include a potentially high number of simulated hosts, resulting in performance and scalability problems for discrete-event sequential simulation tools and methods, on a single physical execution unit (PEU). In a distributed simulation, the main bottleneck becomes the communication and synchronization required to maintain the causality constraints between distributed model components. In this work we propose a HLA-based, dynamic mechanism for the runtime management and allocation of model entities in a distributed simulation of wireless networks models, over a cluster of PEUs. By adopting a runtime evaluation of causal bindings between model entities we map the causal effects of virtual topology changes to dynamic migration of data structures. A prototype migration-heuristic is proposed to dynamically evaluate and balance the migration overheads, and the load distribution, with respect to the reduction in the "external" communication. Preliminary results demonstrate that the mechanism's heuristics lead to a reduction in the percentage of external communication between the PEUs, limited overheads and performance enhancements for a worst-case scenario.

1. Introduction

A wide research work has been done in recent years in the field of tools and methodologies for modeling and simulation of wireless systems [4, 22, 23, 27, 29, 34, 37, 38, 40], and in the simulation-based investigation of wireless systems, e.g. [7, 9, 18, 20, 36]. Among the relevant scenarios considered Cellular systems, PCS networks and the Mobile Ad-Hoc Networks (MANETs) are gaining an increasing relevance [4, 7, 9, 18, 20]. Wireless networks currently considered interesting for the analysis may include a potentially high number of simulated hosts. The simulation of every host may require a relevant computation time (e.g. due to simulation of protocol-stacks and applications running on top). This is often unpractical or impossible to simulate on a classical Von Neumann (mono-processor) architecture [28, 33, 34]. The simulation is unlikely to have success because of huge memory requirements and large amount of time required to complete.

Many practical experiences have demonstrated that a speed-up in the simulation of network systems is achievable using parallel and distributed models and architectures, i.e. a Parallel Discrete Event Simulation (PDES) approach [15, 16], e.g. Glomosim [40] based on PARSEC [1], Maisie [34], parallel and distributed implementations based on Network Simulator (ns-2) [21, 31, 32, 37] based on RTI-KIT [31], on ANSE/WARPED [30], on TeD [27], USSF over Warped/Notime [Rao99], Wippet [22], SWiMNET [6], and many others [21, 23, 34]. In order to exploit the maximum level of computation parallelism, many research activities dealt with dynamic balancing of logical processes' executions (both cpu-loads and virtual time-advancing speeds) by trading-off communication, synchronization and speedup, both in optimistic and conservative approaches [5, 10, 11, 14, 17, 35, 39].

High Level Architecture (HLA) is a recently approved standard (IEEE 1516) dealing with component-oriented distributed simulation [8, 12]. It defines rules and interfaces allowing for heterogeneous components' interoperability in distributed simulation. The definition of model components (formally known as federates) with standard management APIs brings to a high degree of model re-usability. The HLA standard defines APIs for the communication tasks and synchronization between federates. The simulation is supported by a runtime middleware (RTI). The RTI is mainly

This work is supported by MIUR FIRB funds, under the project: "Performance Evaluation of Complex Systems: Techniques, Methodologies and Tools"

responsible for providing support for time management, distributed objects' interaction, attributes' ownership and many other optimistic and conservative management policies [8, 12, 24].

By softly relaxing model accuracy and load balancing issues, respectively, many approaches have been investigated in order to reduce the overhead effects of distributed synchronization and communication in both optimistic and conservative distributed simulations. The motivation for this communication-reduction approach is the frequent adoption of networked cluster of PCs, in the place of shared-memory or tightly-coupled multiprocessors, as the execution units of the distributed simulation, primarily for cost reasons. The high network latency in these clusters could play a fundamental role in determining the weight of communication and synchronization between the distributed model components. Solutions have been proposed, based on relaxation and overhead elimination, by dynamically introducing higher levels of abstraction and merging in system sub-models [13, 19, 30]. Solutions preserving full model-detail have been proposed to maintain a global shared-state, and to reduce the effect of broken event-causality order, by dynamically filtering the event- and state-information. Examples can be found, based on interest management groups [35], responsibility domains, spheres of influence [24], multicast group allocation [2], data distribution management [12], grid distribution and routing spaces [11, 12], model and management partitioning [6]. These approaches rely on the reduction of communication obtained when the update of a event- or state-information (e.g. event and/or anti-message) does not need to be flooded to the whole system, but is simply propagated to all the causally-dependent components. This is the basis of publishing/subscribing mechanisms for sharing state-information and event-notifications between causally dependent components [8, 12, 31].

Simulation models for wireless systems 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. Topology changes due to simulated hosts' mobility map on causality effects in the "areas of influence" of each mobile device, resulting in dynamically shaped causality-domains. Given two or more neighbor-hosts, simply sharing the wireless medium without any end-to-end communication session on, the causal effect of signal interference (and Medium Access Control policies) could result in a chain of local-state events up to the transport and application layers [18]. 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.

We define a *dynamic system* as a system where the interactions (i.e. the causal effects of events) are dynamically subject to fast changes driven by the system (and model) evolution over time. Given this definition, a wireless network (e.g. a MANET) can be considered a highly dynamic system. A static definition of publishing/subscribing lists, groups and causal domains could not be more convenient, in presence of a dynamic system, than implementing a complete state-sharing. A dynamic approach for the distribution of events and state-information (e.g. dynamic lists and groups updates) would lead to network communication overheads. In some scenarios, the cost of list-updates or fine-grained events' communication between a dynamically variable set of components could be traded-off with the migration cost needed to cluster the whole interacting components on a single Physical Execution Unit (PEU). This would be more attractive if the object migration could be implemented incrementally as a simple data-structure (i.e. state) transfer, and if the object interaction would be maintained for a significant time (time-locality). In our approach, we define as Model Entity (ME) the data structure defined to model a Simulated Mobile Host (SMH)¹. A certain degree of time-locality of local communication can be considered an acceptable assumption in many mobile wireless systems, depending on the motion model characteristics. Our proposal is to define and investigate a simple mechanism allowing for SMH migrations in HLA-based distributed simulations. To the best of our knowledge, currently the HLA standard and existing Runtime Infrastructures (RTI) does not define migration facilities. This problem has been considered also in [25, 26]. We realized a prototype migration framework, adopting it together with a heuristic migration policy, whose aim is to dynamically partition and cluster the interacting SMH among federates executed over a set of PEUs. SMHs in our approach have a common definition and can be migrated simply by serializing and transferring their state-information. This would realize an example of a prototype framework and an adaptive, tuneable mechanism able to react to dynamic systems' behavior (like the mobile and wireless systems') under the communication-reduction viewpoint. In this work the prototype implementation of this mechanism is outlined and preliminary results of a set of simulation-tests are presented.

The paper structure is the following: in section 2 we outline some concepts about the wireless mobile networks scenarios and distributed simulation framework at the basis of our proposal; in Section 3 the key issues for the adaptive migration-framework implementation and the proposed migration heuristics are defined; in section 4 a prototype wireless system's model and a preliminary set of simulation results are presented; in section 5 we summarize our conclusions and future work.

¹ Specifically, we consider a Model Entity (ME) as the data structure needed to model the state of a Simulated Mobile Host (SMH) logical entity. In the following we will refer equivalently to ME and SMH as the migrating components of the model.

2. Distributed simulation of wireless networks

The objective of our work is to allow for efficient Parallel Discrete Event Simulation (PDES) of wireless and mobile systems. A reference scenario is a cellular, PCS, or a Mobile Ad-Hoc Network (MANET) topology characterized by mobile hosts, multiple cells and multi-hop routing policies to be investigated [7, 18, 20]. Communication is commonly defined between neighbor hosts and determines a causal effect in the simulated system. The causal effect of communication may be extended to all the neighbor-hosts (even if their communications are partitioned over different logical channels) due to the broadcast nature of the wireless medium. Thus, a high degree of causality in the simulation of the wireless hosts' communication is driven by the local-topology interaction (i.e. messages) between neighbor hosts. This interaction can be considered the most relevant causality effect to be modeled in many wireless system simulations (e.g. when evaluating MAC protocols, routing protocols, or physical interference effects). Even a multi-hop communication between non-adjacent hosts can be considered as a communication based on local interaction between a chain of neighbor hosts. Similar concepts, under the modeling and simulation viewpoint, may be found also in the analysis and simulation of multi-agent systems [24]. As we mentioned before, mobile wireless networks can be considered highly dynamic systems: if a mobile host changes its position, it eventually interacts with a new community of neighbor hosts.

We implement a parallel discrete event simulation of model components (federates) by using a set of physical execution units (PEUs) connected by a physical network (e.g. a networked cluster of PCs). Our approach at this level is mainly focused on the communication reduction between the PEUs where federates are executed. Every federate is allocated and executed on a single PEU (i.e. one single federate cannot be split over two or more PEUs). We implement a federate as a single logical process, managing and updating the state information (data structures) of a set of (at least one) Simulated Mobile Hosts (SMHs). We follow a time-stepped, conservative approach for ease of implementation, and given the high number of unpredictable, simulated entities with a high ratio of interactions performed. This means a conservative, fixed-lookahead time-management based on a HLA RTI implementation. A communication between wireless hosts can be modeled as an interaction (i.e. message-event) between SMHs. If the sender and its neighbors belong to the same federate (i.e. they are executed on the same PEU²) their interactions are local (e.g. shared memory communication) and do not involve physical network communication. On the other hand, every interaction involving extra-federate participants requires at least one physical network communication, resulting in relevant synchronization delays. Clustering neighbor SMHs in the same federate could give the advantage of closing the causality effect of communication inside the PEU where the federate is executed. This would reduce the overhead of network-communication required by the RTI to achieve full synchronization. Before starting the simulation, in the initialization phase, an optimal allocation of SMHs could be evaluated and performed. The allocation optimality can be defined in many ways, depending on the target overheads' reduction: typically with respect to traffic-reduction assumptions (given the high latency of physical network communication) and/or computation-load balancing (to obtain the maximum degree of execution parallelism). By running the simulation, given the dynamic behavior of simulated hosts, the initial allocation could quickly become sub-optimal. As an example, simulated hosts characterized by high mobility change their role and neighbors (i.e. area of influence), affecting the wireless network topology. This result may translate in a performance degradation for the simulation speedup, mainly due to the increasing cost of communication and synchronization required between distributed model components (federates). If we assume a time-locality in the interaction between neighbor hosts, it could be convenient to migrate the foreign SMH to the federate (and to the PEU) where its new neighbors are located, by reducing in this way the cost of successive interactions. This assumption is typically verified in MANETs, e.g. most routing protocols are based on "proximity" concept to decide the routing path of communications, and such communications usually last for a significant time, following a bidirectional session-based scheme. The migration mechanism would lead to a "time-locality" of the causality effect inside each federate.

3. The distributed simulation framework

In the design phase, we evaluated how the modeling components of SMHs can be migrated under the HLA RTI implementation. We realized that at least two choices were possible: i) to implement "object" migration in the HLA meaning, by exploiting the HLA middleware components, and ii) to implement generalized "data structure" migration and data distribution services, implemented on the top of the HLA middleware components. The first choice would require to

² In our testbed a single federate is typically executed on a PEU, excepted some experiments where all the federates have been executed on a single PEU.

deal with pure HLA objects migration, and would represent a complex and expensive effort to create and implement “migration” primitives inside the HLA middleware components. Additional overhead due to the instantiation and re-activation of migrated objects would be required. Specifically, DDM and object ownership management of standard HLA components is expected to reveal a great synchronization overhead due to migrations. This choice was not considered as the best choice given our aims. In our implementation we followed the second approach, described in the following section.

3.1. The Generic Adaptive Interaction Architecture (GAIA)

The PDES simulator built to obtain an experimental evidence of our proposal is based on a distributed architecture made by a set of federates glued together by HLA middleware. A small set of Runtime (RTI) implementations are available today. The first available was the DMSO (Department of Military Simulation Office) implementation, funded by US Department of Defense (DoD). We adopted the DMSO implementation RTI-1.3NGv3.2 as the basis for our work because it was freely available (license based) and compliant with the standard definition [12]. On top of the RTI we built a middleware extension called Generic Adaptive Interaction Architecture (GAIA). GAIA provides the interaction to the simulation core, the location and distribution data management, the random number generator, tracefile-logging and other simulation facilities. We adopt HLA federates as the management components of a set of (at least one) SMH model entities. The HLA definition and components to handle distributed simulation from the federate viewpoint is left untouched. The target of GAIA is to provide migration and service APIs to the simulation developer. Anyway, we don’t migrate federates nor HLA objects. We implement SMH models as code (they have a common code definition in the federates) with data structures to define and maintain the SMH state information. We choose to migrate the “data structure”, i.e. the state information of SMHs between federates. This requires to design and implement a supporting middleware for the “data structure” management and distribution of SMH entities between HLA federates. Our models can be simply executed as HLA federates in the static approach. If it is required to exploit migration, than the models should be defined to deal with migration APIs provided by GAIA on the top of HLA middleware. The equivalent of data distribution management and object ownership in the HLA meaning has been re-implemented in the GAIA framework. This allow us to deal with controlled overheads and ad hoc implementation for our purposes. The overhead related to the object migration, and object ownership management, may be significant in a generalized HLA DDM implementation. Bundling multiple objects within a single ownership transfer could mitigate the object-ownership management overhead, but would require to deal with ownership-clustering management for our models. In the current release, GAIA includes only prototypal data-location and distribution-management services whose overheads have been considered in the final results shown (see section 4.2.3). This allows us to evaluate the effects of the proposed adaptive migration mechanism, shown in this work. In the current release, the GAIA framework includes also the simple heuristic functions defined to evaluate the migration of SMHs between PEUs. The RTI middleware extension and the core simulation are built using the Java environment and connected to the DMSO RTI by JavaBindings originally provided by DMSO [12].

To test our framework we implemented a time-stepped, conservative, parallel discrete-event simulation of a mobile wireless system. The density of simulated hosts in our expectations should reach high values. The amount of physical memory required in a single PEU (limited to 256 MB RAM) to complete the whole simulation could become a problem for massive scenarios’ simulations.

3.2. The heuristic migration-policy definition

The dynamic migration of simulated hosts is not free of costs: some analytical or heuristic metrics are required, to be executed at runtime, to define “if and where” it would be profitable to migrate a SMH. The state size of a SMH would be a relevant parameter to be considered in the heuristic migration metric. Another relevant parameter to be considered is the amount of “time-locality” of the causal dependency between neighbor hosts, depending on the motion models, the interaction rate between SMHs, and the overall load balancing between the PEUs. Under the external communication-reduction viewpoint, it would be optimal to allocate every object on a single PEU, running the distributed simulation over a single PEU. Obviously, this is not the intended purpose of the mechanism: the external communication-reduction needs a trade-off with effective load-balanced parallel execution. The overhead of data distribution and object ownership management may be another relevant cost to be evaluated when a SMH migrates.

The heuristic migration rules in this preliminary study are simple and sub-optimal. Let a tagged SMH(j) be executed on the i -th PEU. Let us define $R_{j_e} = M_e/M_i$ as the ratio of the M_e “external” messages sent to the e -th PEU with respect to the number (M_i) of “local” messages sent within the local (i -th) PEU. Every SMH(j) evaluates the defined ratio R_{j_e}

for every foreign e -th PEU. If the maximum ratio obtained is greater than a threshold-value $k(j)$ (locally defined by every SMH(j), uniformly distributed with global average value K), the corresponding PEU is chosen as the destination for the SMH(j) migration in the next timestep. No migration is performed, otherwise. Upon arrival on a new PEU, every SMH resets its message counters (M_x). The average value of K is a simple tuning parameter that can be used to control the rate of migrations and the threshold of external communication required in order to balance the migration overhead.

3.3. The heuristic load-balancing policy definition

The steady state behavior of the proposed heuristic in isolation would lead to a long-term concentration of the SMHs over a restricted set of the available execution units, because the adaptive effect is focused on the “external” communication overhead. For this reason, we introduced the migration heuristic on the top of a simple load balancing policy implemented by the GAIA middleware supporting the set of federates. The SMHs migration towards/from a tagged PEU is blocked whenever the number of SMHs in execution over that PEU exceeds/falls below a 10% range boundary around the average value of ($\#SMHs / \#PEUs$), i.e. the total number of SMHs in the system divided by the total number of PEUs. The migration is reactivated if the SMHs range for the tagged PEU is valid. Again, this simple heuristic is sub-optimal and has the role of a trivial load-balancing between the PEUs.

It is worth noting that the proposed migration and load balancing policies require the general knowledge of the number of PEUs and SMHs involved in the simulation. In our approach these values are constant and assumed to be known in advance, when the simulation is started. If new PEUs and SMHs are dynamically allocated or generated while the simulation is running, then additional periodic exchange of information among federates would be required. We believe this point should not be critical, since the number of local SMHs would be known at runtime by every federate, and the frequency of information-update could be traded-off with the load-balancing accuracy.

4. Model definition and experimental results

4.1. Wireless system’s model definition

Now we illustrate the key concepts of our target wireless system and model definition. We assume a high number of simulated mobile hosts (SMHs), each one following a mixed variation of the Random direction and Random Waypoint (RWP) motion models [3]. This motion model is quite far from reality. The choice was driven by the unpredictable and uncorrelated mobility pattern of SMHs under the motion viewpoint. This would represent the worst case analysis for our mechanism, because any heuristic definition governing the mechanism cannot rely on any assumption about the motion correlation and predictability of SMHs. The only correlation effect we would exploit in our mechanism is given by the “time-locality” of communication sessions between neighbor-hosts. Given the mechanism definition, our feeling is that other widely used motion models, like any restricted, correlated or Group Mobility models, would be expected to give better reactions than the adopted RWP model, for any migration heuristic defined. In the following, our RWP motion model is defined. Every SMH swings between a sequence of mobile and static epochs. At the beginning of a mobile epoch, it generates a new, uniformly-distributed direction maintained for a geometrically distributed number of timesteps (10 on the average), then it enters a static epoch, whose duration is still geometrically distributed (10 timesteps on the average). The cycle is repeated for the whole simulation by every SMH. Sometimes we considered two motion sub-models related to the motion speed: (slow-mobility) S-RWP and (fast-mobility) F-RWP. The two motion sub-models considered have different speed properties in the mobile epochs: the F-RWP model is characterized by high speeds (uniform distribution in the range 0-100 spaceunits/timestep), and S-RWP is based on lower speeds (uniform distribution in 0-25 spaceunits/timestep). Space is modeled as a torus-shaped 2-D grid-topology, populated by a constant number of SMHs. The torus space topology, indeed unrealistic, is commonly used by modelers to prevent non-uniform SMHs’ concentration in any grid portion. This is our basic approach, in order to evaluate the mechanism behavior in a worst case scenario, where the clustering of SMHs is not trivially determined by high concentration in small areas. We believe that this is a stressing example for our mechanism, because it will lead to a high migration overhead, given the motion model defined. The simulated space is wide and open, without any movement constraints or obstacles: in future work we are interested to model more complex landscape topologies. The modeled communication between SMHs is a constant flow of ping messages (i.e. constant bit rate), transmitted by every SMH to its nearest device. This is a rough abstraction of the MANET communication given by routing processes or applications executed by the SMHs. We made this choice in order to stress the migration mechanism under the mobility effects of continuously transmitting SMHs. In our proposal, since the SMH migration policy is evaluated on the basis of the local and remote interaction (i.e. communication), no

communication translates in no migration needs, hence no additional communication, synchronization and migration overheads. A more complex network traffic model would give additional local computation for each SMH, resulting in additional advantages of parallel execution. This is one of the reasons why we consider the proposed scenario a worst case for the speedup analysis. The rate of ping messages is constant because it is the control parameter of communication: increasing/reducing the ping rate would be equivalent to stretch the simulated time axis, by maintaining the obtained results. We will extend this model with the real implementation of message flows, routing protocols and applications as a future work.

4.2. Experimental results

We ran our simulation experiments over a variable set of M PEUs equipped by Dual-Pentium3 600Mhz, 256MB RAM, connected by a FastEthernet (100Mb/s) LAN. We performed multiple runs of each experiment, and we evaluated the confidence intervals of simulation results, with a 90% confidence level. In all the results the confidence interval size is around 3-5% of the average value shown. In the following we may refer to “static” and “dynamic” distributed simulation as a simulation with migration heuristic OFF and ON, respectively. All the performed experiments start with a pseudo-random, uniform distribution of a variable number of SMHs (300 up to 1500) over the grid topology. Initially, the set of SMHs is randomly allocated over the set of available PEUs, i.e. no optimal allocation is defined, based on topology distribution. This choice was made because it would demonstrate the transient dynamic effect of our migration mechanism. Moreover, this would represent the distribution that would be asymptotically obtained in a “static” simulation, starting from an optimal allocation scheme, without any reaction to the SMHs’ mobility. Most of the figures presented show transient behavior of the performance indices, because this describes the dynamic effect of the proposed mechanism. Steady-state analysis results to capture the asymptotical behavior of the proposed mechanism are discussed. It is worth noting that in this preliminary work we are suggesting a simple prototype-mechanism in order to obtain qualitative indications about our adaptive approach, and a more accurate analysis and implementation will be required.

4.2.1. The reaction to the initial distribution. The visual effect of the dynamic allocation between PEUs is shown in figure 1 and figure 2.

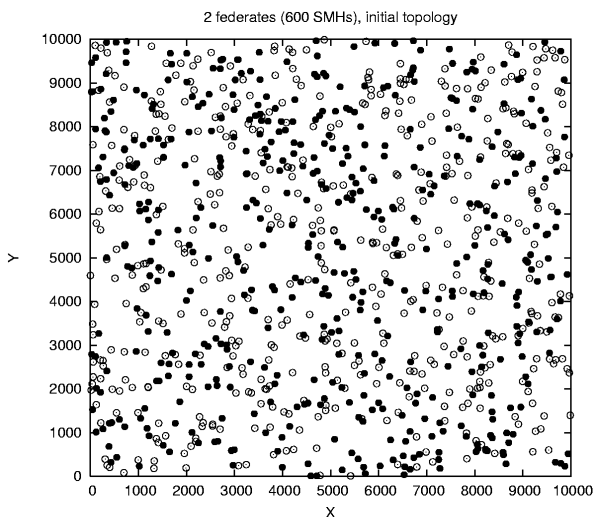


Figure 1. 600 SMHs, initial random distribution over black PEU1 and white PEU2

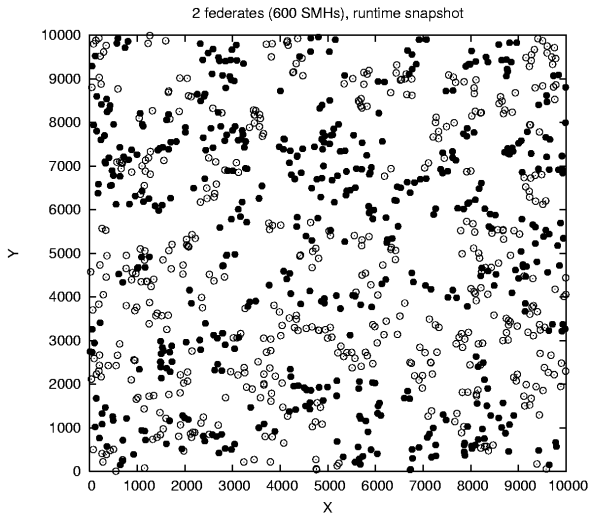


Figure 2. 600 SMHs, steady-state distribution over black PEU1 and white PEU2, migration on

In figure 1 a snapshot of the initial random distribution of 600 (out of 900) SMHs is shown inside the 2D grid space. For readability, two (out of three) PEUs are shown: black dots refer to approx. 300 SMHs initially allocated on PEU1 and white dots refer to approx. 300 SMHs initially allocated on PEU2. Every SMH is mobile and transmits ping messages to its nearest neighbor. The SMH allocation over PEUs in a static distributed simulation (no migration) will maintain a distribution similar to figure 1. In few time-steps the initial, transient migration effect is shown in figure 2. By focusing the attention only to white/black SMHs, the dynamic clustering can be perceived by the presence of clusters and chains of highly related SMHs. This effect is maintained in adaptive way despite the SMHs' mobility. Desert areas, in figure 2, contain the SMH-clusters allocated on PEU3 (not shown for readability).

4.2.2. The local communication ratio (LCR). The wireless system simulation performed mainly focused on the evaluation of the communication cost needed to implement the model interactions between SMHs. We define as a “local communication” (LC) an interaction between SMHs clustered on the same federate (intra-federate communication on the same PEU). On the other hand, an “external” communication (EC) is a message involving a physical network communication between PEUs. We collected results regarding the local communication ratio $LCR = LC / (LC + EC)$. Results have been collected and analyzed with the heuristic migration policy respectively on and off (i.e. with a static allocation), with respect to the SMH density and with respect to the value of the migration parameter K . The LCR index is not related to the size of messages and describes how much the causality effect has been closed inside each federate by adopting the migration mechanism. This index is not relevant about the amount of speedup obtained, because it does not consider the communication overhead for the objects' migration and data distribution management. It simply demonstrates that, for a given mobility model, a given percentage of messages required to perform the simulation runs can be transformed from ECs to LCs. In figure 3 we show the transient effect of the percentage of local communication (LCR) in the considered system due to SMH migrations.

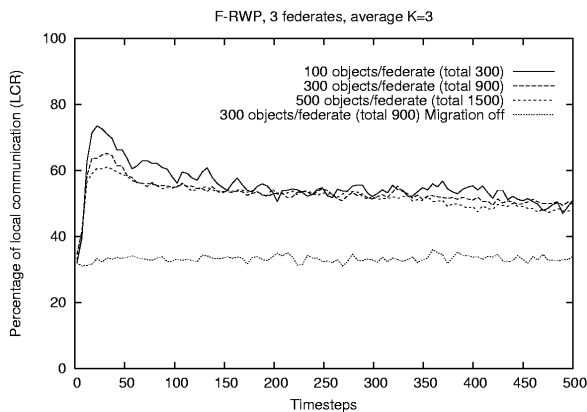


Figure 3. LCR vs. SMH density (only fast mobility F-RWP)

SMHs are initially distributed randomly over the set of PEUs. The dynamic allocation effect is shown: when migration is on, the SMH re-distribution increases the percentage of local communication, almost independently by the SMH density. The absence of migration (migration off) maintains the system on a flat level of local communication ratio (about 33%) which is expected given the uniform distribution of SMH over the 3 PEUs for this scenario. The steady state behavior for the dynamic system leads to a higher percentage of local communication than the static one (around 48% for Fast-RWP, in figure 3). Figure 4 shows the same results by varying the K value defined to control the migration heuristic. With low K values the initial re-allocation is performed a little bit faster than with high K values. The steady state behavior for the dynamic system leads to a LCR around 48% for Fast-RWP, and around 61% for Slow-RWP motion models (see figure 4), almost unaffected by different K values. This is given by the underlying effect of the load-balancing mechanism.

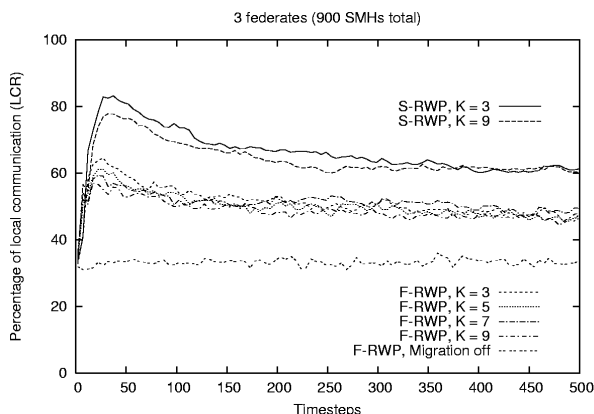


Figure 4. LCR vs. Average K value (slow and fast RWP mobility)

If the value of K is low, then each SMH would perform more frequent migrations. When a tagged PEU reduces its allocated SMHs under a 90% limit of the balanced number of SMHs, the load-balancing policy blocks the PEU borders, then no additional SMHs are exported by that PEU. Eventually, mobile SMHs will be imported and the PEU borders will be opened again for leaving candidates. The difference between F-RWP and S-RWP is a clear indication of the low/high “time-locality” effect of interactions captured by the migration heuristic. The adaptive effect of the migration mechanism is demonstrated also in figure 5: the transient LCR ratio is shown for Fast and Slow motion models. Fast mobility translates to less “time-locality” in the SMH interactions, then less local messages. Figure 5 shows that if the migration mechanism is switched off at runtime (timestep 100 on the figure for F-RWP), then the LCR ratio converges in few timesteps to the average value of a random, static allocation scheme. This demonstrates that any initial, optimal, static allocation policy is not adequate for this kind of dynamic models. In figure 6, the steady-state relationship between the average number of migrations in every timestep (left Y scale) and the percentage of local communication (LCR on the right Y scale) is explored: a low value of K translates in a high number of migrations and high percentage of local communication. By increasing the value of K we can significantly reduce the number of migrations performed by paying only a marginal reduction in the percentage of local messages. This point indicates that additional tuning-investigation of

the prototype mechanism is possible, to balance its overheads, parallel execution speedup and load balancing characteristics.

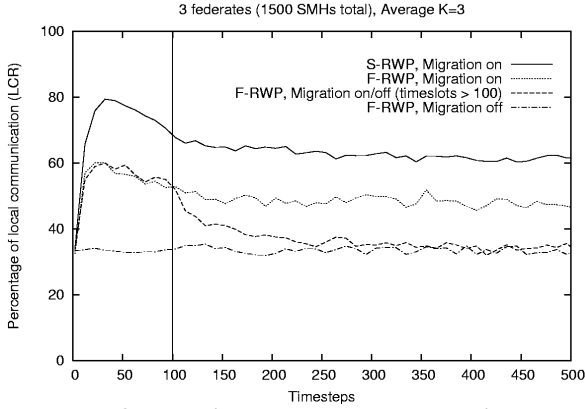


Figure 5. LCR, slow/fast mobility, migration on/off

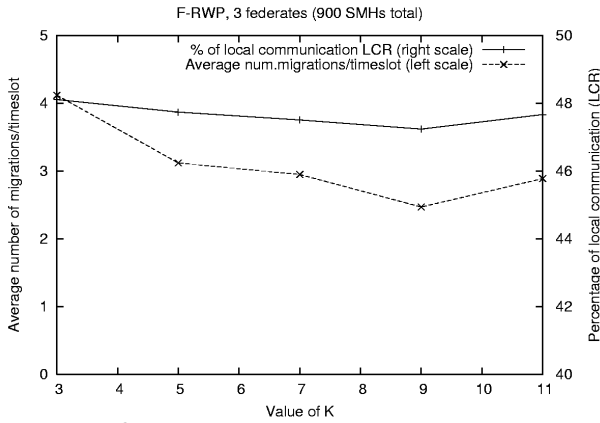


Figure 6. LCR and steady-state migration rate vs. K

4.2.3. Execution-time analysis. In table 1 we are showing the wall-clock time required for a short simulated-time interval, by adopting many distributed simulation implementations. We are not describing the speedup index obtained by distributed simulations with respect to a monolithic sequential simulation. This choice is made since we defined our model as a worst case scenario with respect to the sequential/parallel speedup analysis (e.g. few local computation, frequent communications, massive uncorrelated mobility). Ideally, a distributed simulation should give speedup advantages with respect to a sequential simulation, in a generalized scenario. Maybe this is not true in this specific scenario. The asymptotic speedup of the proposed implementations, and scalability beyond 3 processors will be evaluated as a future work. What we are interested in, is the evaluation of the two approaches for distributed simulation, since we suppose our migration-based mechanism could outperform a static approach, even in the worst-case scenario. As a worst-case scenario we mean a scenario where a high degree of uncorrelated mobility combined with frequent communication is performed. It is worth noting that in a fixed-hosts or in a limited-communication scenario, the effect of our mechanism is null (i.e. no migration) and we would get the static (migration-off) performance³. Before commenting the wall-clock time data shown in table 1, it is worth noting the simulated time of a simulation run is limited to only 300 timesteps. This is a really short simulation run. Significant run-length for simulations would be of many thousands timesteps, depending on the convergence time and variance of simulation indices. The wall-clock time indicated includes the initial re-allocation and object- distribution management overhead which is characteristic of the migration mechanism, in the dynamic approach. Despite the initial migration overheads, the results show that a simulated time of only 300 timeslots is sufficient for our implementation to recover the initial overhead. All the model and implementation choices have been made in

³ excepted the limited background execution-overhead of the migration and load-balancing heuristics.

order to stress the migration mechanism, with trivial migration and load-balancing heuristics implemented. With worst-case assumptions, and limited run length, the absence of additional overheads with respect to static implementation can be considered a success. The initial migration overheads are almost balanced by the external messages' reduction. Ongoing activity to study the steady-state speedup obtained by dynamic vs. static distributed approaches shown that i) in the worst-case model, steady-state speedup up to 23% has been obtained, with dynamic approach, for 1500 SMHs clustered over 3 federates, each one executed on a different PEUs, ii) the steady-state speedup is relative to many factors, e.g. the SMH's mobility parameters, the ping-rate, the tuning value of K.

As shown in table 1 for $M=1$, $N=3$ (i.e. for many federates executed on a single PEU), the migration policy is still convenient with respect to a static approach. This is because the dynamic clustering of SMHs filters the interactions that would be managed by the HLA middleware, reducing inter-federate synchronization overhead over the single PEU.

Table 1. Preliminary execution-time results (single run execution, 300 timesteps)

M PEUs, N federates, 1500 SMHs (constant)	Migration	Wall Clock Time (s) 300 ts
$M = N = 1$	Off	13 min, 18 sec
$M = 1, N = 3$	Off	16 min, 54 sec
$M = 1, N = 3$	On, $K=3$	15 min, 03 sec
$M = N = 2$	Off	9 min, 34 sec
$M = N = 2$	On, $K=3$	9 min, 20 sec
$M = N = 3$	Off	8 min, 19 sec
$M = N = 3$	On, $K=3$	7 min, 42 sec
$M = N = 3$	On, $K=7$	7 min, 31 sec

5. Conclusions and future work

We propose an adaptive framework, named Generic Adaptive Interaction Architecture (GAIA), for the dynamic allocation of model entities (ME) in a HLA-based framework for distributed simulations. GAIA is based on runtime migration and load-balancing policies, to reduce in adaptive way the amount of external communication between PEUs. We tested our mechanism for simple, sub-optimal migration and load-balancing heuristics in the testbed simulation of a prototype mobile wireless system, characterized by Simulated Mobile Hosts (SMHs). The runtime mechanism adapts the MEs' allocation over the PEUs to the dynamic interactions of SMHs. Preliminary results demonstrated the effectiveness of the proposed mechanism and performance enhancements, with controlled overhead for the worst-case scenario defined. We expect increasing performance to be obtained by optimizing the framework code, by tuning the mechanism's heuristics on the hypothesis and assumptions related to real system models.

Our future work will extend our analysis from a qualitative to a quantitative one. Additional efforts will be done in the code-optimization for the migration mechanism implemented, HLA-based interaction management and filtering, development of detailed heuristics based on analysis of multiple metrics and parameters, many different hardware and network architectures. The Ad-Hoc network scenario will be extended to deal with protocols and complex SMHs' behaviors. Specifically, the migration mechanism will be evaluated with respect to many dynamic factors to be modeled, in addition to the host mobility (e.g. dynamic communication-session establishment). The migration-based approach of this work could be extended to a wide set of simulations where the sequential approach gives low performance (e.g. multi-agent systems, genetic and molecular systems, P2P models).

6. Acknowledgments

The authors wish to thank the anonymous referees for their contribution to improve the paper quality.

7. References

- [1] R. Bagrodia, R. Meyer, M. Takai, Y. Chen, X. Zeng, J. Martin, and H.Y. Song, "PARSEC: a parallel simulation environment for complex systems", *IEEE Computer*, 31(10), October 1998, pp.77-85
- [2] A. Berrached, M. Beheshti, O. Sirisaengtaksin, and A. Korvin, "Alternative Approaches to multicast group allocation in HLA data distribution", *Proc. Of the 1998 Spring Simulation Interoperability Workshop*, 1998
- [3] C. Bettstetter, H. Hartenstein, and X. Pérez-Costa, "Stochastic properties of the random waypoint mobility model: epoch length, direction distribution and cell-change rate", *Proc. of 5th ACM int. workshop on Modeling analysis and simulation of wireless and mobile systems MSWiM2002*, Sept. 2002
- [4] A. Boukerche, S.K. Das, A. Fabbri, and O. Yildiz, "Exploiting model independence for parallel PCS network simulation", *Proc. of 13-th Workshop on Parallel and Distributed Simulation (PADS)*, Atlanta, Georgia, May 1999, pp. 166-173.
- [5] A. Boukerche, and S.K. Das, "Dynamic Load Balancing Strategies for Conservative Parallel Simulation", *Proc. of 11-th Workshop on Parallel and Distributed Simulation (PADS'97)*, June 1997, Lockenhaus, Austria, pp. 20-28
- [6] A. Boukerche, and A. Fabbri, "Partitioning Parallel Simulation of Wireless Networks", *Proc. of the 2000 Winter Simulation Conference (WSC)*, 2000
- [7] J. Broch, D. Maltz, D. Johnson, Y.C. Hu, J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols", *Mobile Computing and Networking*, 1998, pp. 85-97
- [8] J. Dahmann, R.M. Fujimoto, and R.M. Weatherly, "High Level Architecture for Simulation: an update", *Winter Simulation Conference*, December 1998
- [9] S.R. Das, R. Castaneda, and J. Yan, "Simulation-based performance evaluation of routing protocols for mobile ad hoc networks", *Mobile Networks and Applications*, Vol.5, Issue 3, September 2000
- [10] S.R. Das, "Adaptive protocols for Parallel Discrete Event Simulation", *Proc. of Winter Simulation Conference*, 1996
- [11] E. Deelman, and B.K. Szymanski, "Dynamic load balancing in parallel discrete event simulation for spatially explicit problems", *Proc. of the 12-th workshop on Parallel and distributed simulation PADS'98*, July 1998
- [12] DMSO: Defence Modeling and Simulation Office (1998), High Level Architecture RTI Interface Specification, Version 1.3, 1998
- [13] D. Dutta, A. Goel, and J. Heidemann, "Faster Network Design with Scenario Pre-filtering", *Proceedings of Mascots'02*, Fort Worth, Texas, October, 2002, pp. 237-246.
- [14] K. El-Khatib, and C. Tropper, "On metrics for the Dynamic Load Balancing of Optimistic Simulation", *Proc. of 32-nd Hawaii International Conference on System Sciences (HICCS'32)*, 1999
- [15] A. Ferscha, "Parallel and Distributed Simulation of Discrete Event Systems", In *Handbook of Parallel and Distributed Computing*, McGraw-Hill, 1995
- [16] Fujimoto, R.M., *Parallel and Distributed Simulation Systems*, John Wiley & Sons, 2000
- [17] B.P. Gan, Y.H. Low, S. Jain, S.J. Turner, W. Cai, W.J. Hsu, and S.Y. Huang, "Load balancing for conservative simulation on shared memory multiprocessor systems", *Proc. of the 14-th workshop on Parallel and distributed simulation (PADS'00)*, May 28-31, 2000, Bologna, Italy, p.139-146
- [18] M. Gerla, K. Tang, and R. Bagrodia, "TCP Performance in Wireless Multi-hop Networks", *Proceedings of IEEE WMCSA'99*, New Orleans, LA, Feb. 1999
- [19] P. Huang, D. Estrin, and J. Heidemann, "Enabling large-scale simulations: Selective abstraction approach to the study of multicast protocols", *proc. Mascots'98*, Oct. 1998
- [20] Internet Engineering Task Force, MANET WG Charter, <http://www.ietf.org/html.charters/manet-charter.html>
- [21] K.G. Jones, and S.R. Das S.R., "Parallel Execution of a sequential network simulator", *Proc. of the 2000 Winter Simulation Conference*, 2000
- [22] O.E. Kelly, J. Lai, N.B. Mandayam, A.T. Ogielski, J. Panchal, R.D. Yates, "Scalable parallel simulations of wireless networks with WiPPET: modeling of radio propagation, mobility and protocols", *Mobile Networks and Applications*, v.5, n.3, September 2000, pp.199-208
- [23] W.W. Liu, C.C. Chiang, H.K. Wu, V. Jha, M. Gerla, and R. Bagrodia, "Parallel simulation environment for mobile wireless networks", *Proc. of Winter Simulation Conference*, 1996
- [24] B. Logan, and G. Theodoropoulos, "The Distributed Simulation of Multi-Agent Systems", *Proc. of the IEEE*, 2001, pp. 174-185
- [25] J. Lüthi, and S. Großmann, "The resource sharing system: dynamic federate mapping for HLA-based distributed simulation", *Proc. of the 15-th workshop on Parallel and distributed simulation (PADS'01)*, May 2001, Lake Arrowhead, California, p.91-98
- [26] M. Myjak, S. Sharp, W. Shu, W. Wei, J. Riehl, D. Berkley, P. Nguyen, S. Camplin, and M. Roche, "Implementing object transfer in HLA", *Proc. 5-th Simulation Interoperability Workshop (SIW'99)*, Orlando, Florida, March 1999

- [27] K. Perumalla, R.M. Fujimoto, and A. Ogielsky, "TeD - A language for modeling telecommunications networks", *Performance Evaluation Review* 25(4), 1998
- [28] D.M. Rao, and P.A. Wilsey, "An Ultra-large Scale Simulation Framework", *Proc. of MASCOTS '99*, October 1999, pp.112-119
- [29] D.M. Rao, and P.A. Wilsey, "An object oriented framework for parallel simulation of ultra-large communication networks", *proc. 3-rd Intern.l symposium on computing and object oriented parallel environments*, Nov. 1999
- [30] D.M. Rao, and P.A. Wilsey, "Parallel Co-simulation of Conventional and Active Networks", *Proc. of MASCOTS'00*, August 2000
- [31] G.F. Riley, R.M. Fujimoto, M.H. Ammar, "A generic framework for parallelization of network simulations", *Proc. of MASCOTS'99*, College Park, MD, October 1999
- [32] G.F. Riley, M.F. Ammar, R.M. Fujimoto, K. Perumalla, and D. Xu, "Distributed Network Simulations using the Dynamic Simulation Backplane", *MASCOTS' 01*, Aug. 2001
- [33] G.F. Riley, and M.H. Ammar, "Simulating Large Networks How Big is Big Enough?", *Proc. of First Intern.l Conference on Grand Challenges for Modeling and Simulation*, Jan. 2002
- [34] J. Short, R. Bagrodia, and L. Kleinrock, "Mobile wireless network system simulation", *Wireless Networks* 1, August 1995, pp. 451-467
- [35] T.K. Som, and R.G. Sargent, "Model structure and load balancing in optimistic parallel discrete event simulation", *Proc. of the 14-th workshop on Parallel and distributed simulation*, May 2000, Bologna, Italy, p.147-154
- [36] K. Tang, M. Correa, and M. Gerla, "Effects of Ad Hoc MAC Layer Medium Access Mechanisms Under TCP", *ACM/Kluwer Mobile Networks and Applications*, 2001
- [37] UCB/LNBL/VINT The NS2 network simulator, <http://www.isi.edu/nsnam/ns/>
- [38] A. Varga, *OMNET++* in "*Software Tools for Networking*", *IEEE Network Interactive*. July 2002, Vol.16 No.4
- [39] V-Y Vee, and W-J Hsu, "Locality-preserving load-balancing mechanisms for synchronous simulations on shared-memory multiprocessors", *Proc. of 14-th workshop on Parallel and distr. simulation*, May 2000, Bologna, Italy, p.131-138
- [40] X. Zeng, R. Bagrodia, and M. Gerla, "GloMoSim: A library for parallel simulation of large-scale wireless networks", *Proc. of Workshop of Parallel and Distributed Simulation (PADS'98)*, Banff, Alberta, Canada, May 1998