Serdica J. Computing **4** (2010), 101–112

Serdica
Journal of Computing

Bulgarian Academy of Sciences
Institute of Mathematics and Informatics

GRID AND SIMULATION OF DIGITAL COMMUNICATION SYSTEMS*

Nikolai L. Manev

Abstract. The purpose of this paper is to turn researchers' attention to the use of grid computing for simulating digital communications and its large potential for decreasing significantly the duration of the experiments and for improving the statistical representativeness and reliability of the obtained results.

1. Simulation of Digital Communication Systems. The block-diagram of a modern digital communication system is depicted in Figure 1 (Sklar [8]).

Each functional block is implemented on a microchip(s), which ensures the required speed and reliability. But usually there are more than one possible algorithms (protocols) that realize the functions of a given block. The dynamical altering of these algorithms is also possible.

The designers and the authors of standards have to be sure that they implement/recommend the best algorithms in terms of performance and simplicity of realization. Unfortunately, the pure theoretical approach very often estimates

ACM Computing Classification System (1998): C.2.4, I.6.8.

Key words: simulation of digital communication systems, coded modulation, computational grid, Matlab.

*This work is partly supported by the National Science Fund of Bulgaria under Grant No. D002-146/16.12.2008.

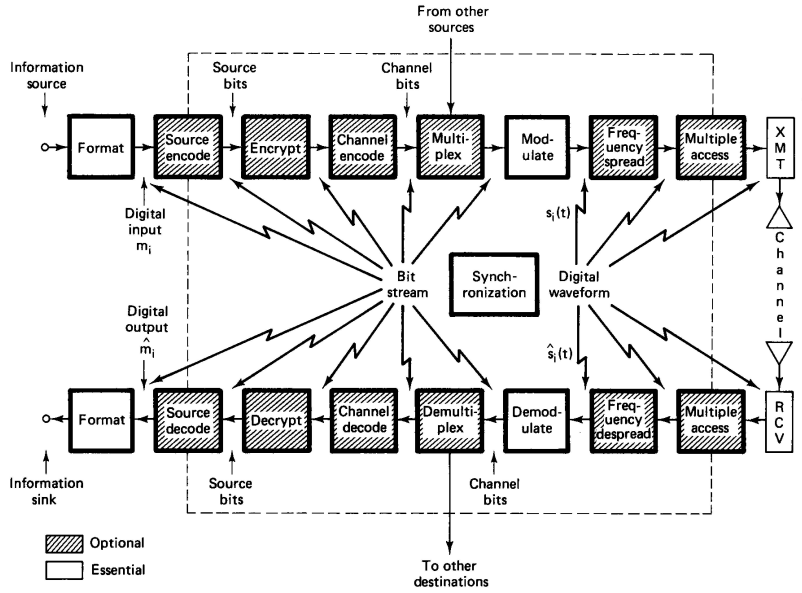

Fig. 1. Functional block diagram

performance within relatively wide limits. Based only on them the designer cannot make the final decision.

The alternative of this issue is software simulations of functional blocks, both separately for each block and combined in an integrated system.

Our interests and research address:

- **Modulation**: QAM, PSK, etc. – uncoded or Trellis coded (TCM), Integer coded, modulation based on algebraic integers, etc.
- **Error control coding (Channel coding)**: Reed-Solomon, Low density parity check (LDPC), Turbo, and Convolutional codes
- **Interleaving**: a tool for combating bursts of errors (due to multi-path propagation, incidental powerful noise sparks, etc.).

In digital communication two (nested) channels can be distinguished:

Analog (continuous) channel: The part of a communication chain between the modulator's output and the input of the demodulator. The modulator/demodulator maps digital symbols onto signals that can be efficiently transmitted over the communication channel. The information stream flowing through this part is represented by waveform signals.

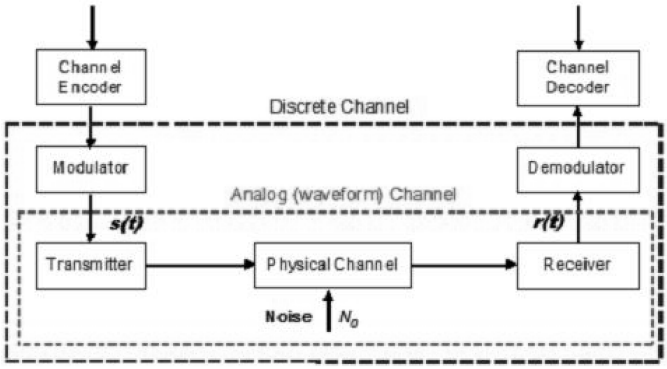

Fig. 2. Discrete and analog channels

Discrete channel: The enlargement of the analog channel obtained by including the modulator and demodulator. Both the input and output data streams of the discrete channel have digital format.

The input binary sequence **u** is transformed into waveform $s(t)$, which is sent through the analog channel (see Fig. 2). The analog channel separates time intervals of a length of T seconds such that the signal during each time interval presents one transmitted symbol. The transmission of one symbol in the analog channel may correspond to more than one symbol (e.g., several bits) across the discrete channel. The received waveform $r(t)$ is decoded into the output binary sequence **v**.

The physical channel attenuates the transmitted signal and introduces noise. The attenuation is generally caused by energy absorbtion and scattering in the propagation medium. The most common noise source is the ambient heat in the transmiter/receiver hardware and the propagation medium. The additive White Gaussian Noise (AWGN) channel model describes accurately satellite and line-of-sight channels. For wireless mobile communications, where signal propagation takes place near the ground, as well as for indoor communications a more adequate model is the Rayleigh noise model.

Discrete Channel Simulation:

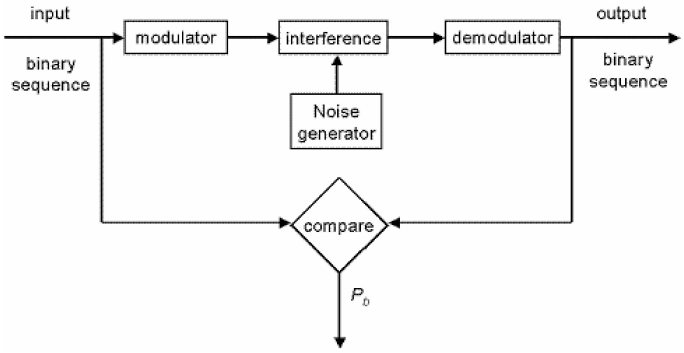

Fig. 3. The simulation flow chart scheme

The presence of noise through the analog channel results in adding an *error sequence* $\mathbf{e}$ at the discrete channel:

$$\mathbf{v} = \mathbf{u} + \mathbf{e}, \quad \mathbf{e} = (e_1, \ldots, e_n, \ldots) \in \mathbb{Z}_2.$$

The simplest error characteristic of the discrete channel is

$$\textit{Bit Error Ratio (BER)} = \frac{\text{number of 1s in } \mathbf{e}}{\text{length of } \mathbf{e}}$$

BER describes completely only the simplest model: *Binary Symmetric Channel* (BSC), which corresponds to a channel with only AWGN. In real communication, due to multi-path propagation, incident powerful noise sparks, moving of the devices, the errors are not uniformly spread, but rather they are packed in bursts. Such channels are described by more complex models, e.g., at least by the Gilbert 2-state model. Nevertheless, BER gives a good evaluation of the performance of the system and this is its very important characteristic.

Analog Channel Simulation:

$$\begin{aligned} r(t) &= \quad s(t) + n(t) && \text{AWGN} \\ r(t) &= a(t)s(t) + n(t) && \text{Rayleigh noise} \end{aligned}$$

Signal-to-noise Ratio (SNR): $\dfrac{E_s}{N_0}$ is defined as the ratio of the energy of the signal to the spectral density of the noise. It is usually measured in dB. ($E_{dB} = 10 \log_{10} E$)

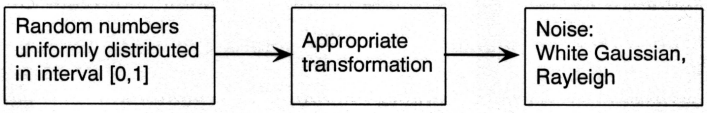

Fig. 4. The noise generator

There are numerous pseudorandom number generators available to the grid community. We have used the Mersenne Twister algorithm and our own generator based on high degree linear recursions.

2. Why Use Computational Grid? The simulation of a communication system or its block has to be performed for

- **various types of noise**: White Gaussian, Rayleigh, etc. noise
- **numerous values of signal-to-noise ratio**: at least every 0.5 dB in an interval of length 30–50 dB.
- **various types of modulation and channel coding schemes, and all possible combinations of them**
- **long bit sequences in each case**: e.g, if the expected BER is 10^{-7} then at least 10^9 bits should be processed
- **Another argument is that software simulations are heavy and run slow**

The simulations described above are independent of each other. Therefore, the task of computer simulation of digital communication devices possesses natural parallelism. The computational grid infrastructure fits very well this type of tasks and the computational grid approach is an adequate response to the needs of shortening development cycle.

We perform simulation on the SEEGRID grid infrastructure as a member of the Virtual Organization (VO) ***env.see-grid-sci.eu***. A VO is a temporary or permanent coalition of individuals, groups, organizational units or entire organizations that pool resources, capabilities and information to achieve common objectives. The SEEGRID infrastructure integrates computational and storage resources, located into research centers in the states of South Eastern Europe (Albania, Armenia, Bosnia and Herzegovina, Bulgaria, Croatia, Hungary, Greece, FYR Macedonia, Moldova, Montenegro, Romania, Serbia, Turkey). Currently there are 39 clusters with 3375 CPUs and more than 500 TB of storage (see `http://goc.grid.sinica.edu.tw/gstat/seegrid/`). All these clusters are integrated in a united infrastructure based on the ***gLite middleware*** (`https://edms.cern.ch/ file/722398/1.2/gLite-3-UserGuide.html`). The

gLite middleware provides Web Service APIs for most of its services, and provides new types of services, like the gLite WMS, gLite FTS, AMGA, etc. It also improved the reliability and scalability of the other services. The mandatory elements of and services offered by each cluster are:

- ***Computing Element (CE)*** provides user access to the grid resources. CE publicizes information about its resources and consists of several, equally configured, worker nodes.
- ***Worker Node (WN)*** provides the computational resource of the site. It is the hardware on which jobs runs. Each WN must be associated with a CE or SE.
- ***Storage Element (dCache, DPM or classic SE)*** provides storage resources for reliable data storage. It is the interface through which grid components communicate with a storage unit that has a protocol for local data access protocol, a protocol for secure wide-area transfer, exposes status of availability and space, and includes an *Storage Resource Manager (SRM)* interface (middleware components that manage shared storage resources on the grid and provide: uniform access to heterogeneous storage, ftp negotiation, access to permanent and temporary types of storage, advanced space and file reservation, and reliable transfer services).
- ***MON box (Monitoring and accounting)*** monitors the current grid status and reports complete jobs and resources used.

The set of services that are not tied to the specific site are called ***core services***. They are usually distributed among partners and include

- ***Virtual organization management system (VOMS)***. A system that manages real-time user authorization information for a VO. VOMS is designed to maintain only general information regarding the relationship of the user with his VO, e.g., groups he belongs to, certificate-related information, and capabilities he should present to resource providers for special processing needs. It maintains no personal identifying information beside the certificate.
- ***MyProxy*** credential repository system consists of a server and a set of client tools that can be used to delegate and retrieve credentials to and from a server. Normally, a user would start by using the *myproxy_init* client program along with the permanent credentials necessary to contact the server and delegate a set of proxy credentials to the server along with authentication information and retrieval restrictions.
- ***Relational Grid Monitoring Architecture (R-GMA)*** provides a ser-

vice for information, monitoring and logging in a distributed computing environment. (`http://www.r-gma.org/`).

- ***BDII (Berkeley Database Information Index)*** provides comprehensive information about the status of the resources and reports complete jobs and resources used.
- ***WMS (Workload Management System)*** distributes and manages the jobs among the different grid sites. The *Workload Manager Proxy (WM-Proxy)* is a service providing access to WMS functionality through a Web Services based interface. The *Workload Manager (WM)* is the core component of the Workload Management System. Given a valid request, it has to take the appropriate actions to satisfy it. To do so, it may need support from other components, which are specific to the different request types.
- ***file transfer service (FTS)***

The key to the job submission and resource matching process is the JDL description file. This file describes the necessary inputs, generated outputs, and resource requirements of a job/DAG through the JDL (Job Description Language). Here is an example of a JDL file:

```
JobType = "Normal";
Executable = "Matlab1.sh";
Arguments = "13 13";
Requirements = other.GlueSoftwareName == glibc &&
               other.GlueSoftwareVersion >= 2.5 &&
               other.GlueHostArchitecturePlatformType=="x86_64";
StdOutput = "Matlab1.out";
StdError = "Matlab1.err";
InputSandbox = {"Matlab1.sh"};
OutputSandbox = {"Matlab1.out","Matlab1.err"};
#DataAccessProtocol = {"rfio","gsiftp"};
Fuzzyrank=True;
rank = 1;
```

For more detailed description of how to use see-grid infrastructure for simulation we refer to [1], [2], [3], and [4].

We direct our efforts to executing MATLAB jobs (scripts) on Grid infrastructure. The argument in favor of doing this is that Matlab offers rich possibilities for using its metalanguage and a rich collection of toolbox routines as well as numerous scripts published of the Internet.

The Matlab can produce standalone (console) applications that enable end users to run MATLAB applications outside the MATLAB environment. These aplications are platform-specific and computer architecture dependable and need the MATLAB Compiler Runtime (MCR), which is an engine for executing compiled MATLAB code. The MCR is a standalone set of shared libraries that enable the execution of M-files, even on computers without an installed version of MATLAB. One has to install MCR (by executing the *MCRInstaller*) and to set the system paths on the target machine so his/her application finds the MCR and supporting files.

Our standalone applications were compiled on 64-bit PC with Fedora 8 and Matlab R2008a. We have tested two ways of running the produced standalone applications:

- Execute the *MCRInstaller* on a worker node (WN) just before running the corresponding standalone application. (Use the command:
`./MCRInstaller.bin -P bean421.installLocation=MATLAB/MCR -silent`)
- Execute (only once) the *MCRInstaller* on a worker node and then copy the installed libraries on a storage element (SE). Then each job makes a replica of these libraries on the WN where the job's standalone application runs.

Both approaches give almost the same delay but maybe the latter is better from a general standpoint.

3. Some Simulation Results. We illustrate the aforesaid in the previous sections with some results from simulation of modulation/demodulation algorithms. These simulations are based on the integer coded modulation research of the author carried out jointly with Hristo Kostadinov and Hiroshi Morita from UEC, Tokyo (see e.g., [5], [6], [7]).

Coded modulation is the collective term for all techniques which combine and jointly optimize channel coding and modulation for digital transmission.

- **Trellis coded modulation (TCM)**: It consists in expanding the input bits by a binary convolutional code and partitioning the used signal constellation into smaller subsets with a larger intra-set distance.
- **Integer coded modulation (ICM)**: A type of block coded modulation – each point of the signal constellation corresponds to a symbol of $\mathbb{Z}_A$ and is coded by a code over $\mathbb{Z}_A$.
- **Others**: Coded modulation based on Gaussian and algebraic integers.

Definition 1. *Let $\mathbb{Z}_A$ be the ring of integers modulo A and* $\mathbf{H}$ *be an $m \times n$ matrix with entries in $\mathbb{Z}_A$. An* integer code *over $\mathbb{Z}_A$ of length n with a*

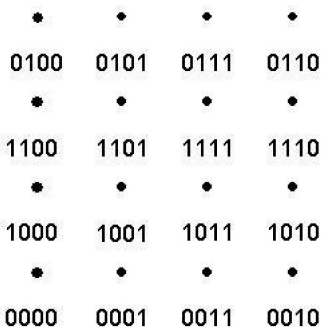

Fig. 5. An example of Grey coding

parity-check matrix $\mathbf{H}$ *is a subset of* $\mathbb{Z}_A^n$ *defined by*

$$\mathcal{C} = \mathcal{C}(\mathbf{H}, \mathbf{d}) = \{\mathbf{c} \in \mathbb{Z}_A^n \,|\, \mathbf{c}\mathbf{H}^T = \mathbf{d} \mod A\}$$

where $\mathbf{d} \in \mathbb{Z}_A^m$. *Usually* $\mathbf{d}$ *is the all-zero vector and then we say that* $\mathcal{C}$ *is an* $[n, n-m]$ *code.*

Definition 2. *Let* C *be an* $[n, k]$ *code over the integer ring* $\mathbb{Z}_A$. C *is a t-multiple* $(\pm e_1, \pm e_2, \ldots, \pm e_s)$*-error correctable code if it can correct (up to) any* t *errors with values from the set* $\{\pm e_1, \pm e_2, \ldots, \pm e_s\}$, *which occur in a codeword.*

($\pm$1)-error correctable codes:

Let us consider a square M-QAM constellation with $M = 2^{2k}$. Let us label each signal point in the M-QAM constellation by a pair $(i, j) \in \mathbb{Z}_A \times \mathbb{Z}_A$ of elements of $\mathbb{Z}_A$ where $A \geq 2^k$ (see Fig. 6). If the signal point $\circ$ is sent through the channel and the receiver makes a wrong decision the neighbor points marked with $*$ are more probable candidates for its output. Therefore, the use of ($\pm$1)-error correctable codes over $\mathbb{Z}_A$ enables correcting such errors and will improve the performance.

Examples

- $[6, 4]$ code over $\mathbb{Z}_{16}$ with

$$\mathbf{H} = \begin{pmatrix} 5 & 4 & 3 & 2 & 1 & 0 \\ 12 & 6 & 3 & 5 & 0 & 1 \end{pmatrix}$$

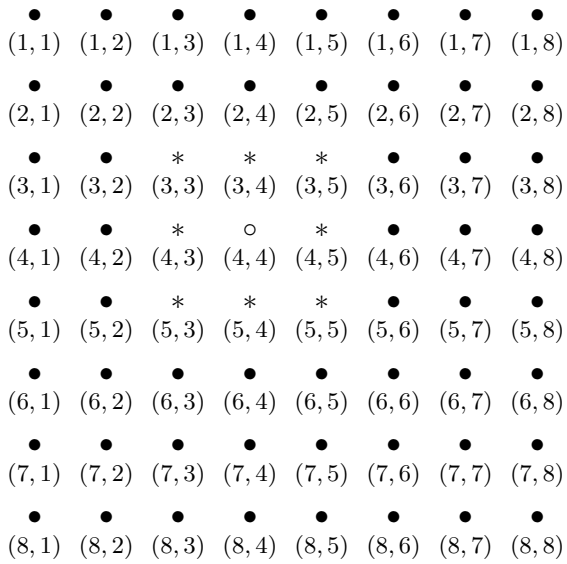

Fig. 6. Indexing a 64-QAM constellation

- $[8,6]$ code over $\mathbb{Z}_{16}$ with

$$\mathbf{H} = \begin{pmatrix} 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\ 7 & 0 & 12 & 6 & 3 & 5 & 0 & 1 \end{pmatrix}$$

- $[8,6]$ code over $\mathbb{Z}_{17}$ with

$$\mathbf{H} = \begin{pmatrix} 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\ 7 & 0 & 12 & 6 & 3 & 5 & 0 & 1 \end{pmatrix}$$

Decoding

Let a sequence of signal points, $s_{i_1 j_1}, s_{i_2 j_2}, \ldots, s_{i_n j_n}$, be sent trough the channel. In the coded case $(i_1, i_2, \ldots, i_n)$ and $(j_1, j_2, \ldots, j_n)$ are codewords of the code over $\mathbb{Z}_A$. At the receiver the decoder based on the received signal sequence outputs a sequence of signal points $s_{i'_1 j'_1}, s_{i'_2 j'_2}, \ldots, s_{i'_n j'_n}$.

- **Hard decoding**: If a syndrome of the received vector does not belong to the list of possible syndromes the decoder leaves the values (on the corresponding axis) unchanged.
- **Soft decoding**: The classical soft decoding for a "big square"(i.e., there are 9 possible values for each signal point).

- **Mixed decoding**: The decoder applies soft decoding when the syndromes are not among the possible ones.

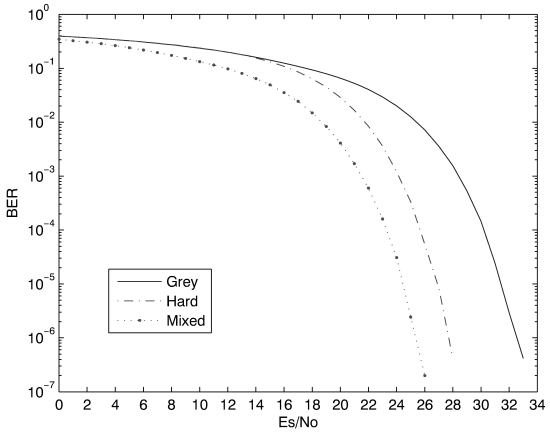

Fig. 7. 256-QAM: Grey, hard, and mixed decoding $[6, 4]$ code over $\mathbb{Z}_{16}$

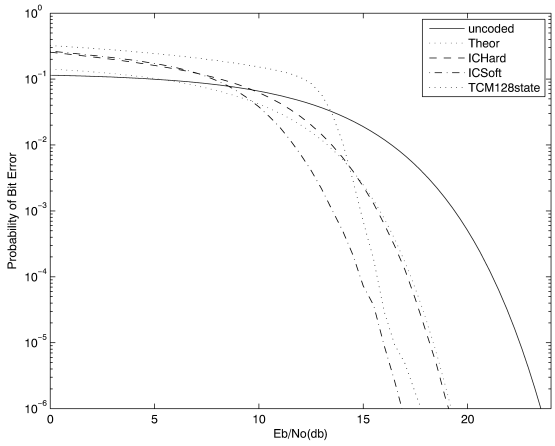

Fig. 8. 256-QAM: $[8, 6]$ code over $\mathbb{Z}_{17}$

4. Conclusion The use of Grid to simulate communication systems significantly decreases the duration of the simulation and improves the statistical representativeness and reliability of the results.

The graph for 128-state trellis coded modulation given in Fig. 8 is obtained by a simulation on a PC. This does not enable in reasonable time enough long

binary sequences to be processed. This is why the form of this graph contrasts with the smoothness of the other graphs. This example underlines the advantage of simulation on grid.

Acknowledgment. The author would like to thank the colleagues from GTA of IPP–BAS for their helping in using SEEGRID infrastructure.

REFERENCES

[1] Atanassov E., T. Gurov, A. Karaivanova. SALUTE application for Quantum Transport – New Grid Implementation Scheme. In: Proceedings of the Spanish conference on e-Science Grid Computing, Spain, 2007, 23–32.

[2] Atanassov E., T. Gurov, A. Karaivanova. Ultra-fast semiconductor carrier transport simulation on the grid. LSSC 2007, LNCS **4818**, Springer-Verlag, 2008, ISSN 0302-9743, 461–469.

[3] Bird I., B. Jones, K.Kee. The organization and management of grid infrastructures. *Computer*, **42** (2009), No 1, 36–46.

[4] Foster I., C. Kesselman, S. Tuecke. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. *Int. J. Supercomputing Applications*, **15** (2001), No 3, 200–222.

[5] Kostadinov H., H. Morita, N. Manev. Integer Codes Correcting Single Errors of Specific Types $(\pm e_1, \pm e_2, \ldots, \pm e_s)$. *IEICE Trans. on Fundamentals*, **E86-A** (2003), No 7, 1843–1849.

[6] Kostadinov H., H. Morita, N. Manev. Derivation on Bit Error Probability of Coded QAM using Integer Codes. *IEICE Trans. on Fundamentals*, **E87-A** (2004), No 12, 3397–3403.

[7] Kostadinov H., Manev N., Iijima N., Morita H. Double error correctable integer code and its application to QAM. In: Intern. Symposium on Information Theory and Its Applications (ISITA2008), Auckland, New Zenland, December 7–10, 2008, 566–571.

[8] Sklar B. Digital Communications: Fundamental and Applications, Prentice-Hall International, Inc., 1988.

Institute of Mathematics and Informatics
Bulgarian Academy of Sciences
Acad. G. Bonchev Str., Bl. 8
1113 Sofia, Bulgaria
e-mail: nlmanev@math.bas.bg

Received November 24, 2009
Final Accepted February 4, 2010