

QoS Management in Smart Grids: a Markovian Approach

Marcelino S. da Silva, Carlos R. L. Francês, João C. W. A. Costa, Diego L. Cardoso

Laboratório de Planejamento de Redes de Alto Desempenho, Universidade Federal do Pará, Rua Augusto Correa 01, Belém, Pará, Brasil, {marcelino,rfrances,jweyl,diego}@ufpa.br

Nandamudi. L. Vijaykumar, Solon V. de Carvalho

Laboratório Associado de Computação e Matemática Aplicada, Instituto Nacional de Pesquisas Espaciais, Av. dos Astronautas, 1758, São José dos Campos, São Paulo, Brasil, {vijay.nl@inpe.br, solon}@lac.inpe.br

Abstract— In Smart Grids, a variety of new applications are available to users of the electrical system (from consumers to the electric system operators and market operators). Some applications such as the SCADA systems, which control generators or substations, have consequences, for example, with a communication delay. The result of a failure to deliver a control message due to noncompliance of the time constraint can be catastrophic. On the other hand, applications such as smart metering of consumption have fewer restrictions. Since each type of application has different quality of service requirements (importance, delay, and amount of data to transmit) to transmit its messages, the policy to control and share the resources of the data communication network must consider them. In this paper Markov Decision Process Theory is employed to determine optimal policies to explore as much as possible the availability of throughput in order to transmit all kinds of messages, considering the quality of service requirements defined to each kind of message. First a non-preemptive model is formulated and after that a preemptive model is derived. Numerical results are used to compare FIFO, non-preemptive and preemptive policies.

Index Terms—Markov Decision Process, Quality of Service, Smart Grid.

I. INTRODUCTION

Smart Grids are characterized by a solid integration between a flexible and secure data communication network with advanced management techniques that control and monitor electric power systems. These advanced techniques are based on a number of sensors and actuators employed in generating, transmitting and distributing electrical power, that interact with the data network to provide users a set of applications that range from automation, control, distributed generation to applications that can perform online verification of the energy costs [1].

Since communication networks play a critical role in the smart grid [2], it has to be properly designed and implemented so that all the functionalities of the Smart Grid may be applied in practice. One important aspect is the quality of service (QoS) management [3]. Therefore, it is crucial to devise mechanisms that can manage and ensure that QoS requirements are satisfied.

There are already several solutions and defined standards to plan and control conventional data communication networks, which are generally dedicated for accessing Internet and the transmission of

multimedia data, e.g. [4-6]. Nevertheless, requirements for the emerging Smart Grid differ substantially from those of today's Internet [7]. In order to ensure QoS requirements, traditional power systems use dedicated links [8] based on standards, e.g. IEC 61850, or use solutions based on published literature [9-10]. However, such solutions are not feasible for Smart Grids due to the large volume of data transmitted by their applications and the inherent network complexity, given that the communication network of traditional power systems were geared around slow transmission speeds, which constrained the rate of data flow [11].

Observing the recent publications in the area, a lot of research is being conducted to: study the impacts of control and communication system vulnerabilities on power systems under contingencies [12]; evaluate, through prototyping or modeling, the feasibility of using several technologies for transmitting data for the purpose of implementing Smart Grids, e.g. [13-15]; characterize QoS requirements of Smart Grid applications, e.g. [16-17]; propose conceptual architectures for the implementation of data transmission networks capable of meeting the restrictions of the Smart Grid applications, e.g. [16, 18-19]; develop algorithms and methodologies to ensure compliance of new protocols, like IPv6, with the QoS requirements for these different classes of applications, e.g. [11].

However, a gap is observed in the development of methodologies that calculate control policies to indicate how much the network resources have to be allocated to each kind of Smart Grid application.

Since Markov Decision Process (MDP) [20] is a mathematical tool suitable to analyze stochastic control problem involving reactive complex systems composed by parallel and concurrent components, this research applied MDP to obtain optimal policies for sharing the available throughput (bits per second) between different classes of Smart Grid applications with distinct QoS requirements. In this problem, the arriving time and the transmission time of messages in some class of priority are independent of the messages in the other classes of priority and every message compete for the available throughput. So, the optimal policy has to indicate if an arriving message should be transmitted or its QoS requirements are not being observed (required throughput and importance) and the consequences of such decision; if the message is dropped some application may become unstable and some part of throughput may become idle; but, if the message is transmitted, a more important message may be rejected because there is not enough throughput to meet this request.

This paper is organized as follows. Section II describes the problem to be studied in this paper and the assumptions taken. It also shows an overview about Smart Grid, its main features and the main requirements for the Smart Grid communication system. An overview about Markov Decision Process is shown in Section III. The models proposed based on MDP are demonstrated in Section IV. Section V shows a case study, its results and the policy analysis. The results obtained using the control policies calculated by the models described in this paper are compared with results obtained using a FIFO (First-In First-Out) policy. Section VI closes with final remarks and ideas that could be explored in the future.

II. PROBLEM DESCRIPTION

Smart Grid is a term referring to the next generation power grid in which the electricity distribution and management is upgraded by incorporating advanced two-way communication and pervasive computing capabilities for improved control, efficiency, reliability and safety [21]. It means that many applications used today to control and manage power grids are being upgraded and new applications are being developed. Many of these applications are related to state estimation and direct state calculation, distributed generation control, system protection, wide-area situational awareness, post-event analysis, demand side management, smart metering and others [16]. All these applications involve data exchange between application servers and field devices such as energy meters, control, actuators, phasor measurement units (PMU), energy storages and others.

So, an appropriate communication network has to be implemented to support the many different applications of the Smart Grid. As described by [13], this communication network must, preferably, be designed based on a generic two-layer architecture as depicted in Figure 1. The server applications are connected on an IP-based network, such as, intranet of utility company or a metropolitan/regional data network interconnecting ISO/RTO (Independent System Operator/Regional Transmission Organization), often based on fiber optical networks. An appropriated field-level network should connect the field devices. The two layers are interconnected by means of Access Points acting as gateways between the two layers.

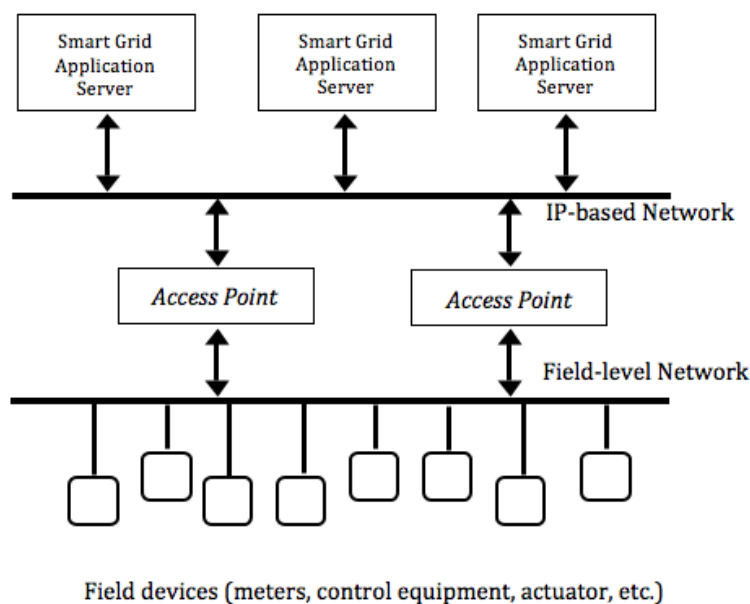


Fig. 1. Two-layer architecture for Smart Grid communication network [13].

Each application has its own QoS requirements and the consequence of not meeting them can affect just some service or the entire Smart Grid, depending on the application importance. For example, if the communication network drops the messages of the monitoring and control application, the power distribution service (the main service of the Smart Grid) may become offline. On the other hand, for demand response applications, customers may not find out the present costs of the power consumption. Observing the two-layer architecture described above, we can understand that the

Access Points can manage the messages being transmitted from one layer to another to guarantee that the applications will have their messages delivered according to their QoS requirements.

This proposal can be considered adequate and well inserted in terms of solutions of architecture to a managed control of QoS for Smart Grid already shown in the published literature. For example, [18] proposes an adaptation of an architecture to control end-to-end QoS developed by ITU-T for next generation networks, in which QoS Broker is the one responsible to centralize management of QoS. In this context, QoS Broker may determine an optimal policy control and transfers to the Access Point so that it can employ this policy to manage resource sharing of data network, during the message exchanges among field devices and application servers.

The idea is to obtain the control policy to be implemented in the Access Point to manage the messages being transmitted. So, we formulated the problem as a MDP in that the system is observed continuously and whenever an event occurs (arrival or end of transmission of a message) a decision is made to guarantee the QoS requirements of the messages.

Two models were developed: in the first, the control policy reserves an amount of throughput to be used just by the most important classes of applications, which is a non-preemptive control; in the second, the control policy is a preemptive control in that the less important messages will be immediately dropped to permit that the most important messages will be transmitted. Also, in both the models, messages from the less important classes of application should be delivered as much as possible observing its importance to the Smart Grid and its further implication to the transmission of messages of the more important classes of applications.

III. MARKOV DECISION PROCESS

In this paper, the problem is formulated as a Continuous Time MDP (CTMDP), since it considers that the times (between arrival of requests and that a request stays in the system) follow an exponential probability distribution. Also, the problem is formulated as an Infinite Horizon problem, since it can perform for a long, undefined period of time.

Briefly, to model a problem as a CTMDP, it is necessary to define [20]:

- The state space S : the set of all possible conditions (states) of the system;
- Sets of actions $\{A(s) / s \in S\}$: for each state $s \in S$, there is a set of possible actions $A(s)$, in which the operator must choose a single action at every decision time;
- A set of costs $\{c(s, a) / s \in S, a \in A(s)\}$: where $c(s, a)$ is the cost entailed to the system when it is in state $s \in S$ and a action $a \in A(s)$ is chosen;
- A set of transitions probabilities $\{p_{sz}(a) / s, z \in S, a \in A(s)\}$: where $p_{sz}(a)$ is the probability that, in the next decision time, the system is in state $z \in S$, given that action $a \in A(s)$ is chosen when it is in the state $s \in S$;

- $\{\tau(s,a) | s \in S, a \in A(s)\}$: expected time until the next decision time if the action $a \in A(s)$ is chosen in state $s \in S$.

Using these five elements, the stationary optimal policy R^* that minimizes the long-run average cost per time unit can be calculated. For this purpose, there are some classical techniques that can be used, e.g. Value Iteration Algorithm, Policy Iteration Algorithm and Linear Programming [20].

Figure 1 shows a diagram representing the transition of states as a time dependent function. An event occurs in a given time t_n ; after this event, the system's state then changes and, simultaneously, a decision is made. Between the instants t_n and t_{n+1} , system behavior will depend on the state and the decision taken in t_n . In t_{n+1} a new event that changes the system's state occurs and the process restarts. The optimal policy calculated indicates which decision (action to be chosen) should be taken at each instant of time (t_{n-1} , t_n , t_{n+1} , and so on); this decision will be stationary and depends only on the state of system.

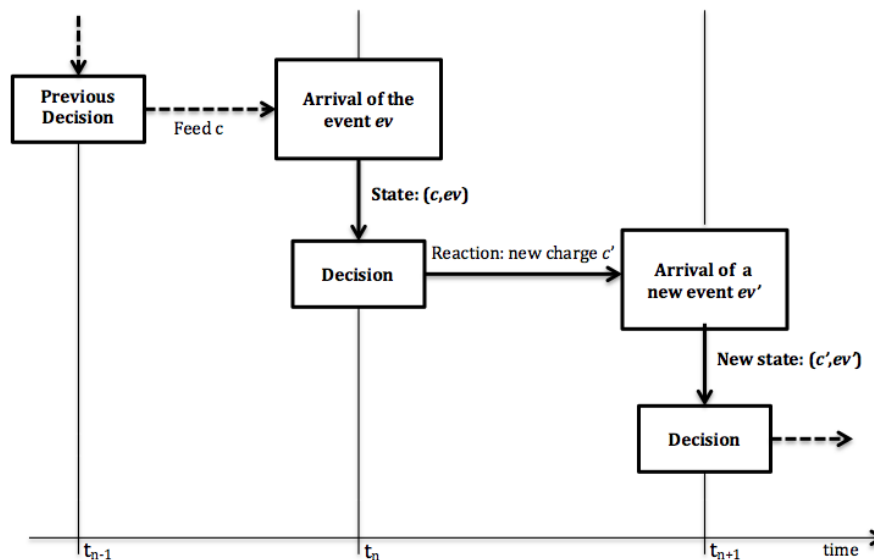


Fig. 2. Diagram representation of the state transition over time

IV. MODELS FORMULATION

The messages transmitted by the Smart Grid applications are characterized by their importance. So, if there are k classes of messages, where class 1 is the most important and class k is the least, possible events would be:

- $X_i \forall i=1, 2, \dots k$: a message in the i th class of priority arrives at the access point;
- $Y_i \forall i=1, 2, \dots k$: the transmission of a message in the i th class of priority is completed.

The interoccurrence time of events X_i and Y_i is assumed to be exponentially distributed with means $1/\lambda_i$ and $1/\mu_i$ respectively and these events are independent. λ_i and μ_i denote rate of occurrence of events X_i and Y_i , respectively.

In the formulated CTMDP, the maximum delay required as well as the amount of data to transmit

are summarized in throughput, which is in fact the resource being shared. The importance of the message is denoted by rejections cost.

In the next subsection, the non-preemptive model is described and, after that, the preemptive model is derived.

A. Non-Preemptive Model

The non-preemptive model is based on a classical model developed in [20] to optimally allocate channels to transmit messages. The resources being shared are the transmission channels, each message occupies only one channel and each channel handles only one message at a time. For this model, the optimal policy has a structure $L_1 \geq L_2 \geq \dots \geq L_k$ indicating that an arriving i th class priority message is accepted only when less than L_i messages are present and not all channels are occupied.

In this paper the resource being shared is the available throughput. Thus, the main difference to the classical model [20] is that each message occupies a different amount of resource (throughput), due to its class of priority. This suggests that, in some situations, it is not possible to accept high priority messages, as the unallocated throughput is not sufficient to transmit it within the specified maximum delay, however, less priority messages can be transmitted ($L_i < L_{i+1}$). So, it is natural that the structure $L_1 \geq L_2 \geq \dots \geq L_k$ may not be applied to share throughput. The model to share throughput is described below.

The state space S is

$$S = \{(n_1, n_2, \dots, n_k, ev) | n_i = 0, 1, \dots, \hat{T} / T_i \quad i = 1, 2, \dots, k; \sum_{i=1}^k n_i T_i \leq T; ev = 0, 1, \dots, k\} \tag{1}$$

where n_i is the number of i th class priority messages being transmitted, T is the total throughput, T_i is throughput required for each i th class priority message and ev is the last event waiting for a decision. $ev=1$ denotes X_1 , $ev=2$ denotes X_2 and so on. $ev=0$ denotes any Y_i (message transmitted). Note that class Y is not necessary to define the states, since no decision is made when the transmission of a message ends.

The optimal policy to be found decides whether each new request to transmit a message is accepted or rejected. So, if the element ev of the state denotes that a request arrives ($ev=1, 2, \dots, k$) and if there is enough throughput to serve this request, the set of possible actions $A(s)$ for each state $s \in S$ is:

$$A(s | ev = 1, 2, \dots, k; T - \sum_{i=1}^k n_i T_i \geq T_{ev}) = \begin{cases} 0, & \text{the arrived message is rejected} \\ 1, & \text{the arrived message is accepted} \end{cases} \tag{2}$$

If throughput is not enough to serve the request, then:

$$A(s | ev = 1, 2, \dots, k; T - \sum_{i=1}^k n_i T_i < T_{ev}) = \{0, \text{ the arrived message is rejected}\} \tag{3}$$

And if $ev=0$, the only option is:

$$A(s | ev = 0) = \{0, \text{ accept the end of the transmission}\} \tag{4}$$

To calculate the probability $P_{sd}(a)$ to move from a given state $s \in S$ to a state $d \in S$ when an action $a \in A(s)$ is chosen, the following property is used: $P\{Z_1 < Z_2, Z_3, \dots, Z_m\} = \sigma_1 / (\sigma_1 + \sigma_2 + \dots + \sigma_m)$ as long as Z_1, Z_2, \dots, Z_m are independent random variables with exponential distribution function with means $1/\sigma_1, 1/\sigma_2, \dots, 1/\sigma_m$, respectively. So, if $Z_i \forall i=1, 2, \dots, m$ is the time between occurrences of some event W_i , then σ_i denotes the mean rate of occurrence of event W_i .

The following function (5) is defined and it will be used in other Equations defined further.

$$h(n_1, n_2, \dots, n_k) = \lambda_1 + \lambda_2 + \dots + \lambda_k + n_1 m_1 + n_2 m_2 + \dots + n_k m_k \tag{5}$$

If $a=0$ when $s=(n_1, n_2, \dots, n_k, ev)$:

$$P_{sd}(0) = \begin{cases} \lambda_1 [h(n_1, n_2, \dots, n_k)]^{-1}, & d = (n_1, n_2, \dots, n_k, 1) \\ \vdots & \vdots \\ \lambda_k [h(n_1, n_2, \dots, n_k)]^{-1}, & d = (n_1, n_2, \dots, n_k, k) \\ n_1 m_1 [h(n_1, n_2, \dots, n_k)]^{-1}, & d = (n_1 - 1, n_2, \dots, n_k, 0) \\ n_2 m_2 [h(n_1, n_2, \dots, n_k)]^{-1}, & d = (n_1, n_2 - 1, \dots, n_k, 0) \\ \vdots & \vdots \\ n_k m_k [h(n_1, n_2, \dots, n_k)]^{-1}, & d = (n_1, n_2, \dots, n_k - 1, 0) \end{cases} \tag{6}$$

If $a=1$ when $s = (n_1, n_2, \dots, n_k, 1)$:

$$P_{sd}(1) = \begin{cases} \lambda_1 [h(n_1 + 1, n_2, \dots, n_k)]^{-1}, & d = (n_1 + 1, n_2, \dots, n_k, 1) \\ \vdots & \vdots \\ \lambda_k [h(n_1 + 1, n_2, \dots, n_k)]^{-1}, & d = (n_1 + 1, n_2, \dots, n_k, k) \\ n_1 m_1 [h(n_1 + 1, n_2, \dots, n_k)]^{-1}, & d = (n_1, n_2, \dots, n_k, 0) \\ n_2 m_2 [h(n_1 + 1, n_2, \dots, n_k)]^{-1}, & d = (n_1 + 1, n_2 - 1, \dots, n_k, 0) \\ \vdots & \vdots \\ n_k m_k [h(n_1 + 1, n_2, \dots, n_k)]^{-1}, & d = (n_1 + 1, n_2, \dots, n_k - 1, 0) \end{cases} \tag{7}$$

Similarly, it is possible to calculate $P_{sd}(a=1)$ when $s=(n_1, n_2, \dots, n_k, ev | ev=2, 3, \dots, k)$.

Using the property $\min(Z_1, Z_2, \dots, Z_m) = 1 / (\sigma_1 + \sigma_2 + \dots + \sigma_m)$ as long as Z_1, Z_2, \dots, Z_m are independent random variables with exponential distribution function with means $1/\sigma_1, 1/\sigma_2, \dots, 1/\sigma_m$, respectively, it is possible to calculate the expected time until the next decision epoch when $s \in S = (n_1, n_2, \dots, n_k, ev)$ and $A(s)=a$:

$$t_s(a) = \begin{cases} [h(n_1, n_2, \dots, n_k)]^{-1}, & a = 0 \\ [h(n_1 + 1, n_2, \dots, n_k)]^{-1}, & a = 1, s = (n_1, n_2, \dots, n_k, 1) \\ [h(n_1, n_2 + 1, \dots, n_k)]^{-1}, & a = 1, s = (n_1, n_2, \dots, n_k, 2) \\ \vdots & \vdots \\ [h(n_1, n_2, \dots, n_k + 1)]^{-1}, & a = 1, s = (n_1, n_2, \dots, n_k, k) \end{cases} \tag{8}$$

Finally, the cost entailed to the system when $A(s)=a$ is chosen when system is in state $s \in S$ is:

$$C_s(a) = \begin{cases} c_i, & s = (n_1, n_2, \dots, n_k, i | i = 1, 2, \dots, k), a = 0 \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

where c_i is the cost entailed to the system each time a message of i th class of priority is rejected.

B. Preemptive Model

In the non-preemptive model, a high priority message can be accepted if $T - \prod_{i=1}^k n_i T_i \geq T_1$, while, in case of preemptive, a high priority message can be accepted if $T - n_1 T_1 \geq T_1$. This means that just the high priority messages being transmitted must be observed to decide whether a new high priority message can be accepted. The lower priority messages being transmitted will be dropped one by one, beginning from the last class of priority (class k), until $T - \prod_{i=1}^k n_i T_i \geq T_1$.

Thus, the set of state space is the same presented in (1) and, for states $s=(n_1, n_2, \dots, n_k, ev \neq 1)$, $A(s)$, $P_{sd}(a)$, $\tau_s(a)$ and $c_s(a)$ are the same as non-preemptive model. However, when $s=(n_1, n_2, \dots, n_k, ev=1)$:

$$A(s | ev=1; T - n_1 T_1 \geq T_1) = \begin{cases} 0, & \text{the arrived message is rejected} \\ 1, & \text{the arrived message is accepted} \end{cases} \quad (10)$$

If $T - n_1 T_1 < T_1$ the only option is:

$$A(s | ev=1; T - n_1 T_1 < T_1) = 0, \text{ the arrived message is rejected} \quad (11)$$

If $s=(n_1, n_2, \dots, n_k, ev=1)$ and $a=0$, P_{sd} is the same defined in (6) and (7) is used if $a=1$ and $T - \prod_{i=1}^k n_i T_i \geq T_1$.

When $s=(n_1, n_2, \dots, n_k, ev=1)$ and $a=1$ some lower priority messages may be dropped. So, the number of j th class of priority messages, $j=2, \dots, k$, that remain in the next state can be calculated by:

$$m_j(n_1, \dots, n_k) = \begin{cases} n_j, & T - \prod_{i=1}^j n_i T_i \geq T_1 \\ n_j - \frac{T - \prod_{i=1}^j n_i T_i - T_1}{T_j}, & T - \prod_{i=1}^j n_i T_i < T_1 \text{ e } T - \prod_{i=1}^{j-1} n_i T_i \geq T_1 \\ 0, & T - \prod_{i=1}^{j-1} n_i T_i < T_1 \end{cases} \quad (12)$$

To simplify the notation, m_j is used as $m_j(n_1, \dots, n_k)$. So, when $a=1$ and $s=(n_1, n_2, \dots, n_k, 1)$:

$$P_{sd}(1) = \begin{cases} 1/h(n_1+1, m_2, \dots, m_k)^{-1}, & d=(n_1+1, m_2, \dots, m_k, 1) \\ \vdots & \vdots \\ 1/h(n_1+1, m_2, \dots, m_k)^{-1}, & d=(n_1+1, m_2, \dots, m_k, k) \\ n_1 m_1 [h(n_1+1, m_2, \dots, m_k)^{-1}], & d=(n_1, m_2, \dots, m_k, 0) \\ n_2 m_2 [h(n_1+1, m_2, \dots, m_k)^{-1}], & d=(n_1+1, m_2-1, \dots, m_k, 0) \\ \vdots & \vdots \\ n_k m_k [h(n_1+1, m_2, \dots, m_k)^{-1}], & d=(n_1+1, m_2, \dots, m_k-1, 0) \end{cases} \quad (13)$$

The expected time until the next decision epoch when $a=1$ and $s=(n_1, n_2, \dots, n_k, 1)$ is:

$$t_s(1) = [h(n_1+1, m_2, \dots, m_k)]^{-1} \quad (14)$$

And the costs entailed to the system when it is in state $s=(n_1, n_2, \dots, n_k, 1)$ are calculated as:

$$c_s(a) = \begin{cases} c_1, & a=0 \\ \prod_{i=2}^k (n_i - m_i) c_i, & a=1 \end{cases} \quad (15)$$

V. NUMERICAL RESULTS AND POLICIES ANALYSIS

Since the United States National Institute of Standards and Technologies (NIST) has defined the IEC 61850 [22] as one of the key standards in achieving interoperability in Smart Grid environment [23], it was used to compose the numerical example. Thus, specific requirements, such as delay, amount of data and importance of the message were studied to delimit the values shown in Table 1.

Therefore, 3 classes of priority messages were selected:

- Type 1 - TRIP messages (high priority message): The trip is the most important fast message defined in IEC 61850. Therefore, this message has more demanding requirements compared to all other fast messages. In this class there is a subclass used for protection, with a transmission time up to 3 ms and payload of 1 bit.
- Type 2 - Medium speed messages: this kind of message has transmission time less critical than type 1. The total transmission time shall be less than 100 ms. The payload of each message has 8 process values, each one with 16 bits.
- Type 3 - Low speed messages: this type should be used for slow speed auto-control functions, transmission of event records, reading or changing set-point values and general presentation of system data. The total transmission time shall be less than 500 ms. Some lower priority messages in this class transmit measured values such as energy. The payload of each message has 64 values, each one with 16 bits.

Applying the values defined in Table 1 to the models (Section IV), Iteration Values Algorithm [20] was used to obtain the policy that minimizes the expected cost in a long term. Then, Successive Overrelaxation Algorithm [20] was employed to obtain the steady-state probabilities. To quantify the benefits of the optimal policies obtained by the non-preemptive and preemptive models, the rejection probabilities of *i*th class of priority were calculated and were compared with a FIFO (First-In-First-Out) policy. A FIFO policy has been employed as this policy represents the case in which there is no kind of policy that can assure that QoS is considered.

TABLE I. PARAMETERS AND NUMERICAL VALUES

Parameter	Value
T_1	321 Kbits/s
T_2	13.44 Kbits/s
T_3	8.064 Kbits/s
λ_1	500 messages/s
λ_2	300 messages/s
λ_3	100 messages/s
μ_1	1000 messages/s
μ_2	30 messages/s
μ_3	60 messages/s
c_1	1
c_2	0.6
c_3	0.2

Fig. 3 shows the rejection probabilities for each type of message, applying FIFO, non-preemptive (R_{np}) and preemptive (R_p) policies, when the total throughput decreases from 1.5 Mbits/s to 500 Kbits/s. High priority messages are related to the integrity of the Smart Grid. So, in cases of

bottleneck that may occur due to a problem in the data network, its capacity is drastically reduced. That is the reason that the policy must assure to give attention that QoS demands are properly satisfied for such high priority messages, although this might affect low priority applications that are subject to starvation. This is what exactly happens, for example, when the total throughput is 500 Kbits/s where the results show that policies R_{np} and R_p act as a threshold avoiding fast increase in rejecting high priority messages, since the probability to reject high priority messages is 33% for R_{np} or R_p whereas for FIFO, it is 96%. The result obtained by the FIFO policy confirms the need for a policy that manages the QoS of applications, since, in case of 500Kbits/s, servers of the most important applications can not exchange messages with the field devices.

The main difference between R_{np} and R_p is observed in the last class of priority. In case of bottleneck, R_{np} blocks every message in this class, while R_p maintains a minimum of messages, accepting approximately 4%.

For the R_p policy, the peaks in blocking probabilities for 2nd and 3rd classes reflect situations where the preemption probability for a message on these classes is higher. These peaks occur when the total amount of throughput approaches a multiple value of the amount required by the high priority application. For the given example, the highest values are: 642Kbits/s, 963Kbits/s e 1284Kbits/s.

On the other hand, since R_{np} has as a feature reserving the throughput for high priority messages, to avoid them to be blocked, it would be necessary to reject all the low priority messages, whenever the total amount of the throughput approaches multiple value required by high priority applications, to insure that as many as possible high priority ones are served. Nevertheless, as R_{np} considers problems caused due to blocking each type of message by the associated cost c_i , rejecting all the low priority messages becomes quite expensive when compared to blocking some of the high priority messages.

This way, as can be observed in Figure 3, close to the values of 642Kbits/s, 963Kbits/s and 1284Kbits/s, results for R_{np} policy approach those of a FIFO policy. If R_{np} policy has to reduce the probability of blocking high priority messages, close to referred values, it is necessary to increase c_1 . However, this leads to blocking, practically, all the low priority messages. This is the case for third class priority messages when the throughput is 500 Kbits/s.

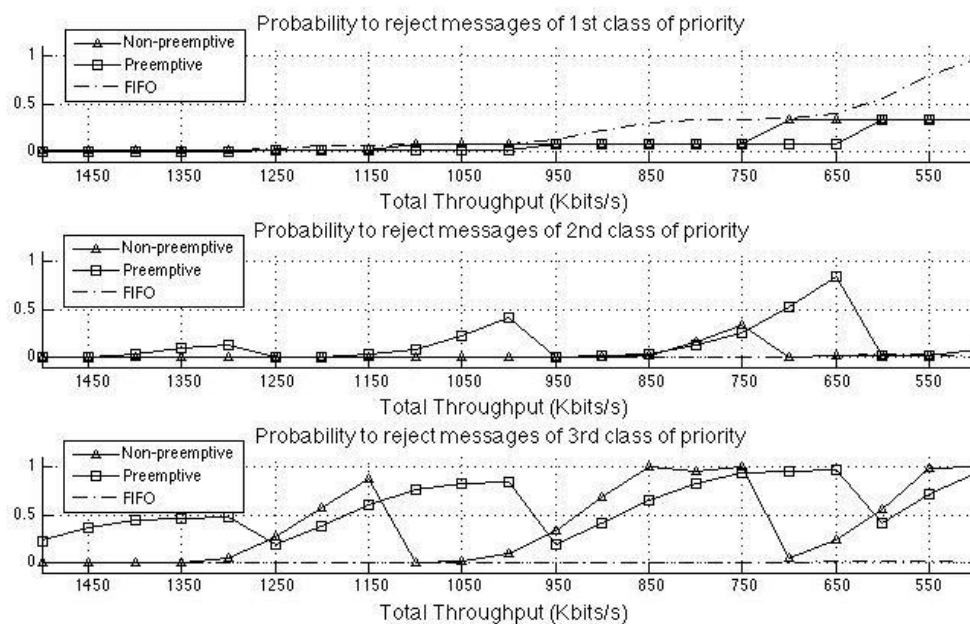


Fig. 3. Rejection probabilities when the throughput decreases from 1.5 Mbits/s to 500 Kbits/s. The lines marked with Δ indicate policy R_{np} results while the lines marked with \square indicate the policy R_p results and lines marked with \bullet indicate the FIFO policy results.

In order to evaluate the coherence of the results based on the policies and considering that preemptive policy transmits the messages as if they are the only ones in the data network, we employed M/M/m/B model from network theory [24] to determine the probability of blocking messages of high priority class. For the M/M/m/B model, λ_1 e μ_1 represent arrival and service rates respectively and m and B are calculated as $\hat{e}T/T_1\hat{u}$. Results show that R_p policy has worked as expected, since the obtained results for both the models for the first class of priority are exactly the same.

Fig. 4 shows the structure of the non-preemptive policy, R_{np} , for a total throughput of 800Kbits/s. In this situation, a high priority message needs 40.125% of the total throughput. So when more than 59.875% of throughput is occupied, it is not sufficient to accept any more high priority messages, although it is sufficient to attend other lower priority messages without affecting the probability to reject high priority messages. In the intervals between 57.9% and 59.875% and between 17.8% and 19.75% the medium priority message have to be rejected since in these intervals the probability to reject a high priority message given that a medium priority message was accepted is high. In order to decide whether to accept or reject a low priority message it is necessary to observe if it will block messages from high and medium priority classes. So it has more intervals in that it has to reject low priority messages.

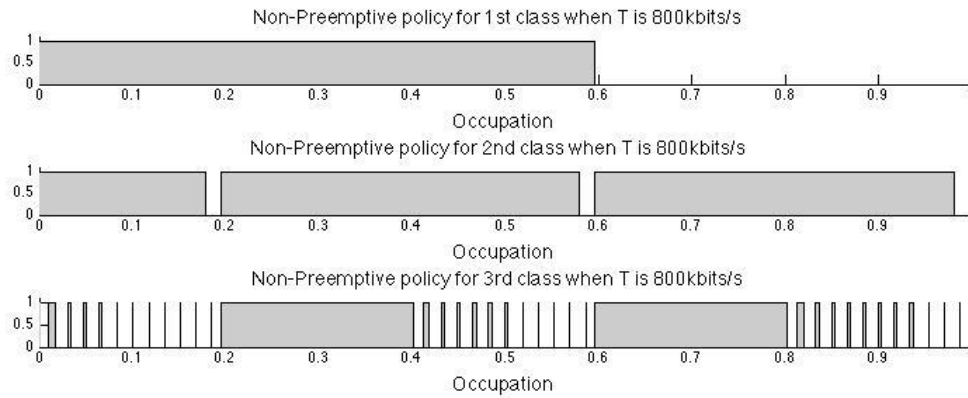


Fig. 4. Structure of the non-preemptive policy for the three classes of priority when the total throughput is 800Kbts/s. The grey rectangles denote the intervals in that the messages have to be accepted.

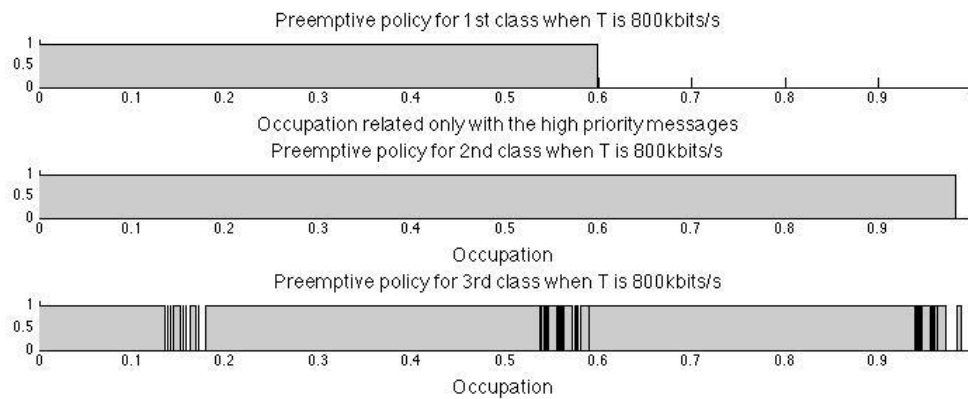


Fig. 5. Structure of the preemptive policy for the three classes of priority when the total throughput is 800Kbts/s. The grey rectangles denote the intervals in that the messages have to be accepted.

In the case of preemptive policy, R_p , the high priority messages will be accepted until 59.875% (same value that R_{np}) of the throughput is occupied, but this occupation considers only the high priority messages. In this preemptive policy, it is not necessary to observe the high priority message to decide if the lower priority messages can be accepted or not. So, the messages in the medium priority class are accepted as long as there is available throughput (similar to first class without preemption). To accept messages in the low priority class, it is necessary to observe just the medium priority class. In general terms, to allocate message of i th class of priority, one should observe the classes 2 to $i-1$, without any concern on the highest priority class 1. The structure described is demonstrated in Figure 5.

Observe that, with respect to the throughput occupation, each class of priority has some windows (intervals) in which their messages must be accepted. In the appendix, the structure of non-preemptive and preemptive policies for other values of total throughput is demonstrated. Observing these results, it is possible to conclude that, for the i th class of priority, the optimal policy has a structure:

$$W_{b_1} \in W_{e_1} \in W_{b_2} \in W_{e_2} \in \dots \in W_{b_w} \in W_{e_w} \tag{16}$$

where $W_{bi,n}$ and $W_{ei,n}$ are, respectively, the beginning and the end of the n th window of acceptance of messages of i th class of priority. w_i is the number of windows of acceptance of messages of i th class of priority.

The idea of the optimal policy is that messages of lower priority should not increase the probability to reject higher priority messages. However, depending on a certain limit, lower priority messages can be accepted as long as there is no more possibility to continue accepting messages with a high priority.

VI. FINAL REMARKS AND FUTURE WORK

This paper studied two control policies for throughput sharing in Smart Grid applications. The major contribution lies on the formulated models, in comparing the two policies, non-preemptive and preemptive policies, and in the demonstration that the preemptive policy is the most appropriate since it decreases the probability to reject high priority messages and do not reject all lower priority messages like the non-priority policy. Also the models are quite generic in the sense that they may also be applied for multi-service communication systems.

Since the messages, in the application layer, are divided in package units of the network layer protocol, the packages must be dequeued and forwarded in a sequence that satisfies the required throughput for each application. In the models and analyses conducted in this paper, we considered that elements of the network are capable to dequeue appropriately. As a future work, we propose to complement the work developed here with jitter as well as discarding of the packages based on the overflow of the buffers.

Based on the analysis of the results, it is possible to observe that there are limits that define that applications with a low priority must be rejected. On the other hand, from a certain limit, such applications can be accepted as long as there is no more possibility to continue accepting applications with a high priority. This means that available throughput is not sufficient to attend high priority applications, but there is room to attend those with low priority.

There is an important issue to be clarified. The model must be implemented and run only during the network planning. So, once it is run during the planning stage, the obtained optimal policy is converted into a table of values and will be embedded in the network elements. So, the computational effort of the network elements in which the policy is employed lies in querying a table of limits that define whether applications, with different classes of priority, are accepted or rejected.

IEC 61850 recommendations were used to determine the model parameters as they are a reference for QoS requirements for applications within the context of Smart Grids. Nevertheless, other references, if used, will not affect the model structure.

Markov Decision Processes are very useful to obtain optimal control policies of stochastic events. However, there is a drawback when considering computational effort for those problems that lead to a large number of states. The number of states increases according to its function of the number of class

priority and the total throughput as defined in eq. (1). In order to cope up with such a drawback, some algorithms from Artificial Intelligence, Reinforcement Learning and Approximate Dynamic Programming [25] are being investigated to devise solutions, optimal or sub-optimal, for large systems.

Simulators, specialized for network communications, such as OPNET or NS-3, will be used for a more detailed analysis of the system behavior under the obtained optimal policy. This detailed analysis will enable to look at the overhead of the communication protocols that may interfere in the expected results when the policy is applied to the system.

APPENDIX

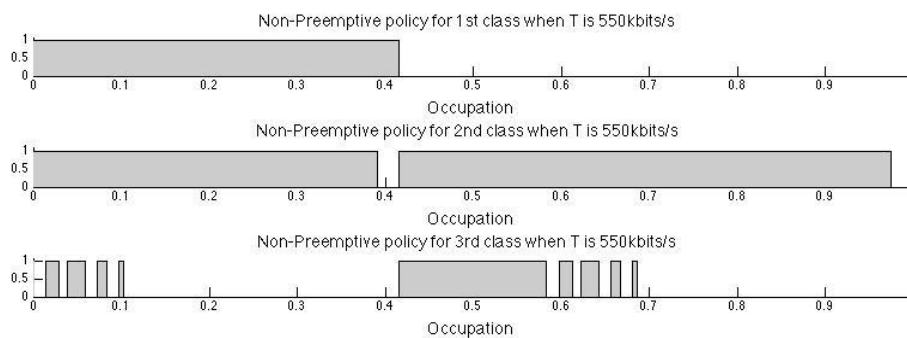


Fig. A1. Windows of acceptance for the three classes of priority using the non-preemptive policy when the total throughput is 550Kbits/s.

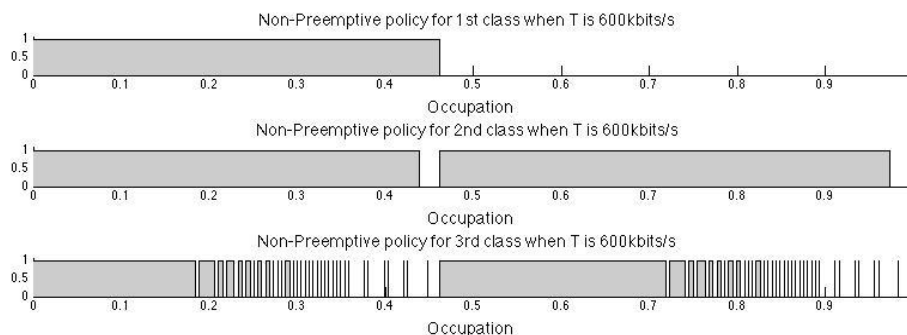


Fig. A2. Windows of acceptance for the three classes of priority using the non-preemptive policy when the total throughput is 600Kbits/s.

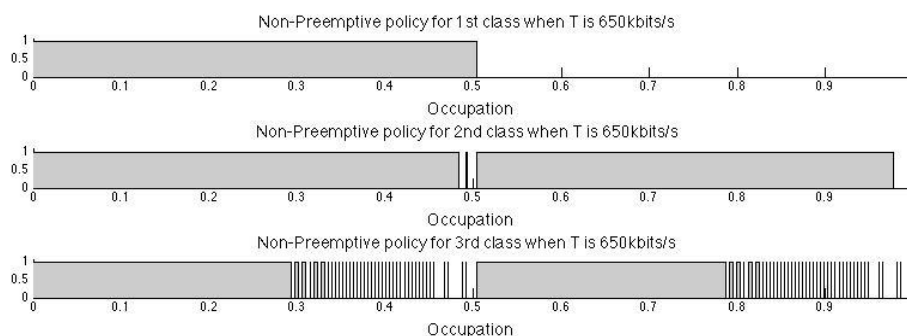


Fig. A3. Windows of acceptance for the three classes of priority using the non-preemptive policy when the total throughput is 650Kbits/s.

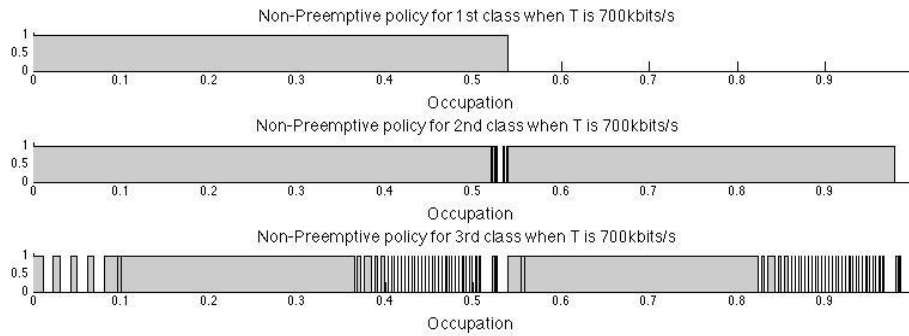


Fig. A4. Windows of acceptance for the three classes of priority using the non-preemptive policy when the total throughput is 700Kbits/s.

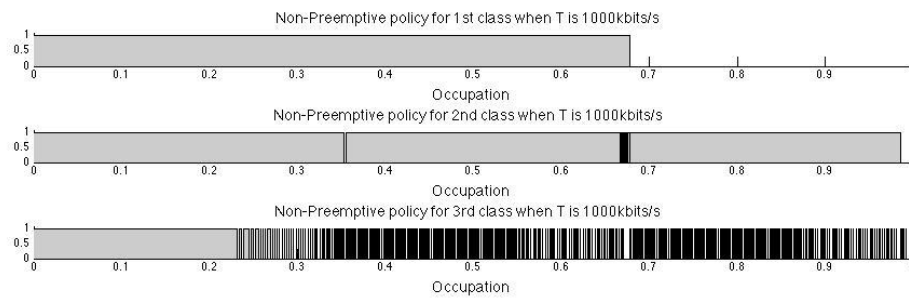


Fig. A5. Windows of acceptance for the three classes of priority using the non-preemptive policy when the total throughput is 1000Kbits/s.

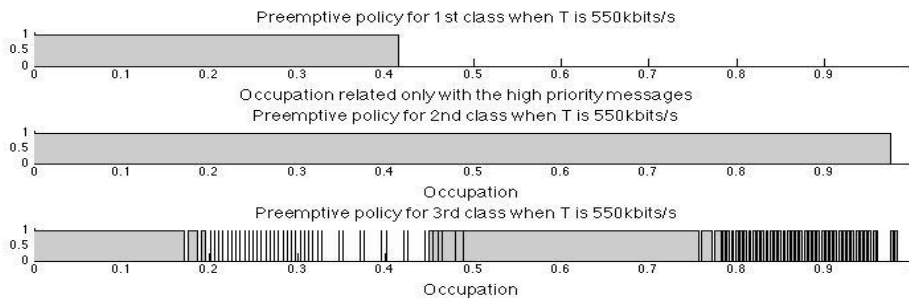


Fig. A6. Windows of acceptance for the three classes of priority using the preemptive policy when the total throughput is 550Kbits/s.

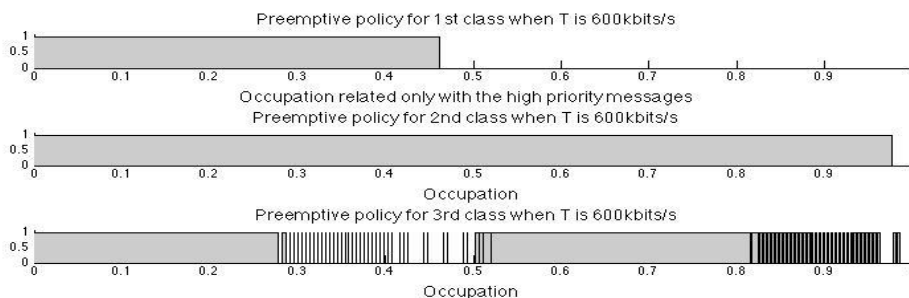


Fig. A7. Windows of acceptance for the three classes of priority using the preemptive policy when the total throughput is 600Kbits/s.

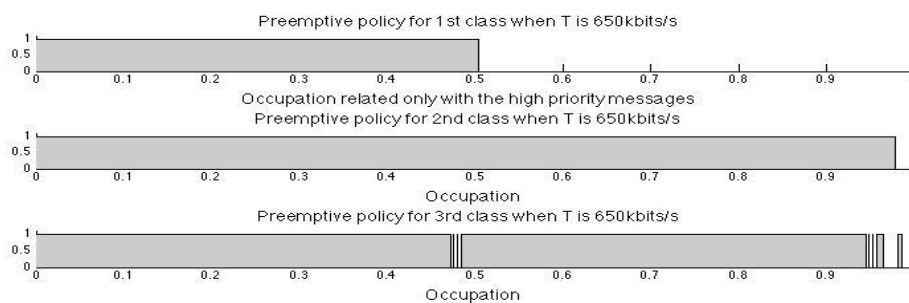


Fig. A8. Windows of acceptance for the three classes of priority using the preemptive policy when the total throughput is 650Kbits/s.

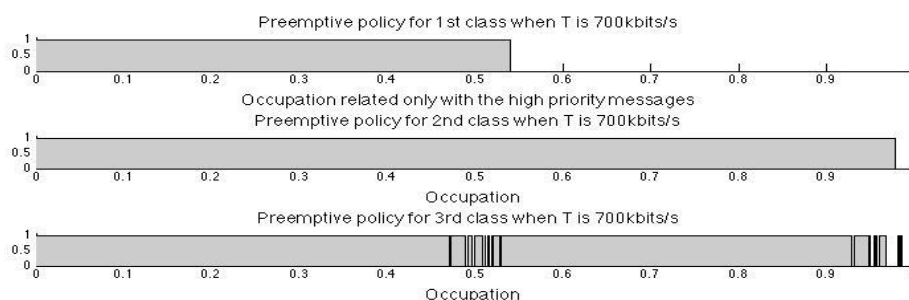


Fig. A9. Windows of acceptance for the three classes of priority using the preemptive policy when the total throughput is 700Kbits/s.

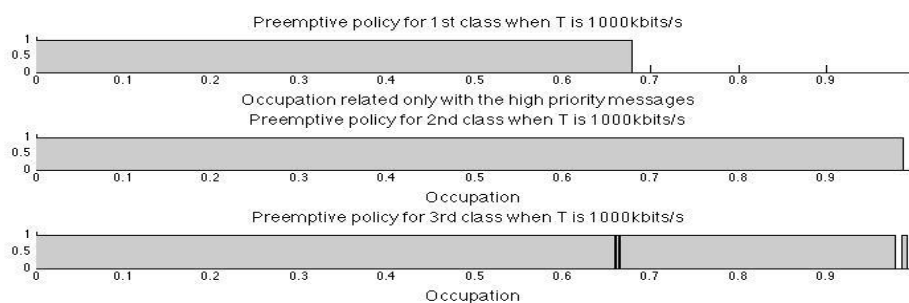


Fig. A10. Windows of acceptance for the three classes of priority using the preemptive policy when the total throughput is 1000Kbits/s.

ACKNOWLEDGMENT

The authors would like to thank CAPES (PROCAD-NF 2009 program) for its financial support.

REFERENCES

- [1] M. Sooriyabandara, J. Ekanayake. "Smart Grid – Technologies for its realization", IEEE International Conference on Sustainable Energy Technologies (ICSET), Sri Lanka, 2010.
- [2] G. W. Arnold. "Challenges and Opportunities in Smart Grid: A Position Article". Proc. of IEEE, vol. 99, no. 6, pp. 922–927, 2011.
- [3] E. Ancillotti, R. Bruno, M. Conti, "The role of communication systems in smart grids: Architectures, technical solutions and research challenges". Comput. Commun., vol. 36, no 17-18, pp. 1665-1697, 2013.
- [4] S. Latre, B. D. Vleeschauer, W. V. Meerssche, K. D. Schepper, C. Hublet, W. V. Leekwijck, F. D. Turck. "PCN Based Admission Control for Autonomic Video Quality Differentiation: Design and Evaluation". J. Netw. Syst. Manage., vol. 19, no. 1, pp. 32-57, 2011.
- [5] R. M. Salles, J. M. A. Carvalho. "An Architecture for Network Congestion Control and Charging of Non-cooperative Traffic". J. Netw. Syst. Manage., vol. 19, no. 1, pp. 367-393, 2011
- [6] P. L. Montessoro, "Efficient Management and Packets Forwarding for Multimedia Flows". J. Netw. Syst. Manage, vol. 21, no. 2, pp. 194-217, 2013.
- [7] S. Das, Y. Ohba, M. Kanda, D. Famolari, S. K. Das. "A key management framework for AMI networks in smart grid". IEEE Commun. Mag., vol. 50, no. 8, pp. 30-37, 2012.

- [8] K. Tomsovic, D. E. Bakken, V. Venkatasubramanian, A. Bose. "Designing the Next Generation of Real-Time Control, Communication, and Computations for Large Power Systems". Proc. of the IEEE, vol. 93, no. 5, pp. 965-979, 2005.
- [9] L. Du, Q. Liu. "The Design of Communication System on the Real-Time Relay Protection Based on GOOSE". Asia-Pacific Power and Energy Engineering Conference (APPEEC), China, 2012.
- [10] V. Medina, I. Gomez, J. Luque, S. Martin. "ESTELLE: a method to analyze automatically the performance of telecontrol protocols in SCADA systems". IEEE Trans. Power Deliv. vol. 17, no. 3, pp. 712- 717, 2002.
- [11] Y. Ting, Z. Zhidong, W. Jiaowen, L. Ang, "Research on Transmission Data System of Smart Grid based on IPv6 DiffServ Model". Asia-Pacific Power and Energy Engineering Conference (APPEEC), China, 2010.
- [12] M. Rahnamay-Naeini, Z. Wang, A. Mammoli, M. M. Hayat, "Impacts of control and communication system vulnerabilities on power systems under contingencies". IEEE Power and Energy Soc. Gen. Meet., USA, 2012
- [13] T. Sauter, M. Laboshov, "End-to-end communication architecture for smart grid". IEEE Trans. on Ind. Electron., vol. 58, no 4, pp. 1218- 1228, 2011.
- [14] G. Bumiller, L. Lampe, H. Hrasnica, "Power Line Communication Networks for Large-Scale Control and Automation Systems". IEEE Commun. Mag., vol. 48, no. 4, pp. 106-113, 2010.
- [15] C. Müller, H. Georg, C. Wietfeld, "A modularized and distributed simulation environment for scalability analysis of smart grid ICT infrastructures". International Conference on Simulation Tools and Techniques (SIMUTOOLS), Italy, 2012.
- [16] D. E. Bakken, A. Bose, C. H. Hauser, D. E. Whitehead, G. C. Zweigle, "Smart Generation and Transmission With Coherent, Real-Time Data". Proc. of the IEEE, vol. 99, no. 6, pp. 928-951, 2011.
- [17] Alcatel-Lucent Enterprise. Technology White Paper - Smart Choices for the Smart Grid. http://enterprise.alcatel-lucent.com/private/images/public/si/pdf_smartChoice.pdf. Accessed 03 February 2013.
- [18] A. Vallejo, A. Zaballos, J. M. Selga, J. Dalmau, "Next-generation QoS control architectures for distribution smart grid communication networks". IEEE Commun. Mag. vol. 50, no. 5, pp. 128-134, 2012.
- [19] M. Islam, H. Lee, "IEC61850 based operation, control and management of utility connected microgrid using wireless technology". International Conference on Intelligent Computing (ICIC), China, 2012.
- [20] H. C. Tijms. "Stochastic models: an algorithmic approach". John Wiley & Sons, Michigan, 1994.
- [21] Y. Yan, Y. Qian, H. Sharif, David Tripper. "A survey on Smart Grid Communication Infrastructures: Motivations, Requirements and Challenges". IEEE Communications Surveys & Tutorials, vol 15, no 1, pp. 5–20, 2013.
- [22] IEC 61850: Communication Networks and Systems in Substations. IEC standard, 2003.
- [23] National Institute of Standards and Technology. NIST Special Publication 1108 – NIST Framework and Roadmap for Smart Grid Interoperability Standards. Release 1.0, 2010.
- [24] R. Jain, "The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling," Wiley- Interscience, New York, NY, April 1991.
- [25] O. Sigaud, O. Buffét. "Markov decision Process in Artificial Intelligence". John Willey & Sons, USA, 2010.