Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Fault diagnosis in a wireless network [1]

A. D'Innocenzo, M.D. Di Benedetto and S. Di Gennaro

Department of Electrical Engineering and Computer Science, Center of Excellence DEWS. University of L'Aquila, Italy. email: (adinnoce,dibenede,digennar)@ing.univaq.it

Abstract: Given a wireless network and a graph modelling the node connection topology, we address the problem of fault diagnosis of nodes. When the external point of access to network information is at a lower layer of the ISO/OSI protocol stack, the problem is trivial. However, as often occurs, a user is able to access the network at the *application layer*: this implies that the available *observation* of the network status is considerably restricted, and solving the diagnosis problem is not trivial. In this paper, we state necessary and sufficient conditions for being able to detect which node is faulty, and propose a diagnosis algorithm on the basis of diagnosability definitions and theoretical studies developed for timed and hybrid automata in the computer science community.

Keywords: Wireless networks, fault diagnosis, verification.

1. INTRODUCTION

Recently there has been a growing interest in research on the interaction between control and wireless communication (Di Benedetto *et al.*, 2006*a*; Marco *et al.*, 2006; Liu and Goldsmith, 2003; Liu and Goldsmith, 2004; Santucci and Graziosi, 1999; Sgroi *et al.*, 2003; Sinopoli *et al.*, 2005). In particular, distributed sensing and control where the communication infrastructure is offered by wireless networks and control methods for energy efficient operations on wireless networks are two general topics of research interest. However, the problem of energy consumption can only be alleviated, since wireless nodes are often equipped with a tiny energy source and are usually destined to discharge in short time. An interesting application of wireless networks is the monitoring of wide geographic areas, e.g. to collect geological measures on a landslide or sample rainfall distribution. Once the network has been positioned and activated, maintenance assumes a important role, e.g. for replacement of discharged or faulty nodes. A diagnosis protocol might be implemented in the lower layers of the ISO/OSI protocol stack (e.g. at the routing level - network layer): in this case, each node is aware of faults in the neighbors, and can communicate the address to the application layer. However, working at the application layer, often one can access a wireless network as a black box, using services of the underlying protocol stack that are essentially *queries* to a destination node address. Routing is hidden in the lower layers. The only feedback to a query command is a positive answer after a measurable finite time, or no answer if the node is not reachable.

For these reasons, we model a wireless network by a graph defining the node connection topology, and a relay time estimate associated to each node. We assume that our interface with the network is a gateway node that accepts **ping** commands (the control input), and replies with **pong** responses after a time delay that we can measure (the observed output). For addressing the diagnosability problem of a wireless network, we use a variation of the theoretical framework on diagnosability of timed (Tripakis, 2002) and hybrid automata (Di Benedetto *et al.*, 2007*a*). Diagnosability of timed and hybrid automata is a generalization of the observability property as defined in (D'Innocenzo *et al.*, 2006), and corresponds to failure detection in finite time. Diagnosability has many applications, e.g. the detection of an error in an air traffic management procedure (Di Benedetto *et al.*, 2005; Di Benedetto *et al.*, 2006*b*), the detection of a failure in an automotive system (Fourlas *et al.*, 2002; Di Benedetto *et al.*, 2007*b*), in a component of an industrial plant, or in a communication system (Sheth *et al.*, 1999). A system is diagnosable if, within a finite-time bound and only using the observable output, it is possible to detect whether a fault has occurred, that is if a system execution visits a given *faulty* subset of the state space. In (Di Benedetto *et al.*, 2007*b*) we addressed diagnosability of hybrid automata and durational graphs, with output given by discrete output symbols associated to the discrete transitions and delays between observed output symbols. We will use these theoretical results to establish whether a given node is faulty or not, and propose a diagnosis algorithm.

The paper is organized as follows: in Section 2 we introduce the mathematical framework, in Section 3 we show how to apply it for solving the diagnosis problem on a wireless network. We investigate in Section 4 the Zigbee protocol, to check the possibility of implementing our algorithm in a real device. Conclusions are offered in Section 5.

2. BASIC DEFINITIONS

Referring to (Di Benedetto *et al.*, 2007*b*) for more details, we informally give basic definitions of durational graphs

[1] This work was partially supported by the HYCON Network of Excellence, contract number FP6-IST-511368, and by Ministero dell'Istruzione, dell'Universita' e della Ricerca under Project SCEF (PRIN05)

and corresponding languages of executions and observations.

We call *durational graph* (Di Benedetto *et al.*, 2007*b*) a timed automaton (Alur and Dill, 1994) characterized by only one clock that is reset to 0 for all edges. A durational graph can be uniquely identified by a tuple $\mathcal{G} = (Q, Q_0, E, \Psi, \eta, Inv, G)$. Q is the discrete state space, where finite cardinality $|Q| = N$ and initial condition $Q_0 \subseteq Q$. $v \in \mathbb{R}_+ \cup \{0\}$ is the only clock variable, with continuous dynamics $\dot{x} = 1$ and initial condition $x(0) = 0$. $E \subseteq Q \times Q$ is a collection of edges, where each edge $e \in E$ is an ordered pair of discrete states. Ψ is the finite set of discrete output symbols $\{\varepsilon, \psi_1, \psi_2, \cdots, \psi_r\}$, where ε is the unobservable output, that corresponds to the empty string. $\eta : E \to \Psi$ is the output function, that associates to each edge a discrete output symbol. $\{Inv_q\}_{q\in Q}$ associates to each discrete state a rectangular [2] invariant set $Inv_q \subseteq X$, and $\{G_e\}_{e\in E}$ associates to each edge a rectangular guard set $G_e \subseteq Inv_{s(e)}$. This class of timed automata is in general non deterministic. The clock evolves following deterministic dynamics, and the discrete state evolution depends only on the clock according to guards, possibly with non deterministic behaviors in the discrete transitions.

Given a durational graph $\mathcal{G}$, we define a timed string ρ as a sequence of pairs $\{(q_k, \Delta_k)\}_{k\geq 0}$ with cardinality $|\rho|$, where q_k denotes the discrete state after k switchings and Δ_k denotes the dwelling time in q_k. We define the timed language of executions of the discrete state of $\mathcal{G}$ is given by $\mathcal{L}$. An example of timed execution $\rho \in \mathcal{L}$ is $q_1, 3, q_2, 6, q_3, \cdots$. Given a subset of discrete states $Q^* \subseteq Q$, we define $\mathcal{L}_{Q^*}$ the language of executions with finite cardinality, such that the last visited discrete state belongs to Q^*. Given an execution $\rho = \{(q_k, \Delta_k)\}_{k=0}^{|\rho|}$, we introduce the following notations:

- $\rho|_i = q_i$ is the discrete state in the time interval I_i of the execution associated to ρ;
- $\rho|_i^j = q_i, \Delta_i, \cdots, q_j, \Delta_j$ is the substring of ρ from index i to j;
- $time(\rho) = \sum_{k=0}^{|\rho|} \Delta_k$ is the time duration of ρ.

Given an execution $\rho = \{(q_k, \Delta_k)\}_{k=0}^{|\rho|}$, we define the associated output string as

$$\Delta_0, \eta\big((q_0, q_1)\big), \Delta_1, \eta\big((q_1, q_2)\big), \Delta_2, \cdots$$

The associated *observation* $P(\rho)$ is obtained from the output by erasing all ε (unobservable) symbols and adding up the adjacent time delays. For instance, the output string $3, \psi_1, 4, \varepsilon, 5, \psi_2, 2$ is observed as $3, \psi_1, 9, \psi_2, 2$.

Given a hybrid automaton $\mathcal{H}$, let $Q_c \subset Q$ be a set of discrete states that model a failure in $\mathcal{H}$: Q_c is called faulty set. A δ–faulty execution is a trajectory that enters the faulty set at a certain time instant, and then continues flowing for a time duration δ.

Definition 1. (δ–faulty execution). An execution $\rho \in \mathcal{L}$ is δ–faulty if there exists a finite index $k_c, 0 \leq k_c \leq |\rho|$ such that:

$$\forall k < k_c,\ \rho|_k \notin Q_c;\ \rho|_{k_c} \in Q_c;\ time(\rho|_{k_c}^{|\rho|}) = \delta.$$

For any faulty execution ρ, we use the notation $\rho|_{k_c}$ to denote the first faulty state visited by ρ. We define $\mathcal{F}_\delta$ the set of all δ–faulty executions, and $\mathcal{F} = \bigcup_{\delta\geq 0} \mathcal{F}_\delta \subseteq \mathcal{L}$ the set of all faulty executions. We say that a set Q_c is δ–diagnosable for a system $\mathcal{H}$ if it is possible to detect within a delay upper bounded by δ whether an execution has visited the faulty set, only using the observable output. It is straightforward to state necessary and sufficient δ–diagnosability conditions:

Proposition 1. A set Q_c is δ–diagnosable for $\mathcal{H}$ if and only if

$$\forall \rho \in \bigcup_{\delta^*\geq\delta} \mathcal{F}_{\delta^*},\ \forall \rho' \in \mathcal{L} \setminus \bigcup_{\delta^*\geq\delta} \mathcal{F}_{\delta^*},\ P(\rho) \neq P(\rho')$$

This is equivalent to say that for any observation of the system, it is possible to determine whether the associated execution is δ^*–faulty with $\delta^* \geq \delta$, or it is not. Notice that δ–diagnosability is much more general than discrete state observability as defined in (D'Innocenzo *et al.*, 2006). A set Q_c is defined observable for a system $\mathcal{H}$ if it is possible to *immediately* detect using the observable output whether the current discrete state is visiting Q_c.

Proposition 2. (D'Innocenzo *et al.*, 2006) A set Q_c is observable for $\mathcal{H}$ if

$$\forall \rho \in \mathcal{L}_{Q_c},\ \forall \rho' \in \mathcal{L}_{Q\setminus Q_c},\ P(\rho) \neq P(\rho')$$

Proposition 3. Q_c is observable if and only if Q_c is 0–diagnosable.

Thus observability is a particular case of diagnosability. If a system is diagnosable for some finite δ, the following property shows that it is very interesting to compute the minimum value δ_m for which $\mathcal{H}$ is δ_m–diagnosable:

Proposition 4. Given $\mathcal{H}$, the following statements hold:

(1) If Q_c is δ–diagnosable, then it is δ^*–diagnosable for all $\delta^* \geq \delta$.
(2) If Q_c is not δ–diagnosable, then it is not δ^*–diagnosable for all $\delta^* \leq \delta$.

The verification algorithm presented in (Di Benedetto *et al.*, 2007*b*) consists of two parts. In the first part, the tricky one, we deal with edges associated to an unobservable output symbol: we proposed an algorithm to construct a durational graph without unobservable outputs, that preserves diagnosability. In the second part, we proposed a verification algorithm for systems that do not generate unobservable outputs. We proved that the δ–diagnosability verification problem for the class of durational graphs belongs to the complexity class P, namely it can be solved in polynomial time. Moreover, as discussed above, we showed how to compute the minimum value δ_m for which Q_c is δ_m–diagnosable. The main results can be resumed in the following theorem:

Theorem 1. (Di Benedetto *et al.*, 2007*b*) Given a durational graph $\mathcal{G}$ and a faulty set Q_c, it is possible to compute the minimum value δ_m for which Q_c is δ_m–diagnosable in polynomial time with respect to $N = |Q|$.

[2] a rectangular set in $\mathbb{R}^n$ is any subset that can be defined by a finite union of cartesian products of intervals.

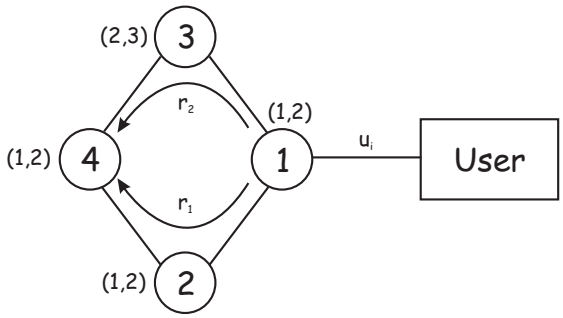

Fig. 1. Sensor network graph structure.

3. FAULT DIAGNOSIS OF NODES

A wireless network can be modeled as in Figure 1 by a graph given by a set of N nodes and a set of edges connecting the nodes by a radio link. We associate to each node j a range of possible latency times as an open rectangular interval $(t_{min}(j), t_{max}(j))$, that models the time needed for the node to relay a packet. The node labeled by 1 is the gateway, namely we can access the network through node 1 sending a ping command u_i to interrogate node i. If node i is reachable, a pong response is received with a time delay given by the sum of the latency times in each node of the round trip routed path connecting nodes 1 and i. In Figure 1 the round trip delays corresponding to routing paths r_1 and r_2 are respectively $(5, 10)$ and $(7, 12)$: if all nodes are working correctly, the latency of a response to an input u_4 is given by the time interval $(5, 12)$. If a node i is not reachable because of faults in the network, the command u_i will receive no answer. Since there is a maximum latency time t_M due to the network topology, if no answer is received after t_M then node i is not reachable. In our example, a command u_4 will receive no answer if one of the following holds: (1) node 1 is faulty, (2) node 4 is faulty, (3) nodes 2 and 3 are faulty. The maximum latency time for the input u_4 is 12 time units.

It is clear that the latency time of the network for an input command u_i depends on the set of nodes that are faulty. To model the dynamical behavior of faults in a network with N nodes, we consider a durational graph $\mathcal{G}$ where the set Q of discrete states is the power set of all possible 2^N combinations of faulty nodes. Clearly, the empty set $\{\varnothing\} \in Q$ models absence of faulty nodes, and the set $\{1, \cdots, N\} \in Q$ models a fault in each node. The set of outputs $\Psi = \{u_1, \cdots, u_N\}$ is given by all commands that can be sent to the network. Edges between different states model a new fault in the network, and are associated to a guard $[0, \infty)$ (namely a fault might occur at any time instant), and to an unobservable output ε. Self-loop edges $e = (q, q), \eta(e) = u_i$ model the result of an input command u_i, and are associated to a guard given by the latency time of the network to a command u_i: to each discrete state q (that corresponds to a combination of faulty nodes) is associated a different latency time G_e. For the computation of G_e, we state the following assumptions:

(1) The routing paths do not contain cycles [3].
(2) If there exists at least a path from the gateway to the destination node, then it is used.
(3) The routing path of ping and pong is the same.
(4) Faults do not occur during a diagnosis procedure, namely the diagnosis is much faster than the expected value of the "hitting time" of a new node fault.

[3] The theoretical results are not affected by this assumption, but we stated it since it is undesired for routing algorithms to perform cycles.

Since the algorithmic construction of $\mathcal{G}$ from the network is very simple, we directly show an example for the network in Figure 1. Since $N = 4$, then $|Q| = 2^4 = 16$: for the sake of simplicity and without loss of generality, we will consider $Q = \{\{\varnothing\}, \{1\}, \{2\}, \{3\}, \{4\}\}$, namely only states that correspond to no more than one faulty node. Considering all 16 possible states can be achieved by trivial iteration.

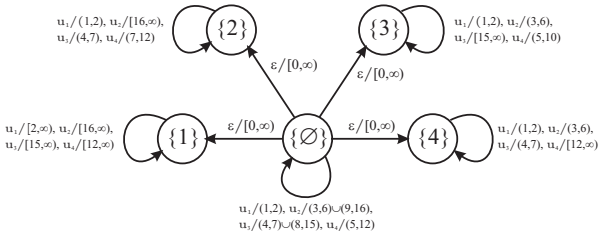

Fig. 2. Durational graph $\mathcal{G}$ obtained from the network in Figure 1.

Informally, we say that the fault of node i is diagnosable if there exists a sequence of inputs

$$u = \{u(k)\}_{k=1}^{n}, u(k) \in \Psi, n \leq N$$

such that it is possible to establish, using the latency time delays of each command, whether node i is faulty or it is not. Because of Assumption (4), transitions between different states of $\mathcal{G}$ cannot occur during the diagnosis procedure, thus we can infer formal diagnosability conditions on the durational graph $\mathcal{G}^*$ given by $\mathcal{G}$ deprived of unobservable transitions. Let $Q_i \subset Q$ be the set of states of $\mathcal{G}^*$ that model a fault of node i, namely

$$Q_i \triangleq \{q \in Q : i \in q\}$$

Proposition 5. A node i is diagnosable if and only if

$$\exists u = \{u(k)\}_{k=1}^{n}, n \leq N : \\ \forall \rho \in \mathcal{L}^u_{Q_i}, \forall \rho' \in \mathcal{L}^u_{Q \setminus Q_i}, P(\rho) \neq P(\rho') \qquad (1)$$

With abuse of notation, $\mathcal{L}^u_{Q_i}$ is the set of executions $\rho \in \mathcal{L}_{Q_i}$ such that the corresponding observation $P(\rho)$ deprived of the time delays is equal to u. The computation of $\mathcal{L}^u_{Q_i}$ and $\mathcal{L}^u_{Q \setminus Q_i}$ is straightforward given $\mathcal{G}^*$ and u. To verify if a node is diagnosable, a trivial algorithm consists of searching over all 2^N combinations of control inputs u. For each input string u, condition (1) is equal to 0-diagnosability condition on the system $\mathcal{G}^*$ restricted to the executions triggered by u, and thus the theoretical computational framework developed in (Di Benedetto *et al.*, 2007*b*) and illustrated in this paper can be used to perform the verification in polynomial time. Current work aims to find the minimum number of inputs to perform a diagnosis of a node i using direct analysis of the graph associated to the network. In our example, node 4 is diagnosable with a diagnosis input u_3u_4 of length 2.

4. FAULT DIAGNOSIS ON ZIGBEE

In this section, we investigate the Zigbee protocol to check the possibility to implement our algorithm in a

real device. The standard specification IEEE 802.15.4 defines the protocol and the compatible connections for communication devices that require low data rate, low power, and short ray/low complexity RF transmissions, in the framework of a Wireless Personal Area Network (WPAN). More precisely, it specifies the physical and MAC layers of the ISO/OSI protocol stack. The IEEE 802.15.4 does not require infrastructures, thus it can be used for communications within a maximum distance of 10 meters. For this reason, the standard guarantees easy installation, low cost, and reasonable battery duration of the devices. The interconnection graph of a Zigbee

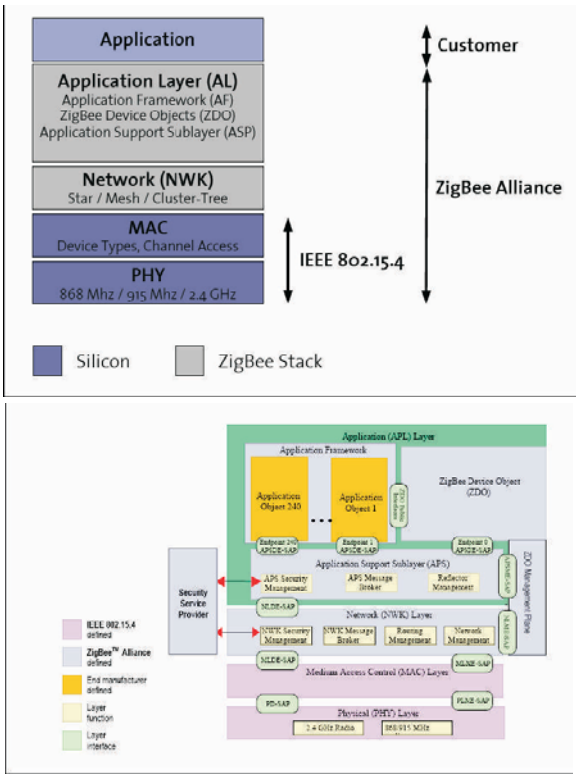

Fig. 3. Zigbee protocol stack.

compliant network can be one of the following:

- Star,
- Tree,
- or Mesh.

As already discussed, a fault may occur in a node because of physical damage, battery exhausted, software bugs, etc. The idea is performing diagnosis using standard services and primitives offered by the ZDO sublevel (ZigBee Device Objects) and by the ZDO Management Plane, which define the interface between the Network level (NLME-SAP) and the Application level (APSME-SAP). Using NLME primitives implemented in the ZDO, it is possible to obtain information on the status of nodes. Complying with the theoretical framework illustrated in the above sections, these primitives can be used to send to the Network layer a **ping** request to a certain node/endpoint. The obtained response primitive **Route Error Reporting** allows the Network layer to inform the higher levels about node faults. The returned parameters are the following:

- **Short Address** is a 16-bit network address of the destination node associated to a routing failure. Notice that if a node cannot be reached, it does not mean that it is faulty: on the contrary, it means that at least one node on the route path to reach it is faulty.
- **Status** takes value in a finite set of error codes:
  (1) *No route available*: no path is available to reach the destination node.
  (2) *Tree link failure*: routing failed since no tree path has been found.
  (3) *Non-tree link failure*: a tree path for routing has been found, but still there has been a routing error.
  (4) *Low battery level*: the battery level of a destination node is very low.
  (5) *No routing capacity*: a destination node has no routing capacity.
  (6) *No indirect capacity*: the buffer of a destination node has no available capacity.
  (7) *Indirect transaction expiry*: time-out error in the buffer of a destination node.
  (8) *Target device unavailable, Target address unallocated*: for the objective of the paper, it is sufficient to define these codes as SW failures in a destination node.
  (9) *Parent link failure*: RF failure of a destination node.
  (10) *Validate route*: invalid multicast route address.
  (11) *Source route failure*: failure of a routing path.
  (12) *Many-to-one route failure*: failure of a many-to-one routing request.

The diagnosis algorithm formalized in the previous sections can be implemented at the Application layer, and takes as observation of the network the error codes (that do not specify the exact address of the faulty node!). By computing the time delay between a **ping** to a node and the error code message, it is possible to perform a diagnosis to detect which node is faulty, and to determine if it must be recharged, repaired or replaced.

5. CONCLUSION

In this paper, we addressed a fault diagnosis problem on a wireless network modeled by a graph defining the node connection topology, and a relay time estimate associated to each node. We assumed that our interface with the network is a gateway node that accepts **ping** commands (control input), and replies with **pong** responses after a time delay that we can measure (observed output). We stated necessary and sufficient conditions for being able to detect which node is faulty and propose a diagnosis algorithm, on the basis of diagnosability definitions and theoretical studies developed for timed and hybrid automata in the computer science community. We finally investigated the Zigbee specification to check the possibility to implement our algorithm on a real communication system.

REFERENCES

Alur, R. and D.L. Dill (1994). A theory of timed automata. *Theoretical Computer Science* **126**, 183–235.

Di Benedetto, M. D., A. D'Innocenzo, G. Pola, C. Rinaldi and F. Santucci (2006*a*). Modeling of adaptive behaviours in control over wireless networks. In:

Proceedings of the 17^{th} *International Symposium on Mathematical Theory of Network and Systems. Kyoto, Japan.*

Di Benedetto, M. D., S. Di Gennaro and A. D'Innocenzo (2005). Error detection within a specific time horizon and application to air traffic management. In: *Proceedings of the Joint* 44^{th} *IEEE Conference on Decision and Control and European Control Conference (CDC–ECC'05), Seville, Spain.* pp. 7472–7477.

Di Benedetto, M. D., S. Di Gennaro and A. D'Innocenzo (2006*b*). Critical states detection with bounded probability of false alarm and application to air traffic management. In: *Proceedings of the* 2^{nd} *IFAC Conference on Analysis and Design of Hybrid Systems (ADHS), Alghero, Sardinia, Italy.*

Di Benedetto, M. D., S. Di Gennaro and A. D'Innocenzo (2007*a*). Diagnosability verification for hybrid automata. In: *Hybrid Systems: Computation and Control.* Lecture Notes in Computer Science. Springer Verlag.

Di Benedetto, M.D., S. Di Gennaro and A. D'Innocenzo (2007*b*). Diagnosability verification for hybrid automata and durational graphs. In: *Proceedings of the* 46^{th} *IEEE Conference on Decision and Control. New Orleans, Louisiana, USA.*

D'Innocenzo, A., M. D. Di Benedetto and S. Di Gennaro (2006). Observability of hybrid automata by abstraction. In: *Hybrid Systems: Computation and Control* (J. Hespanha and A. Tiwari, Eds.). Vol. 3927 of *Lecture Notes in Computer Science.* pp. 169–183. Springer Verlag.

Fourlas, G. K., K. J. Kyriakopoulos and N. J Krikelis (2002). Diagnosability of hybrid systems. In: *Proceedings of the 10th Mediterranean Conference on Control and Automation - MED2002, Lisbon, Portugal.* pp. 3994–3999.

Liu, X. and A. Goldsmith (2003). Wireless communication tradeoffs in distributed control. In: *Proceedings of the IEEE Conference on Decision and Control.*

Liu, X. and A. Goldsmith (2004). Wireless medium access control in networked control systems. In: *Proceedings of the IEEE American Control Conference.*

Marco, P. Di, C. Rinaldi, F. Santucci, K.H. Johansson and N. Möller (2006). Performance analysis and optimization of tcp over adaptive wireless links. In: *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications. Helsinki, Finland.*

Santucci, F. and F. Graziosi (1999). Power allocation in a multimedia cdma wireless system with imperfect power control. In: *Proceedings of the IEEE International Conference on Commununications. Vancouver, Canada..* pp. 192–194.

Sgroi, M., A. Wolisz, A.L. Sangiovanni-Vincentelli and J. Rabaey (2003). A service-based universal application interface for ad-hoc wireless sensor networks. White paper, berkeley wireless research center. berkeley, california, usa.. UC Berkeley, EECS.

Sheth, A., C. Hartung and R. Han (1999). A decentralized fault diagnosis system for wireless sensor networks. In: *Proceedings of the* 2^{nd} *IEEE International Conference on Mobile Ad-Hoc and Sensor Systems (MASS) 2005.* pp. 192–194.

Sinopoli, B., L. Schenato, M. Fraschetti, K. Poolla and S. Sastry (2005). An lqg optimal linear controller for control systems with packet losses. In: *Proceedings of the IEEE conference on decision and control and european control conference. Sevilla, Spain.*

Tripakis, S. (2002). Fault diagnosis for timed automata. *Lecture Notes in Computer Science, W. Damm and E.R. Olderog Eds., Springer Verlag* **2469**, 205–221.