Supervisory Control for the AODV Routing Protocol

Nikolaos D. Kouvakas and Fotis N. Koumboulis
Department of Digital Industry Technologies
School of Science
National Kapodistrian University of Athens
Psahna Evias, Greece
nkouvak@dind.uoa.gr, fkoumboulis@dind.uoa.gr

Dimitrios G. Fragkoulis and Nikolaos I. Georgiou
Core Department
National Kapodistrian University of Athens
Psahna Evias, Greece
dfragkoulis@uoa.gr, nikosgeor92@hotmail.com

***Abstract*— The model of the AODV routing protocol, being favorable in several wireless networks (VANET, ZigBee etc.) is presented for a network with parametric number of nodes in the Ramadge Wonham (RW) framework of discrete event systems. Simulation of the automaton of the protocol, for a given four-node network, is provided. For a generic network consisting of *n* nodes, the goal is to introduce a first simple hierarchy of the requests of the nodes and the routing paths. To this end, a specification is introduced to express the desired hierarchy. The specification of the hierarchy is presented in the form of a simple rule. The rule is translated to a regular language. A supervisor automaton is designed to realize the desired regular language. The nonblocking property of the request flow between the nodes of the network and the satisfactory marked behavior of the controlled automaton are proved. To illustrate the effectiveness of the present control scheme, a four-node network simulation of the controlled automaton of the protocol is provided.**

Keywords — Discrete Event Systems, Supervisory Control, Communication Protocols, Wireless Networks

I. Introduction

The Ad-hoc on Demand Distance Vector (AODV) protocol is usually met in VANET networks as well other wireless ad hoc networks (e.g. ZigBee). The protocol is of particular interest for robotic vehicles. The operation of AODV protocol is analyzed to the following steps (see [1]-[4]): Until a new connection is requested, the protocol is in idle mode. When a new route is required, then the source node (e.g., a requesting vehicle) transmits a request message. When the destination node (e.g., a destination vehicle) receives the request, it sends back a reply package to the source node. The source then initiates the connection between the two nodes. The routing table entries are removed from the queue (deleted) after a predefined period (see also [1]-[4]). In most cases, a network using AODV routing protocol consists of many interactive nodes. A promising issue, in the field of communication protocols, appears to be modelling and supervisor control of the protocols using discrete event systems (see [5]).

In the present paper, a network, with a parametric number of nodes and using AODV protocol, will be studied. The model of the AODV routing protocol, using discrete event systems, will be presented. This is the first contribution of the paper. The goal of the paper is to attempt to introduce a first simple hierarchy in the request/answer flow of the network. Here, this hierarchy among the nodes of the network and with respect to the time of the submission of each request, is developed using the supervisory control theory and specifically the RW framework (see [6]-[7]). A single supervisor will be designed for every node to guarantee the desired hierarchy of routing requests for all nodes. This is the second contribution of the paper. The nonblocking property of the total controlled automaton will be proved. This is the third contribution of the paper. To validate the effectiveness of the proposed control scheme, simulation results for a four-node network will be presented, for both the uncontrolled and the controlled protocol.

The paper is organized as follows: In Section II, the discrete event model of the i-th node, where $i \in \{1,...,n\}$ and n is the number of the nodes of the network, of an AODV routing protocol is presented. Also, the model of the parametric n node network is presented. Finally, in this section, simulation of the system for a four-node network is provided. In Section III, the desired specifications for the hierarchy of the routing requests and routing paths are formulated in the form of a regular language. Appropriate supervisor is designed and the nonblocking property of the total controlled automaton is presented. Simulation of the controlled system for a four-node network is presented.

II. THE AODV ROUTING PROTOCOL

A. *Modeling of the AODV Routing Protocol*

The automaton describing the i-th node of an AODV Routing Protocol (see [1]-[2]) is in the 6-tuple form (see [5], [8]-[12])

$$ {}^{i}\mathbf{G}_N = ({}^{i}\mathbb{Q}_N, {}^{i}\mathbb{E}_N, {}^{i}f_N, {}^{i}\mathbb{H}_N, {}^{i}x_{N,0}, {}^{i}\mathbb{Q}_{N,m}) . $$

The set of the states of the i-th node is

$$ {}^{i}\mathbb{Q}_N = \{ {}^{i}q_{N,1}, {}^{i}q_{N,2}, {}^{i}q_{N,3}, {}^{i}q_{N,4} \} . $$

The state ${}^{i}q_{N,1}$ is the idle state. The state ${}^{i}q_{N,2}$ is the case where a Route Request Packet (RREQ) has been transmitted. The state ${}^{i}q_{N,3}$ is the case where a Route Request Packet or a Route Receive Packet has been received. The state ${}^{i}q_{N,4}$ is the case where a Route Receive Packet has been created. The alphabet is

$$ {}^{i}\mathbb{E}_N = \{ {}^{i}e_{N,1}, {}^{i}e_{N,2}, {}^{i}e_{N,3}, {}^{i}e_{N,4}, {}^{i}e_{N,5}, {}^{i}e_{N,6} \} . $$

The command to transmit a Route Request Packet (RREQ) corresponds to the event ${}^{i}e_{N,1}$. The reception of a Route Receive Packet (RREP) corresponds to the event ${}^{i}e_{N,2}$. The receipt of a RREQ or a RREP corresponds to the event ${}^{i}e_{N,3}$. The transmission of a RREQ or a RREP corresponds to the

979-8-3503-9958-5/22/$31.00 ©2022 IEEE

event ${}^{i}e_{N,4}$. The command to create of packet, if the node can transmit a packet, corresponds to event ${}^{i}e_{N,5}$. The command to transmit a RREP corresponds to the event ${}^{i}e_{N,6}$. Obviously, all the events can be deactivated. Thus, they are all controllable events, i.e., ${}^{i}\mathbb{E}_{N,c} = {}^{i}\mathbb{E}_{N}$ and ${}^{i}\mathbb{E}_{N,uc} = \varnothing$.

The active event sets per state of the automaton ${}^{i}\mathbf{G}_{N}$ are:

$$ {}^{i}\mathbb{H}_N({}^{i}q_{N,1}) = \{{}^{i}e_{N,1}, {}^{i}e_{N,3}\},\ {}^{i}\mathbb{H}_N({}^{i}q_{N,2}) = \{{}^{i}e_{N,2}\}, $$
$$ {}^{i}\mathbb{H}_N({}^{i}q_{N,3}) = \{{}^{i}e_{N,4}, {}^{i}e_{N,5}\},\ {}^{i}\mathbb{H}_N({}^{i}q_{N,4}) = \{{}^{i}e_{N,6}\}. $$

The values of the transition function of all states and events of ${}^{i}\mathbf{G}_{N}$ are

$$ {}^{i}f_N({}^{i}q_{N,1}, {}^{i}e_{N,1}) = {}^{i}q_{N,2},\ {}^{i}f_N({}^{i}q_{N,1}, {}^{i}e_{N,3}) = {}^{i}q_{N,3}, $$
$$ {}^{i}f_N({}^{i}q_{N,2}, {}^{i}e_{N,2}) = {}^{i}q_{N,1},\ {}^{i}f_N({}^{i}q_{N,3}, {}^{i}e_{N,4}) = {}^{i}q_{N,1}, $$
$$ {}^{i}f_N({}^{i}q_{N,3}, {}^{i}e_{N,5}) = {}^{i}q_{N,4},\ {}^{i}f_N({}^{i}q_{N,4}, {}^{i}e_{N,6}) = {}^{i}q_{N,1}. $$

The initial state of ${}^{i}\mathbf{G}_{N}$ is the idle state i.e., ${}^{i}x_{N,0} = {}^{i}q_{N,1}$. The set of the marked states of ${}^{i}\mathbf{G}_{N}$ is ${}^{i}\mathbb{Q}_{N,m} = \{{}^{i}q_{N,1}\}$.

The closed behavior of the automaton is

$$ \mathbb{L}({}^{i}\mathbf{G}_N) = \overline{\left(({}^{i}e_{N,1}\,{}^{i}e_{N,2})^{*}({}^{i}e_{N,3}({}^{i}e_{N,4} + {}^{i}e_{N,5}\,{}^{i}e_{N,6}))\right)^{*}}. $$

The marked behavior of the automaton is

$$ \mathbb{L}_m({}^{i}\mathbf{G}_N) = \left(({}^{i}e_{N,1}\,{}^{i}e_{N,2})^{*}({}^{i}e_{N,3}({}^{i}e_{N,4} + {}^{i}e_{N,5}\,{}^{i}e_{N,6}))\right)^{*}. $$

Obviously, it holds that $\mathbb{L}({}^{i}\mathbf{G}_N) = \overline{\mathbb{L}_m({}^{i}\mathbf{G}_N)}$. Thus, the automaton ${}^{i}\mathbf{G}_{N}$ is a nonblocking automaton. In Figure 1, the state diagram of the automaton ${}^{i}\mathbf{G}_{N}$ is presented.

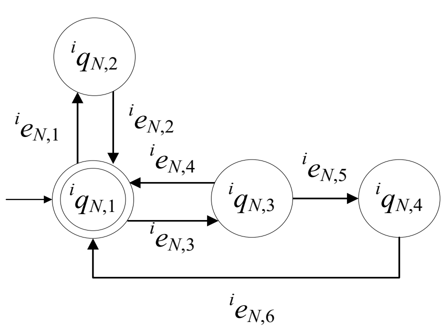

Fig. 1. State diagram of the automaton of the AODV Routing Protocol

The n-node network is modelled as follows

$$ \mathbf{G}_N = {}^{1}\mathbf{G}_N \parallel {}^{2}\mathbf{G}_N \parallel \cdots \parallel {}^{n}\mathbf{G}_N, $$

where the symbol "$\parallel$" denotes the synchronous product of two or more automata (see [6]). In the special case of disjoint sets, as the present case, the synchronous product becomes a shuffle (see [6] and [7]) of the participating automata. The closed and marked behavior of $\mathbf{G}_{N}$ are

$$ \mathbb{L}(\mathbf{G}_N) = \bigcap_{\lambda=1}^{n}\left({}^{\lambda}P^{-1}\left(\mathbb{L}({}^{\lambda}\mathbf{G}_N)\right)\right), $$
$$ \mathbb{L}_m(\mathbf{G}_N) = \bigcap_{\lambda=1}^{n}\left({}^{\lambda}P^{-1}\left(\mathbb{L}_m({}^{\lambda}\mathbf{G}_N)\right)\right), $$

where ${}^{\lambda}P$ is the projection of $\mathbb{E}_N$ to ${}^{\lambda}\mathbb{E}_N$ and $\mathbb{E}_N = \bigcup_{\lambda=1}^{n} {}^{\lambda}\mathbb{E}_N$.

It is important to mention that, to the best of our knowledge, this is the first attempt towards modelling of the AODV protocol in the RW framework of discrete event systems.

B. *Simulation of the AODV Routing Protocol - the four-node case*

Consider a network with four nodes, i.e., $n = 4$. The nodes are named as node 1, node 2, node 3 and node 4. Let node 1 attempts to create a route. In Table I, the simulation of this routing is presented. In the first column, the sequence of the events that take place are presented. In the next four columns, the states of the automata of the 4 nodes are presented. According to Table I, node 1 sends a RREQ and waits for its reception. After node 1 sent the request, node 2 also sends a RREQ. Next, node 3 receives and serves a request. Node 4 sends a RREQ and also node 2 receives and serves a request. Finally, node 3 sends a RREQ and node 4 receives and serves a request. It is observed that the requests of the nodes 1 and 2 and take place before any of them is answered. The same holds for the requests of the nodes 3 and 4. Furthermore, although the request of node 1 precedes the one of node 2, the latter is answered firstly. Roughly speaking, the sequence of requests - answers is not preserved.

TABLE I. SIMULATION OF THE AODV PROTOCOL

Events	Node 1 state	Node 2 state	Node 3 state	Node 4 state
	${}^{1}q_{N,1}$	${}^{2}q_{N,1}$	${}^{3}q_{N,1}$	${}^{4}q_{N,1}$
${}^{1}e_{N,1}$	${}^{1}q_{N,2}$	${}^{2}q_{N,1}$	${}^{3}q_{N,1}$	${}^{4}q_{N,1}$
${}^{2}e_{N,1}$	${}^{1}q_{N,2}$	${}^{2}q_{N,2}$	${}^{3}q_{N,1}$	${}^{4}q_{N,1}$
${}^{3}e_{N,3}$	${}^{1}q_{N,2}$	${}^{2}q_{N,2}$	${}^{3}q_{N,3}$	${}^{4}q_{N,1}$
${}^{3}e_{N,4}$	${}^{1}q_{N,2}$	${}^{2}q_{N,2}$	${}^{3}q_{N,1}$	${}^{4}q_{N,1}$
${}^{2}e_{N,2}$	${}^{1}q_{N,2}$	${}^{2}q_{N,1}$	${}^{3}q_{N,1}$	${}^{4}q_{N,1}$
${}^{4}e_{N,1}$	${}^{1}q_{N,2}$	${}^{2}q_{N,1}$	${}^{3}q_{N,1}$	${}^{4}q_{N,2}$
${}^{2}e_{N,3}$	${}^{1}q_{N,2}$	${}^{2}q_{N,3}$	${}^{3}q_{N,1}$	${}^{4}q_{N,2}$
${}^{2}e_{N,5}$	${}^{1}q_{N,2}$	${}^{2}q_{N,5}$	${}^{3}q_{N,1}$	${}^{4}q_{N,2}$
${}^{2}e_{N,6}$	${}^{1}q_{N,2}$	${}^{2}q_{N,1}$	${}^{3}q_{N,1}$	${}^{4}q_{N,2}$
${}^{1}e_{N,2}$	${}^{1}q_{N,1}$	${}^{2}q_{N,1}$	${}^{3}q_{N,1}$	${}^{4}q_{N,2}$
${}^{3}e_{N,1}$	${}^{1}q_{N,1}$	${}^{2}q_{N,1}$	${}^{3}q_{N,2}$	${}^{4}q_{N,2}$
${}^{4}e_{N,2}$	${}^{1}q_{N,1}$	${}^{2}q_{N,1}$	${}^{3}q_{N,2}$	${}^{4}q_{N,1}$
${}^{4}e_{N,3}$	${}^{1}q_{N,1}$	${}^{2}q_{N,1}$	${}^{3}q_{N,2}$	${}^{4}q_{N,3}$
${}^{4}e_{N,5}$	${}^{1}q_{N,1}$	${}^{2}q_{N,1}$	${}^{3}q_{N,2}$	${}^{4}q_{N,4}$
${}^{2}e_{N,1}$	${}^{1}q_{N,1}$	${}^{2}q_{N,2}$	${}^{3}q_{N,2}$	${}^{4}q_{N,4}$
${}^{4}e_{N,6}$	${}^{1}q_{N,1}$	${}^{2}q_{N,2}$	${}^{3}q_{N,2}$	${}^{4}q_{N,1}$
${}^{3}e_{N,2}$	${}^{1}q_{N,1}$	${}^{2}q_{N,2}$	${}^{3}q_{N,1}$	${}^{4}q_{N,1}$

III. SUPERVISORY CONTROL ARCHITECTURE OF THE AODV ROUTING PROTOCOL

A. Specifications of the Routing Protocol

The desired behavior of the AODV Routing Protocols will be described for a network consisting of n nodes with joint paths [1], with $n > 2$. It is important to mention that according to [4] the AODV protocol is used for nodes with joint paths and the AOMDV protocol, being an extension of the AODV protocol, is used to cover cases with also disjoint paths. One of the AODV protocol goals is to reduce the network bandwidth with less broadcasts ([1] and [2]). Towards strengthening the above characteristic of the AODV protocol, a quite simple and restrictive specification, regarding the hierarchy of routing requests, will be imposed. So, the desired specification of the AODV protocol to a n-node network is described as follows:

- When a node sends a routing request, no other node is allowed to send a routing request

The above rule guarantees that there will not be more than one routing request in the system at the same time, thus contributing to the decrease of routing requests for a short or medium period.

The rule is formulated as the following regular language

$$ {}^{N}\mathbb{K}=\overline{\left(\left(\mathop{+}_{\lambda=1}^{n}({}^{\lambda}e_{N,4}+{}^{\lambda}e_{N,6})\right)^{*}\left(\mathop{+}_{\lambda=1}^{n}({}^{\lambda}e_{N,1})\right)\left(\mathop{+}_{\lambda=1}^{n}({}^{\lambda}e_{N,4}+{}^{\lambda}e_{N,6})\right)\right)^{*}} . $$

A characteristic of the language ${}^{N}\mathbb{K}$ is the allowability of all nodes to send their packets (events ${}^{\lambda}e_{N,4}$ and ${}^{\lambda}e_{N,6}$) even though a packet has been received by the requesting node. Therefore, the Kleene star of the events ${}^{\lambda}e_{N,4}$ and ${}^{\lambda}e_{N,6}$ occurs at the beginning of the language.

B. Automaton realizing the regular language

Let $\mathbf{S}_N$ be an automaton described by the 6-tuple

$$ \mathbf{S}_N=(\mathbb{Q}_{S,N},\mathbb{E}_{S,N},f_{S,N},\mathbb{H}_{S,N},x_{S,N,0},\mathbb{Q}_{S,N}) . $$

The set of its states is

$$ \mathbb{Q}_{S,N}=\{q_{S,N,1},q_{S,N,2}\} . $$

The alphabet of $\mathbf{S}_N$ is

$$ \mathbb{E}_{S,N}=\bigcup_{\lambda=1}^{n}\{{}^{\lambda}e_{N,1},{}^{\lambda}e_{N,4},{}^{\lambda}e_{N,6}\} . $$

The active event sets for all states of $\mathbf{S}_N$ are

$$ \mathbb{H}_{S,N}(q_{S,N,1})=\bigcup_{\lambda=1}^{n}\{{}^{\lambda}e_{N,1},{}^{\lambda}e_{N,4},{}^{\lambda}e_{N,6}\} \text{ and} $$

$$ \mathbb{H}_{S,N}(q_{S,N,2})=\bigcup_{\lambda=1}^{n}\{{}^{\lambda}e_{N,4},{}^{\lambda}e_{N,6}\} . $$

The values of the transition function of the states and the events of the automaton $\mathbf{S}_N$ are

$$ f_{S,N}(q_{S,N,1},{}^{i}e_{N,1})=q_{S,N,2},\ f_{S,N}(q_{S,N,1},{}^{i}e_{N,4})=q_{S,N,1}, $$
$$ f_{S,N}(q_{S,N,1},{}^{i}e_{N,6})=q_{S,N,1};\ \forall i\in\{1,...,n\}, $$
$$ f_{S,N}(q_{S,N,2},{}^{i}e_{N,4})=q_{S,N,1},\ f_{S,N}(q_{S,N,2},{}^{i}e_{N,6})=q_{S,N,1}; $$
$$ \forall i\in\{1,...,n\} . $$

The initial state of $\mathbf{S}_N$ is $x_{S,N,0}=q_{S,N,1}$. The set of the marked states of $\mathbf{S}_N$ is $\mathbb{Q}_{S,N,m}=\mathbb{Q}_{S,N}$.

The marked behavior of $\mathbf{S}_N$ is

$$ \mathbb{L}_m(\mathbf{S}_N)= \overline{\left(\left(\mathop{+}_{\lambda=1}^{n}({}^{\lambda}e_{N,4}+{}^{\lambda}e_{N,6})\right)^{*}\left(\mathop{+}_{\lambda=1}^{n}({}^{\lambda}e_{N,1})\right)\left(\mathop{+}_{\lambda=1}^{n}({}^{\lambda}e_{N,4}+{}^{\lambda}e_{N,6})\right)\right)^{*}} . $$

Obviously, it holds that

$$ \mathbb{L}_m(\mathbf{S}_N)=\mathbb{L}(\mathbf{S}_N)={}^{N}\mathbb{K} . $$

The state diagram of $\mathbf{S}_N$ is presented in Figure 2.

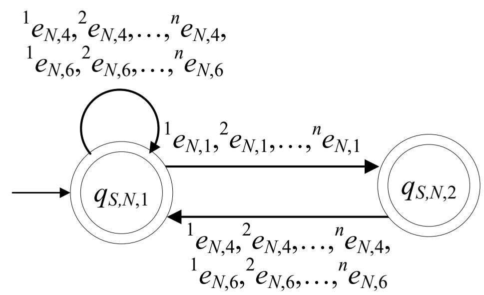

Fig. 2. State diagram of the automaton of $\mathbf{S}_N$

C. Desired behavior

The desired behavior of the controlled system is formulated by the following desired language

$$ {}^{N}\mathbb{K}_D={}^{N}P^{-1}({}^{N}\mathbb{K})\cap\mathbb{L}_m(\mathbf{G}_N), \tag{1} $$

where ${}^{N}P$ is the projection of $\mathbb{E}_N$ to $\mathbb{E}_{S,N}$. The desired behavior of the total controlled system is the sublanguage of ${}^{N}\mathbb{K}$ that is also a sublanguage of the marked behavior of the automaton modelling the n-node network. An advantage of the proposed desired language is that it covers the specification of every node facilitating the language implementation.

D. The controlled automaton and the distributed architecture of the supervisory control

To the best of our knowledge, the present paper is the first effort towards designing a supervisor in the RW framework to modify characteristics of an AODV protocol. A similar direction is followed in [5] for other communication protocols. According to [8] and [13], in order to satisfy the desired behavior, it suffices to apply to the system $\mathbf{G}_N$, through a synchronous product relation, the automaton realizing the language ${}^{N}\mathbb{K}$, namely the supervisor automaton $\mathbf{S}_N$. Hence, the controlled automaton is of the form

$$ \mathbf{G}_{N,c}=\mathbf{S}_N \parallel {}^{1}\mathbf{G}_N \parallel {}^{2}\mathbf{G}_N \parallel\cdots\parallel {}^{n}\mathbf{G}_N , $$

or equivalently of the form

$$ \mathbf{G}_{N,c}=\mathbf{S}_N\parallel\mathbf{G}_N . \tag{2} $$

The closed behavior of $\mathbf{G}_{N,c}$ is

$$ \mathbb{L}(\mathbf{G}_{N,c})=\mathbb{L}(\mathbf{G}_N)\cap{}^{N}P^{-1}\left(\overline{{}^{N}\mathbb{K}}\right) \tag{3} $$

and the marked behavior of $\mathbf{G}_{N,c}$ is

$$ \mathbb{L}_m(\mathbf{G}_{N,c})=\mathbb{L}_m(\mathbf{G}_N)\cap{}^{N}P_1^{-1}({}^{N}\mathbb{K}) . \tag{4} $$

Using (1) and (2) as well as (3) and (4), the following relations are derived

$$\mathbb{L}(\mathbf{G}_{N,c}) = \mathbb{L}(\mathbf{G}_N) \cap {}^{N}P^{-1}\left({}^{N}\mathbb{K}\right) \qquad (5)$$

$$\mathbb{L}_m(\mathbf{G}_{N,c}) = {}^{N}\mathbb{K}_D \,. \qquad (6)$$

From (6) it is observed that the marked behavior of the controlled automaton of the parametric network is equal to the desired behavior.

As already mentioned, the proposed supervisory control architecture is distributed, and the nodes are not controlled or commanded through a central control unit. The supervisor automaton is proposed to be implemented to the computer hardware of every node. This way, appropriate events of the protocol model of a node can be disabled. Finally, it is mentioned that the synchronization of the supervisors is inherently achieved since the events of the alphabet $\mathbb{E}_{S,N}$ are all "public" events of the type of request events that can be received (heard) by all nodes.

E. The nonblocking controlled scheme

To guarantee that the supervisor, realized in Subsection III.B, is a proper supervisor (see [6]), the following properties must be satisfied

1 The language ${}^{N}\mathbb{K}_D$ must be controllable regarding the automaton $\mathbf{G}_N$.

2 The language ${}^{N}\mathbb{K}_D$ must be $\mathbb{L}_m(\mathbf{G}_N)$ -closed, i.e., ${}^{N}\mathbb{K}_D = \overline{{}^{N}\mathbb{K}_D} \cap \mathbb{L}_m(\mathbf{G}_N)$.

According to [6] (see page 163), if the above two properties and relation (6) are satisfied, then the controlled automaton $\mathbf{G}_{N,c}$ is a nonblocking automaton. It is important to mention that, although the automaton $\mathbf{G}_N$ (routing protocol model) is a nonblocking automaton, it is important for the efficiency of the proposed supervisor control scheme to verify that the controlled automaton $\mathbf{G}_{N,c}$ is also nonblocking

Regarding the controllability (see [6]) of the desired language with respect to $\mathbf{G}_N$ the proof is straightforward as there are no uncontrollable events in the system.

Proposition 1: The language ${}^{N}\mathbb{K}_D$ is $\mathbb{L}_m(\mathbf{G}_N)$ -closed.

Proof: Using (1) it holds that

$$\overline{{}^{N}\mathbb{K}_D} \cap \mathbb{L}_m(\mathbf{G}_N) = \overline{{}^{N}P^{-1}({}^{N}\mathbb{K}) \cap \mathbb{L}_m(\mathbf{G}_N)} \cap \mathbb{L}_m(\mathbf{G}_N)$$

Decompose the prefix-closed language ${}^{N}P^{-1}({}^{N}\mathbb{K})$ into two prefix-closed sublanguages, i.e., ${}^{N}P^{-1}({}^{N}\mathbb{K}) = {}^{1}\hat{\mathbb{K}} \cup {}^{2}\hat{\mathbb{K}}$, where ${}^{1}\hat{\mathbb{K}} \subseteq \mathbb{L}(\mathbf{G}_N)$ and ${}^{2}\hat{\mathbb{K}} \cap \mathbb{L}(\mathbf{G}_N) = \varnothing$. Such a decomposition always exists as $\mathbb{L}(\mathbf{G}_N)$ is a prefix-closed language. Hence, it holds that ${}^{N}\mathbb{K}_D = {}^{1}\hat{\mathbb{K}} \cap \mathbb{L}_m(\mathbf{G}_N)$. Since ${}^{1}\hat{\mathbb{K}}$ is a prefix-closed language, it is observed that $\overline{{}^{1}\hat{\mathbb{K}} \cap \mathbb{L}_m(\mathbf{G}_N)} \subseteq {}^{1}\hat{\mathbb{K}}$. Thus,

$$\overline{{}^{N}\mathbb{K}_D} \cap \mathbb{L}_m(\mathbf{G}_N) =$$
$$\overline{{}^{1}\hat{\mathbb{K}} \cap \mathbb{L}_m(\mathbf{G}_N)} \cap \mathbb{L}_m(\mathbf{G}_N) \subseteq$$
$${}^{1}\hat{\mathbb{K}} \cap \mathbb{L}_m(\mathbf{G}_N) = {}^{N}\mathbb{K}_D$$

Since ${}^{N}\mathbb{K}_D \subseteq \mathbb{L}_m(\mathbf{G}_N)$, it holds that ${}^{N}\mathbb{K}_D \subseteq \overline{{}^{N}\mathbb{K}_D} \cap \mathbb{L}_m(\mathbf{G}_N)$. Thus, the proof has been completed. ■

The proposed supervisory control architecture and the supervisor realization facilitate the implementation as the supervisors can be implemented by the same function block in the computer platforms of all nodes. More details regarding the implementation of supervisors through function blocks can be found in [8], [14] and [15] and the references therein.

F. Simulation of the controlled AODV Routing Protocol for the four-node case

To validate the effectiveness of the proposed control scheme, the simulation of the four-node network presented in Subsection II.B under the influence of the supervisor and using the event sequence of Subsection II.B, will be presented. In Table II the simulation results of controlled routing protocol are presented. Comparing Table I and Table II, it is observed that after the RREQ request of node 1 the RREQ request of node 2 cannot take place (event ${}^{2}e_{N,1}$) as no answer has yet been transmitted to node 1. Similarly, the RREQ request of node 2 (event ${}^{3}e_{N,2}$) after the RREQ request of node 3 cannot take place. The above simulation makes it clear that after a RREQ request of a node, a reply (path or simple answer) must be sent to the respective node or else the other nodes cannot send a RREQ request to the network. This way the sequence of requests - answers is preserved, in the sense that each request takes place after the answer of the previous request. This characteristic appears to be useful in designing an additional possibly lumped supervisory control scheme of the network.

TABLE II. SIMULATION OF THE CONTROLLED AODV PROTOCOL

Events	Node 1 state	Node 2 state	Node 3 state	Node 4 state
	${}^{1}q_{N,1}$	${}^{2}q_{N,1}$	${}^{3}q_{N,1}$	${}^{4}q_{N,1}$
${}^{1}e_{N,1}$	${}^{1}q_{N,2}$	${}^{2}q_{N,1}$	${}^{3}q_{N,1}$	${}^{4}q_{N,1}$
${}^{2}e_{N,1}$	${}^{1}q_{N,2}$	${}^{2}q_{N,1}$	${}^{3}q_{N,1}$	${}^{4}q_{N,1}$
${}^{3}e_{N,3}$	${}^{1}q_{N,2}$	${}^{2}q_{N,1}$	${}^{3}q_{N,3}$	${}^{4}q_{N,1}$
${}^{3}e_{N,4}$	${}^{1}q_{N,2}$	${}^{2}q_{N,1}$	${}^{3}q_{N,1}$	${}^{4}q_{N,1}$
${}^{2}e_{N,2}$	${}^{1}q_{N,2}$	${}^{2}q_{N,1}$	${}^{3}q_{N,1}$	${}^{4}q_{N,1}$
${}^{4}e_{N,1}$	${}^{1}q_{N,2}$	${}^{2}q_{N,1}$	${}^{3}q_{N,1}$	${}^{4}q_{N,2}$
${}^{2}e_{N,3}$	${}^{1}q_{N,2}$	${}^{2}q_{N,3}$	${}^{3}q_{N,1}$	${}^{4}q_{N,2}$
${}^{2}e_{N,5}$	${}^{1}q_{N,2}$	${}^{2}q_{N,5}$	${}^{3}q_{N,1}$	${}^{4}q_{N,2}$
${}^{2}e_{N,6}$	${}^{1}q_{N,2}$	${}^{2}q_{N,1}$	${}^{3}q_{N,1}$	${}^{4}q_{N,2}$
${}^{1}e_{N,2}$	${}^{1}q_{N,1}$	${}^{2}q_{N,1}$	${}^{3}q_{N,1}$	${}^{4}q_{N,2}$
${}^{3}e_{N,1}$	${}^{1}q_{N,1}$	${}^{2}q_{N,1}$	${}^{3}q_{N,2}$	${}^{4}q_{N,2}$
${}^{4}e_{N,2}$	${}^{1}q_{N,1}$	${}^{2}q_{N,1}$	${}^{3}q_{N,2}$	${}^{4}q_{N,1}$
${}^{4}e_{N,3}$	${}^{1}q_{N,1}$	${}^{2}q_{N,1}$	${}^{3}q_{N,2}$	${}^{4}q_{N,3}$
${}^{4}e_{N,5}$	${}^{1}q_{N,1}$	${}^{2}q_{N,1}$	${}^{3}q_{N,2}$	${}^{4}q_{N,4}$
${}^{2}e_{N,1}$	${}^{1}q_{N,1}$	${}^{2}q_{N,1}$	${}^{3}q_{N,2}$	${}^{4}q_{N,4}$
${}^{4}e_{N,6}$	${}^{1}q_{N,1}$	${}^{2}q_{N,1}$	${}^{3}q_{N,2}$	${}^{4}q_{N,1}$
${}^{3}e_{N,2}$	${}^{1}q_{N,1}$	${}^{2}q_{N,1}$	${}^{3}q_{N,1}$	${}^{4}q_{N,1}$

CONCLUSIONS

The model of the AODV routing protocol for one node has been presented and the generic model of the protocol for an n node network has been developed. The desired hierarchy of the parametric network has been satisfied by expressing the hierarchy in the form of a desired regular language and realizing the regular language by appropriate supervisor. The nonblocking property of the resulting controlled automaton has been proven.

The use of Supervisory Control theory for modeling and control of other routing and communicating protocols is currently under investigation. The issue of developing more sophisticated specification for the hierarchy of the requests in AODV protocol is also under investigation.

ACKNOWLEDGMENT

Research supported by the MSc Program in "Advanced Control Systems and Robotics", National and Kapodistrian University of Athens, Euripus Campus, Greece. The authors would like to thank the anonymous reviewers for their fruitful comments contributing at large to the distributed type of the present control configuration.

REFERENCES

[1] A. B. M. Moniruzzaman, Md. Waliullah, and Md. Sadekur Rahman, "Analysis of Topology Based Routing Protocols for Vehicular Ad-Hoc Network (VANET)," *International Journal of Future Generation Communication and Networking*, Vol. 8, No. 3, pp. 97-110, 2015.

[2] O. Salman, R. Morcel, O. Al Zoubi, I. Elhajj, A. Kayssi and, A. Chehab, "Analysis of Topology Based Routing Protocols for VANETs in Different Environments," *2016 IEEE International Multidisciplinary Conference on Engineering Technology (IMCET), pp. 27-31, 2016.*

[3] A. Hsu, F. Eskafi, S. Sachs and P. Varaiya, "Protocol Design for an Automated Highway System," *Discrete Event Dynamic Systems: Theory and Applications*, vol. 2, pp. 183-206, 1993.

[4] K. N. Qureshi and A. H. Abdullah, "Study of Efficient Topology Based Routing Protocols for Vehicular Ad-Hoc Network Technology," *World Applied Sciences Journal 23*, vol. 5, pp 656-663, 2013.

[5] F. N. Koumboulis, D. G. Fragkoulis and L. E. Gikas, "Supervisory control for hierarchical switching of communication protocols," *2017 South Eastern European Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM)*, pp. 1-6., 2017.

[6] W. M. Wonham and C. Kai, Supervisory control of discrete-event systems, *Springer International Publishing*, 2019.

[7] C. G. Casandras and S. Lafortune, Introduction to Discrete Event Systems, *Kluwer Academic Publishers*, 1999.

[8] F. N. Koumboulis, D. G. Fragkoulis, I. Kalkanas, and G. F. Fragulis, "Supervisor Design for a Pressurized Reactor Unit in the Presence of Sensor and Actuator Faults," *Electronics*, vol. 11, no. 16, p. 2534, Aug. 2022.

[9] F. N. Koumboulis, D. G. Fragkoulis, and S. Arapakis, "Supervisor design for an assembly line in the presence of faults," *IEEE International Conference on Emerging Technologies and Factory Automation*, 6th-9th September, Stuttgart, Germany, 2022.

[10] F. N Koumboulis, D. G. Fragkoulis, and A. N. Menexis, "Supervisory Control for Flexibility of Production Manufacturing Processes," *IEEE 21st Int. Conf. on Intelligent Engineering Systems* 2017 (INES 2017), Larnaca, October, 20-23, Cyprus, 2017.

[11] F. N. Koumboulis, D. G. Fragkoulis, and K. A. Ioannou, :Control of Router Nodes in Production Manufacturing Processes," *2018 7th Int. Conf. on Systems and Control (ICSC)*, pp. 372-377, 2018.

[12] F. N. Koumboulis, D. G. Fragkoulis, and G. K. Diveris, "Function Supervisors for Storage Systems," *Int. Conf. on Modern Circuits and Systems Technologies (MOCAST)*, May 7-9, Thessaloniki, Greece, 2018.

[13] M. H. de Queiroz and J. E. R. Cury, "Modular supervisory control of large scale discrete-event systems," *Discrete Event Systems: Analysis and Control*, Kluwer Academic Publishers, pp. 103-110. (Proc. WODES 2000, Ghent, Belgium), 2000.

[14] M. H Queiroz and J.E.R. Cury, "Synthesis and implementation of local modular supervisory control for a manufacturing cell," *Proc. of the IFAC 6th International Workshop in Discrete Event Systems,* Zaragoza, Spain, 4 October 2002.

[15] G. Cengic, O. Ljungkrantz and K.Akesson, "Formal Modeling of Function Block Applications Running in IEC 61499 Execution Runtime," Proc. of the 2006 IEEE Conference on Emerging Technologies and Factory Automation (ETFA), Prague , Czech, 20–22 September 2006.