Intelligent Solution for Congestion Control in Wireless Ad hoc Networks

L. Khoukhi, S. Cherkaoui
Department of Electrical and Computer Engineering, University of Sherbrooke
J1K 2R1, QC, Canada
{Lyes.Khoukhi, Soumaya.Cherkaoui}@USherbrooke.ca

***Abstract*—In this paper, an intelligent solution is proposed for the congestion control in wireless ad hoc network. The presented QoS architecture exploits the fuzzy logic for improving the congestion control in the aim to support voice and video multimedia applications, and non-real-time traffic services. We present two techniques: the first one uses a fuzzy logic system to perform the threshold buffer management. The second technique is based on Fuzzy Petri nets (FPWICC) for modeling and analyzing the QoS decision making for congestion control in wireless ad hoc network. The performance evaluation was studied under different channel, mobility, and traffic conditions. The simulation results confirm that the intelligent fuzzy logic tool is a promising solution to support QoS for multimedia applications in wireless ad hoc networks.**

***Index Terms*—QoS, multimedia application, congestion control, fuzzy logic, threshold management, fuzzy Petri nets.**

I. Introduction

THE rapid development of the wireless technology has been accompanied by an evolution of new multimedia applications. Sample multimedia applications include videoconferencing, distance learning, distributed games, video on demand, etc. These applications consist of voice and video traffic, and they need delay and loss guarantees. Others applications, such as World Wide Web and FTP, are delay-insensitive. However, maintaining the end-to-end voice and video quality is too difficult to be accommodated by the wireless ad hoc networks.

Wireless ad hoc networks are self-creating, self-organizing, and self-administering [1]. The dynamic nature of these networks presents significant technical challenges to ensure better quality delivery to the multimedia applications. These applications generate traffic at varying rates and usually require that the network be able to carry traffic at the rate at which they generate it. Besides, they are more or less tolerant in terms of traffic delays and variation in traffic delay. Therefore, it is vital to support the Quality of Service (QoS) for multimedia applications.

In response to the challenges posed by the wireless ad hoc technology, various approaches and protocols have been proposed [4]-[15]. Note that the classical QoS architectures (e.g., IntServ [2] and DiffServ [3]) proposed for the wired networks are not readily applicable to this new technology. Recently, many researches have focused on addressing the QoS issue in the ad hoc networks: SWAN [4], INSINGIA [5], and FQMM [6]. The later model is a hybrid approach combining the advantages of per-class granularity of DiffServ with the per-flow granularity of IntServ. It tries to preserve the per-flow granularity for a small portion of traffic in MANETs, given that a large amount of the traffic belongs to per aggregate of flows, that is, per-class granularity. FQMM offers a good solution for small- and medium-size ad hoc network, but it is not suitable for large networks. INSIGNIA is one of the noteworthy QoS frameworks with per-flow granularity and reasonable treatment for mobility. The main goal of INSIGNIA is to provide adaptive QoS guarantees for real-time traffic. It employs an in-band signaling system that supports fast reservation, restoration, and adaptation algorithms. Three levels of services are implemented: best-effort, minimum, and maximum. The bandwidth is the only QoS parameter used in INSIGNIA. SWAN proposes a service differentiation in stateless wireless ad hoc networks by using distributed control algorithms. It relies on feedback from the MAC layer as a measure of congestion in the network by using a mechanism of rate control and source-based admission control. It promotes a rate control system that can be used at each node to treat traffic either as real-time or best-effort traffic. However, one of the drawbacks of SWAN is how to calculate the threshold rate limiting any excessive delay that might be experienced [7]. SWAN uses merely two levels of services: real-time and best-effort traffic, but it remains the best example of stateless distributed QoS framework developed for wireless ad hoc networks.

We have proposed in [9] GQOS model, which is an intelligent QoS model with service differentiation based on neural networks for mobile ad hoc networks. GQOS is composed of a kernel plan which assures basic functions of routing and QoS support control, and an intelligent learning plan which assures the training of GQOS kernel operations by using multilayered feedforward neural network (MFNN). The advantages of using neural networks algorithm is the fast learning of different operations performed by the kernel and the reduction of time processing in the network. However, the results of simulation show that GQOS is not suitable for high dynamic networks. To overcome this drawback, we have explored in [10] the use of a fuzzy logic semi-stateless QoS approach for service differentiation in wireless ad hoc networks, called FuzzyMARS. This architecture support both real-time UDP traffic and best-effort UDP and TCP traffic. The resulted simulations have shown the benefits of using fuzzy logic semi-stateless model, the average delay obtained is quite stable and low under different channel conditions, traffic scalability, and mobility scenarios. Nevertheless, in FuzzyMARS we did not consider buffer management.

In this paper, we propose a new QoS approach for wireless ad hoc networks, named FuzzyCCG. This approach explores the fuzzy logic for improving the control of congestion for multimedia applications. FuzzyCCG exploration is useful first, because of the dynamic nature of buffer occupancy and congestion at a node, second, because of the uncertain nature of information in wireless ad hoc networks due to the network mobility. FuzzyCCG proposes to use fuzzy logic approach for threshold selection in order to deal with the dynamic buffer occupancy and the uncertain and imprecision nature of wireless ad hoc network information. Using fuzzy logic, FuzzyCCG investigates the fuzzy thresholds ability to adapt to the dynamic conditions over the classical inflexible thresholds. The notion of threshold is practical for discarding data packets and adapting the traffic service depending on the occupancy of buffers. Therefore, the selection of a particular threshold may be decisive to the control of congestion, and therefore to the network performances.

This paper proposes also a new fuzzy Petri nets technique, named FPWICC. The objective is to model and analyze the QoS decision making for congestion control in wireless ad hoc networks. The fuzzy Petri nets tool is used for its efficiency and flexibility over other modeling tools (such as Petri nets) in the aim of better modeling and representation the process of buffer management.

The performances of FuzzyCCG are studied under different network conditions in terms of network conditions. The simulations results shown in Section IV confirm that the proposed approach offers promising results to support multimedia applications.

The rest of the paper is structured as follows: Section II describes the proposed architecture. Section III illustrates the proposed fuzzy Petri model for congestion control. The simulation results under different network conditions are shown in Section IV. Finally, Section V concludes the paper.

II. FUZZYCCG ARCHITECTURE

A. Overview of Fuzzy logic

L. Zadeh has introduced in the 1960s the Fuzzy logic theory [16]-[17] as a tool for modeling the uncertain of natural language, which has been commonly employed for supporting intelligent systems. This technology has proven efficiency in a various applications such as decision support and intelligent control, especially where a system is difficult to be characterized. A fuzzy logic system considers basically three steps: fuzzification, rules evaluation, and deffuzification. The first step is responsible for mapping discrete (called also crisp) input data into proper values in the fuzzy logic space. For that end, membership functions (fuzzy sets) are used to provide smooth transitions from false to true (0 to 1). The second step performs reasoning on the input data by following predefined fuzzy rules. Once the input data are processed by fuzzy reasoning, the deffuzification takes the task of converting back these input data into crisp values.

B. Fuzzy logic approach for threshold management

The choice of applying fuzzy logic is justified by the fact that fuzzy logic is well adapted to systems characterized by imprecision states such as the case of ad hoc networks. Also given the results found with FuzzyMARS [10], fuzzy logic promises to offer an efficient tool for buffer management by using adequate thresholds that deal with the imprecise information in a wireless ad hoc network. Also, fuzzy logic has been successfully applied to the queue management in the cell-switching networks [18]. Nevertheless, to the best of our knowledge, this is the first work that uses fuzzy logic for buffer management in MANETs. We aim to apply a fuzzy technique based on fuzzy sets theory. The later extends the classical logic set {0, 1} to use linguistic variables (e.g. full buffer, merely full buffer, empty buffer).

Using fuzzy logic, we investigate the fuzzy thresholds ability to adapt to the dynamic conditions over the classical inflexible thresholds. The classical thresholds are characterized by their limitation and restriction, because the selection of threshold is based on a single value. Thus, the utilization of a buffer may be either “poor” or “surcharged”. When the selected value is small (e.g. 30% of capacity), then the admission of new packets is possible only when the buffer occupancy is low. This means a poor utilization of the buffer; since most of incoming packets are rejected even if the buffer is almost unfilled. On the other side, when the selected value is big (e.g. 90% of capacity), problems may happen when the bursty traffic is used. The transmission of packets generated by a bursty traffic is very changing. It can vary from small to “near-peack” rate in a short period of time.

Manually predefining a value for threshold in ad hoc network is not suitable because most of events occurring in an ad hoc network are dynamic and random. In addition, it is important to note that the rate of packets arriving on a particular node is not static. The threshold value divides the buffer into an “admitted” part and a “no-admitted” part. Let consider that the threshold of the buffer shown in Fig. 1.a is equal to 60%. In this scheme, the occupancy level may range from 0 to 60%. When the buffer occupancy is superior to 60%, no incoming packets is accepted in the buffer. Therefore, the change in decision making from “admit state” to “no-admit state” is performed from 60-61%. This means that a small variation in the buffer occupancy may influence the decision making of incoming packets.

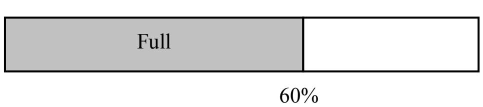
The aim of introducing fuzzy logic is to develop a more realistic representation of buffer occupancy that helps to offer an efficient decision making. The proposed architecture attempts to extend the two-discrete states “admit” and “no-admit” of the buffer occupancy by using fuzzy logic. Hence, the definition of “buffer occupancy” will consider the two fuzzy cases of “getting full” and “not getting full”, rather than “admit” and “no-admit” in the existing approaches. This fuzzy representation replaces the two-discrete sets by a continuous set membership, and performs small gradual transitions between different states of buffer occupancy.

The fuzzy membership function aims to determine the fuzzy threshold based on the fullness of the buffer. For that purpose, several membership functions may be used: “triangular”, trapezoidal”, or “sigmoid” function. These functions can give a representation about the buffer fullness level. In FuzzyCCG, we used the sigmoid membership function. This choice is based on the fact that this function would reflect well the dynamic occupancy of the buffers that we want to model.

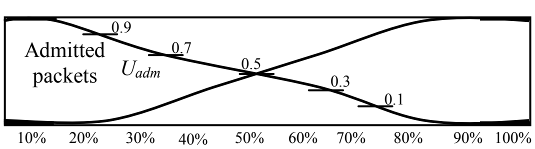
It is observed in Fig. 1.b that the admit membership function is inversely proportional to the occupancy fullness level of buffer. Thus, when the occupancy fullness is small, the value of the admit membership function is big. At higher fullness occupancy levels, the admit membership function value becomes small. When the value of the “no-admit” membership function is getting big, then only a small quantity of packets will be permitted to enter the buffer. In Fig. 1.b, the value of the membership function is represented by the symbol u_{adm}. The fuzzy rules associated are as follows:

<< When the value of the admit membership function is big, then increase the accepted incoming packets into buffer >>.
<< When the value of the admit membership function is small, then reduce the accepted incoming packets into buffer >>.

These fuzzy rules are illustrated by Fig. 1.b. The rejection of packets is controlled based on the degree of fullness of the buffer. For instance, when the buffer is occupied at 40%, this means that the value of u_{adm} is about 0.7 (i.e. the amount of packets admitted is about 70%). Then, about “30%” of incoming packets will be not admitted. Note that the fuzzy threshold approach covers the continuous set of values representing possible buffer occupancy (i.e. from 0 to u_{adm}). This is opposite to the classical threshold approaches that hold only one predefined single value. Therefore, fuzzy logic adds more flexibility to the threshold selection.



1.a) Classical buffer scheme



1.b) Fuzzy Buffer occupancy scheme

Fig. 1. Classical and fuzzy buffer schemes

III. Congestion Control Model Based on Fuzzy Petri Nets

In the previous sections, we presented the fuzzy logic approach for the congestion control. We interest in what follows, to model and analyse the QoS making decision for congestion control using a Fuzzy Petri tool. In [27], we have presented a similar model for the traffic regulation.

The proposed model, called FPWICC, studies the fuzzy congestion control rules in order to deal with the imprecise information caused by the dynamic topology of ad hoc networks. The representation of different fuzzy processes for decision making can be performed by formulating the production rules of these processes. Each fuzzy production rule is a set of antecedent input conditions and consequent output propositions. We proceed to construct the previous aspects (the input and output parameters) of the production rules in order to better represent and understand the process of congestion control in wireless ad hoc networks. The congestion control process used to avoid the congestion depends on the buffer occupancy and the dynamic topology of the network. These constraints represent the input parameters of FPWICC. The buffer occupancy is given the FuzzyCCG admit membership function. The later can give information about the status of a network in terms of congestion. A small value of this parameter signifies that congestion may be appeared in the network. Therefore, the process of traffic regulation should be started. The amount of the accepted incoming packets into buffer represents the output parameter of FPWICC. The choice of using fuzzy Petri nets tool is due to its efficiency and flexibility over other modeling tools for better representing the congestion control process.

A. *Fuzzy Petri Nets*

It is observed that classical Petri Nets [19] do not have sufficient capacity to model the uncertainty in systems [20]. This limitation of Petri nets has encouraged researchers to extend the exiting models by using the fuzzy reasoning theory [21] [22]. The combination of Petri nets models and fuzzy theory has given rise to a new modeling tool called Fuzzy Petri Nets (FPN). FPN formalism has been widely applied in several applications such as, robotics systems [23], and real-time control system [20], fuzzy reasoning systems [25], etc…

A brief description about the FPN modeling tool is presented in the following [22] [24].

Let consider FPN = (PN, CND, MF, FSR, FM).

a. The tuple PN = (P, T, A, FW, FH) is called Petri nets if: (P, T, A) is a finite net, where [19]:
   $P = \{P_1, P_2, \ldots, P_n\}$ is a finite non-empty set of places,
   $T = \{T_1, T_2, \ldots, T_n\}$ is a finite non-empty set of transitions,
   $A \subseteq (P \times T) \cup (T \times P)$ is a finite set of arcs between the places and transitions or vice versa.
   FW: $A \rightarrow N^+$ represents a weighting function that associates with each arc of PN a non-negative integer of N^+.

$FH \subset (P \times T)$: represents an inhibition function that associates a place $P_i \in P$ contained in FH (T_j) to a transition T_j itself.

b. CND = {cd_1, cd_2, …., cd_n} represents a set of conditions that will be mapped into the set P; each $cd_i \in CND$ is considered as one input to the place $P_i \in P$. A condition cd_i takes the form of "X is Z", which means a combination between the fuzzy set Z and the attribute X of the condition. For instance, in the condition "the admit membership value is small", the attribute "X = admit membership value" is associated to the fuzzy set "Z = small", but other fuzzy sets can also be considered (e.g. "Z = medium", "Z = large", etc.).

c. Consider MF: $u_Z(x) \rightarrow T$, a membership function which maps the elements of X (as defined in b.) into the values of the range [0,1]. These values represent the membership degree in the fuzzy set Z. The element x belonging to X represents the input parameter of the condition "X is Z", and $u_z(x)$ measures the degree of truth of this condition. Note that the composition of membership function degrees of the required conditions is performed by fuzzy operators such as MIN/MAX.

d. Let consider the following rule R_i:

R_i: if x_1 is z_1 and /or x_2 is z_2 then A is B

The firing strength function of rule R_i (FSR_i) represents the strength of belief in R_i. The conclusion of R_i (modeled by CSR_i) can take one of the following forms:

$$CSR_i = MIN(u_{z1}(x_1), u_{z2}(x_2)) = u_{z1}(x_1) \wedge u_{z2}(x_2)$$

$$CSR_i = MAX(u_{z1}(x_1), u_{z2}(x_2)) = u_{z1}(x_1) \vee u_{z2}(x_2)$$

e. SWR is the selected wining rule R_L among the n-rules R_1, R_2, …., R_n. SWR is the rule which has the highest degree of truth. Let FSR_L be the corresponding firing strength of R_L, then the selected rule SWR is given as follows:

$$SWR = MAX(FSR_1, FSR_2, ..., FSR_n)$$

f. The marking task in FPN illustrates the satisfaction of events occurred during the performance of fuzzy rules. This marking function called "fuzzy marking" (FM) distributes the tokens over the places of the nets.

The sequence $\delta = \langle T_1, T_2, ..., T_n \rangle$ is said to be reachable from a fuzzy marking FM_1, if $T_i \in T$ is a firable from $FM_{i-1} \in FM$ and leads to $FM_{i+1} \in FM$, for all transitions $T_i \in \delta$. The firing of transition $T_i \in T$ (Fig. 2) is performed in two steps: a) T_i removes tokens and then, b) T_i places tokens.

B. *Fuzzy Congestion Control Rules Using*

The following form of modeling is used by most of fuzzy systems [26]:

Rule R: if Ip_1 is A AND Ip_2 is B then Op is C

Where:

- Ip_1 and Ip_2 are the input parameters,
- Op is an output parameter,
- A, B, and C are fuzzy sets,
- AND represent fuzzy operator,
- The fuzzy conditions of rule R are "Ip_1 is A", and "Ip_2 is B".

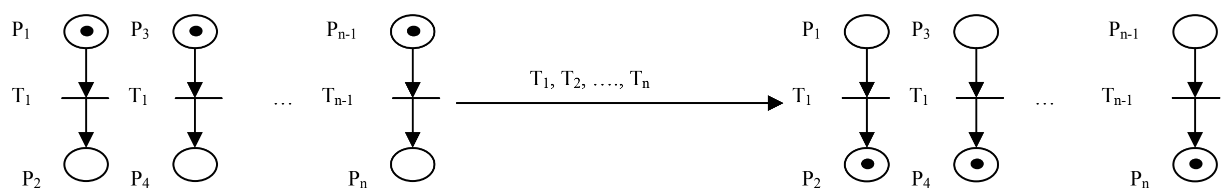


Fig. 2. The transitions firing in FPN

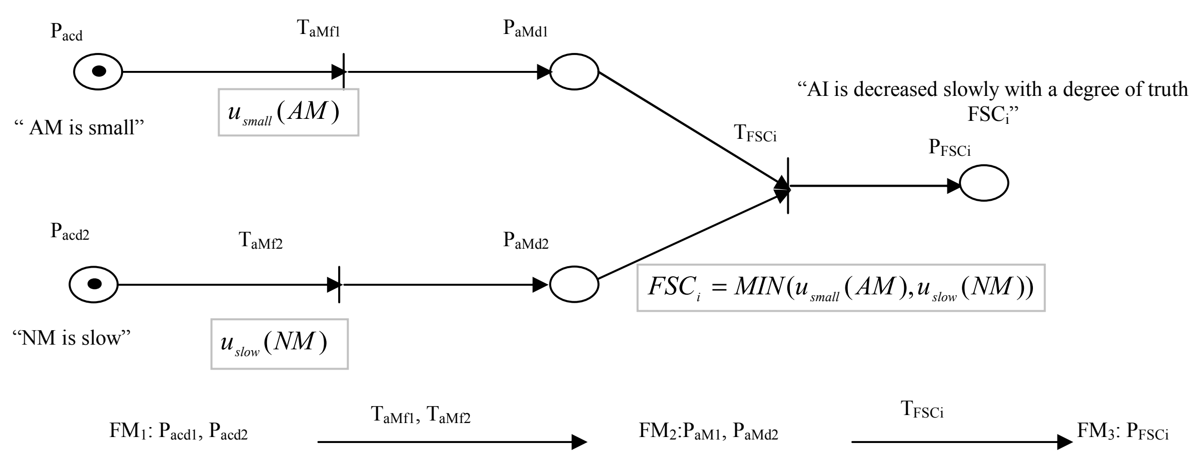
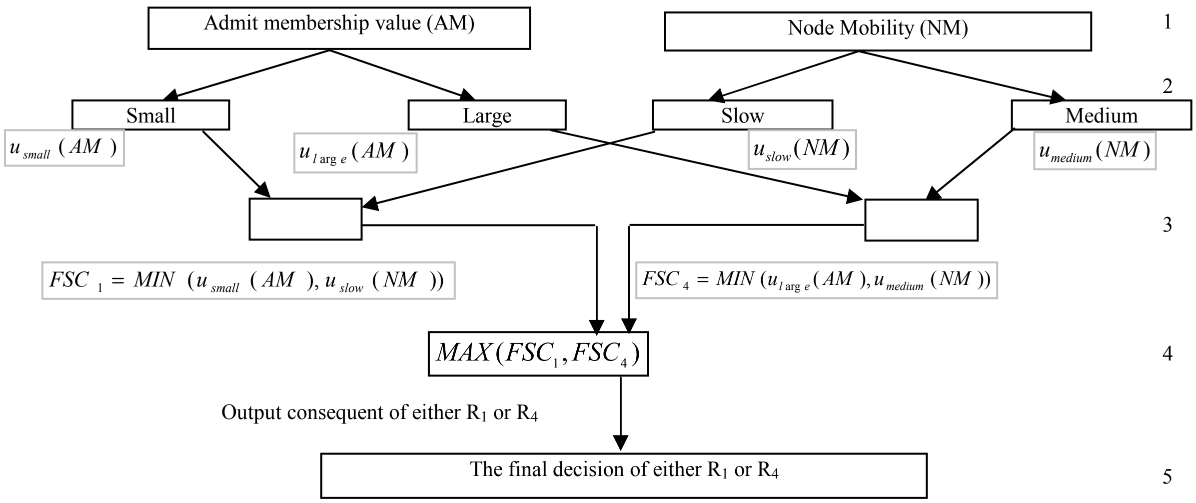


Fig. 3. The modeling of fuzzy rules structure and its dynamic behaviour



Decision making algorithm:

- Phase 1: enter the input parameters of the rules R_1, R_2, R_3, and R_4.
- Phase 2: calculate the degree of truth of the antecedent conditions.
- Phase 3: apply the operation of minimum composition (MIN) with the fuzzy operator AND/OR in order to generate the firing strength value for each rule R_1, R_2, R_3, and R_4.
- Phase 4: apply the operation of maximum composition to select the wining rule among the rules R_1, R_2, R_3, and R_4.
- Phase 5: generate the output consequent of the selected wining rule.

Fig. 4. The fuzzy decision making mechanism of FPWICC

These above aspects (inputs, outputs, and fuzzy sets) are constructed to perform the control of congestion depending on the buffer occupancy and the dynamic topology of wireless ad hoc networks. For that aim, FPWICC uses both the buffer occupancy and the network mobility parameters. Thus, the previous fuzzy aspects can take various values:

- The first input parameter: is represented by the Admit Membership value (AM) at a mobile node. AM can be either small or large.
- The second input parameter: is represented by the Node Mobility (NM). NM can either be slow or medium (note that "fast node mobility" is included in the case of "medium node mobility").
- The output parameter: is represented by the Accepted Incoming packets into buffer (AI). AI can either be decreased (slowly or largely) or increased (slowly or largely).

FPWICC uses the previous rules to help to establish production rules that make an efficient QoS decision. In the following, we explain the proposed fuzzy tool for the QoS decision making.

Let consider the following fuzzy rule R_L:
Rule R_L: if AM is small and NM is slow, then AI is decreased.

R_L takes into consideration the input parameter of the admit membership value AM in the buffer and the node mobility NM in wireless ad hoc networks. The accepted incoming packets into buffer AI represents the output parameter.

FPN that models the dynamic aspect of the fuzzy rule R_L is illustrated in Fig. 3.

- P_{acd1}: models the antecedent condition 1 (acd_1) of R_L; acd_1 = "AM is small".
- P_{acd2}: models the antecedent condition 2 (acd_2) of R_L; acd_2 = "NM is slow".
- T_{aMf1}: models the membership function of the antecedent condition 1; $T_{aMf1=}u_{small}(DM)$.
- T_{aMf2}: models the membership function of the antecedent condition 2; $T_{aMf2=}u_{slow}(NM)$.
- P_{aMd1}: models the membership degree value of the condition 1 of a rule R_L. This value determines the satisfaction degree of the AM input parameter to the fuzzy set "small".
- P_{aMD2}: models the membership degree value of the condition 2 of a rule R_L. This value determines the satisfaction degree of the NM input parameter to the fuzzy set "slow".
- T_{FSCL}: models the operation of minimum composition "MIN" between the antecedent conditions (e.g. condition 1 and condition 2) of a rule R_L. The firing strength of R_L is represented by the MIN operation: $MIN(u_{small}(DM), u_{slow}(NM))$.
- P_{FSCL}: models the value of the firing strength of R_L. This value defines the degree of truth of the output proposition "AI is decreased".

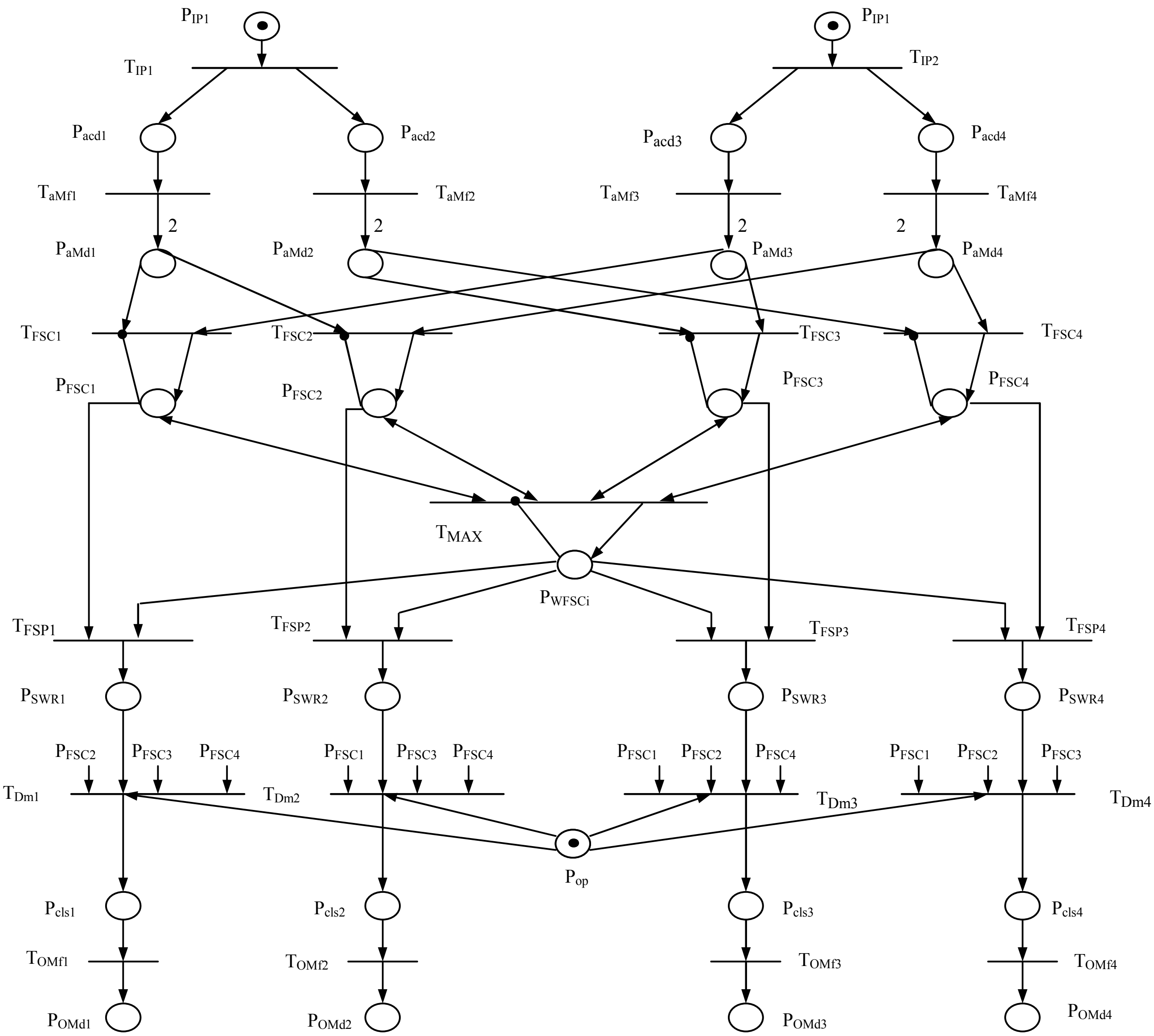


Fig. 5. FPWICC model

C. *Fuzzy Petri Nets Model for Congestion Control*

FPWICC considers the following rules:

R_1: if AM is large and NM is slow then AI is increased largely,
R_2: if AM is large and NM is medium then AI is increased,
R_3: if AM is small and NM is slow then AI is decreased,
R_4: if AM is small and NM is medium then AI is decreased largely.

- Input parameters:
  - The input parameter of the first antecedent condition of the rules R_1, R_2, R_3, and R_4 is the admit membership value AM.
  - The input parameter of the second antecedent condition of the rules R_1, R_2, R_3, and R_4 is the node mobility NM.
- Fuzzy sets:
  The fuzzy set of the antecedent conditions of the defined rules R_1, R_2, R_3, and R_4 are: small, large, slow, and medium.
- Antecedent conditions (acd_i):
  - The first antecedent condition (acd_1) in the rules R_1, R_2, R_3, and R_4 is:
    acd1: AM is small.
    acd2: AM is large.
  - The second antecedent condition (acd_2) in the rules R_1, R_2, R_3, and R_4 is:
    acd1: NM is slow.
    acd2: NM is medium.
- Output parameters:
  The output parameter of the rules R_1, R_2, R_3, and R_4 is the accepted incoming packets into buffer (AI).
- The rules R_1, R_2, R_3, and R_4 use the following decisions making: increased largely, increased, decreased, decreased largely,
- The fuzzy logic operator used by the rules R_1, R_2, R_3,

and R_4 is AND

The fuzzy operator "AND" is used in order to combine the two antecedent conditions of each rule using the MIN function. This provides the firing strength value for each rule. After that, MAX composition function is used to combine all firing strength values of the defined rules R_1, R_2, R_3, and R_4 in the aim of determining the highest one that will be the selected wining rule. Fig. 4 shows the fuzzy logic scheme for decision making of rules R_1, R2, R3, and R4.

In the following, we illustrate the steps of the proposed FPN model.

a. Enter the input parameters into the places and transitions:

- $P_{IP} = \{P_{IP1}, P_{IP2}, \ldots, P_{IPn}\}$ is a set of places that represent the input parameters. In the Fig. 5, the places used are P_1 and P_2 which represent respectively, the first (e.g. admit membership value AM) and second (e.g. node mobility NM) antecedent condition of the rules R_1, R_2, R_3, and R_4.
- $T_{IP} = \{T_{IP1}, T_{IP2}, \ldots, T_{IPn}\}$ represents a set of input parameter transitions. The transitions T_{IP1} and T_{IP2} illustrated in Fig. 5 are used to distribute respectively, the input parameters "AM" and "NM" for making the first and second antecedent conditions of the defined rules R_1, R_2, R_3, and R_4.

b. Represent the antecedent conditions, and compute the membership function for each condition.

- $P_{acd} = \{P_{acd1}, P_{acd2}, \ldots, P_{acdn}\}$ is a set of places that represent the antecedent conditions. P_{acd1} and P_{acd2} in the model presented in Fig. 5 describe respectively, the antecedent conditions "acd_1" and "acd_2".
- $T_{aMf} = \{T_{aMf1}, T_{aMf2}, \ldots, T_{aMfn}\}$ is a set of transitions that represent the antecedent membership functions. T_{aMf1}, T_{aMf2}, T_{aMf3}, T_{aMf4} observed in Fig. 5 represent the membership functions of respectively, $u_{small}(DM)$, $u_{large}(DM)$, $u_{slow}(NM)$, $u_{medium}(NM)$.
- $P_{aMd} = \{P_{aMd1}, P_{aMd2}, \ldots, P_{aMdn}\}$ is a set of places that represent the antecedent membership degrees. The values of the place P_{aMd1} indicates the degree of satisfaction of the input parameter AM to the fuzzy set "small".

c. Compute the firing strength of conditions

- $T_{FSC} = \{T_{FSC1}, T_{FSC2}, \ldots, T_{FSCn}\}$ represent a set of transitions that model firing strength conditions. For instance, the transition T_{FSC1} shown in Fig. 5 performs the operation of minimum composition (MIN) on the antecedent conditions of the rule R_1: $MIN(u_{small}(MD), u_{slow}(NM))$. Note that the fuzzy operator AND is integrated with the MIN operation to combine the first and second conditions of R_1.
- $P_{FSC} = \{P_{FSC1}, P_{FSC2}, \ldots, P_{FSCn}\}$ is a set of places that represent the firing strength. P_{FSCi} tokens are proportional to the number of antecedent conditions of a rule R_i. This number is shown by the label illustrated between the transitions T_{aMfi} and the place P_{aMdi}. The construction of the antecedent conditions of a rule R_i is performed by firing a transition T_{FSCi}. The inhibitor arc designed between a place P_{FSCi} and T_{FSCi} is useful to note that T_{FSCi} should fire one time.

d. Determine the selected wining rule among the activated rules:

- $T_{FMAX} = MAX\ \{P_{FSC1}, P_{FSC2}, \ldots, P_{FSCn}\}$ is a transition that models the maximum composition operation (MAX) for the defined rules. The firing strength value of a rule R_i is stored in the place P_{FSCi}.
- P_{WFSCi} represents the firing strength condition FSC_i of the selected wining rule R_i. The later rule is determined as in the following step.
- $T_{FSP} = \{T_{FSP1}, T_{FSP2}, \ldots, T_{FSPn}\}$ is a set of transitions that model the firing strength comparison. For instance, the transition T_{FSP3} is useful to make a comparison between FSC_3 of the rule R_3 and the selected wining firing strength $WFSC_i$.
- $P_{SWR} = \{P_{SWR1}, P_{SWR2}, \ldots, P_{SWRn}\}$ is a set of places that models the selected wining rules. The rule R_i is selected to be fired if the place P_{SWRi} contains a token.

e. The conclusion of the selected rules:

- $T_{Dm} = \{T_{Dm1}, T_{Dm2}, \ldots, T_{Dmn}\}$ is a set of transitions that represent the decision of the selected rule. T_{Dmi} deletes the firing strength values of other rules in order to fire only the selected rule R_i.
- P_{op} is a place that models the output parameter. As shown in Fig. 5, the place P_{op} represents the accepted incoming packets into buffer.
- $P_{cls} = \{P_{cls1}, P_{cls2}, \ldots, P_{clsn}\}$ models a set of places that describe the different decisions of the defined rules. The places P_{cls1}, P_{cls2}, P_{cls3}, and P_{cls4} illustrate the following conclusions respectively, "increased largely", "increased", "decreased", and "decreased largely". Only one place among all places will contain a token which represent the conclusion of the selected wining rule. For instance, the conclusion of the selected rule R_1 is "increased largely" if T_{Dm1} transfers a token from the place P_{SWR1} to the place P_{cls1}.
- $T_{OMf} = \{T_{OMf1}, T_{OMf2}, \ldots, T_{OMfn}\}$ is a set of transitions that represent the output membership functions. T_{OMf1}, T_{OMf2}, T_{OMf3}, and T_{OMf4} represent the calculation performed by the used fuzzy method to compute the membership degree of respectively, $u_{large\_increase}(TR)$, $u_{increase}(TR)$, $u_{decrease}(TR)$, $u_{large\_decrease}(TR)$,
- $P_{OMd} = \{P_{OMd1}, P_{OMd2}, \ldots, P_{OMdn}\}$ is a set of places that represent the output membership degree. The places P_{OMd1}, P_{OMd2}, P_{OMd3}, and P_{OMd4} indicate that the output parameters of "AI is increased", "AI is increased largely", "AI is decreased", and "AI is decreased largely" are satisfied with the following membership degree, $u_{large\_increase}(TR)$, $u_{increase}(TR)$, $u_{decrease}(TR)$, $u_{large\_decrease}(TR)$, respectively.

IV. SIMULATION

The simulation of the proposed QoS architecture is studied with ns-2 simulator. Each mobile host has a transmission range of 250 meters and shares an 11 Mbps radio channel with its neighboring nodes. We compare the performance of FuzzyCCG with the 'original model' and SWAN model described in [4]. We use the word 'original model' to refer to IEEE 802.11 wireless networks without FuzzyCCG mechanisms. The simulation is realized in two steps: the first one investigates the performance of the proposed model in an environment characterized by a single shared channel. The second simulation considers a multihop environment.

A. *Performance of a single shared channel*

We consider a single hop environment that consists of a square shape of 150m x 150m. The simulation includes a variety of traffic types; FTP macro-flows, WEB micro-flows, and real-time flows. The video and voice flows representing real-time traffic are active and monitored for the duration of 100 seconds. Video traffic is modeled as 200 Kbps constant rate traffic with a packet size of 512 bytes. Voice traffic is modeled as 32 Kbps constant rate traffic with a packet size of 80 bytes.

The simulation considers a multiple scenarios of TCP best-effort traffic, 4 voice and 4 video flows. The TCP traffic is modeled as a mixture of FTP and Web traffic. Web traffic represents micro-flows, whereas FTP traffic corresponds to macro-flows. TCP flows are greedy FTP type of traffic with packet size of 512 bytes. Web traffic is modeled as short TCP file transfers with random file size and random silent period between transfers. The file size is driven from a Pareto distribution with a mean file size of 10 Kbytes and a shape parameter of 1.2. The length of the silent period between two transfers is also Pareto in distribution with the same shape parameter with a mean of 10 seconds.

We explore in Figs. 6 and 7, the impact of scalability of number of UDP video flows on the average end-to-end delay. We consider a mixture of real-time traffic and TCP best-effort traffic which consists of 16 Web and FTP flows. It is observed in Fig. 6 that the original model shows an average delay larger than 12 msec with only 5 video flows and over 20 msec with 15 or more video flows. FuzzyCCG shows delays inferior to 2 msec with 5 video flows and less than 2.4 msec with 20 video flows. Hence, the reduction achieved by FuzzyCCG in terms of the average delay is about 90% in comparison to the original model. On the other hand, Fig. 7 illustrates that for up to 20 video flows, FuzzyCCG outperforms SWAN by about 13%.

Figs. 8 and 9 show the impact of the scalability of a growing number of web micro-flows on the average end-to-end delay. It is observed in Fig. 8 that the increasing number of web micro-flows has much more impact on the average delay in the original model than in FuzzyCCG. The average delay in FuzzyCCG remains around 1.8 msec, whereas in the original model the average end-to-end delay grows from 1.8 to 7 msec when the number of web micro-flows increases from 8 to 72 web micro-flows. On the hand, it is observed in Fig. 9 that the average delay in SWAN and FuzzyCCG models is similar for up to 16 web micro-flows. For the highest number web micro-flows, the average delay of traffic in FuzzyCCG becomes smaller than in SWAN by about 18%.

B. *Performance in multihop environment*

In what follows, the simulation considers a multihop network of 50 mobile nodes. The network area has a rectangular shape of 1500m x 300m. The AODV protocol [28] is chosen as a routing protocol. The flows traverse three intermediate nodes on average between source and destination. In this multihop network, we consider a mixture of real-time and TCP best-effort traffic. The real-time traffic is modeled as 4 voice and 4 video flows. The TCP traffic is modeled as a mixture of web micro-flows and FTP macro-flows traffic.

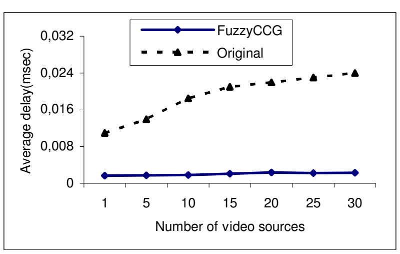


Fig. 6. Average delay in the original and FuzzyCCG models vs. number of video flows

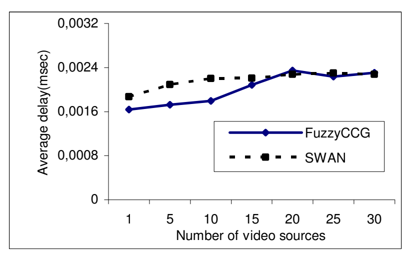


Fig. 7. Average delay in FuzzyCCG and SWAN models vs. number of video flows

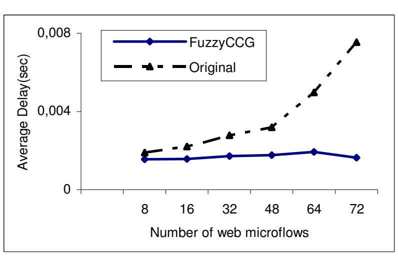


Fig. 8. Average delay in the original and FuzzyCCG models vs. number of web micro-flows

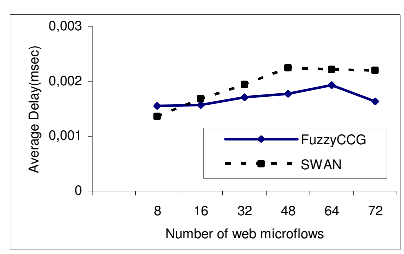


Fig. 9. Average delay in FuzzyCCG and SWAN models vs. number of web micro-flows

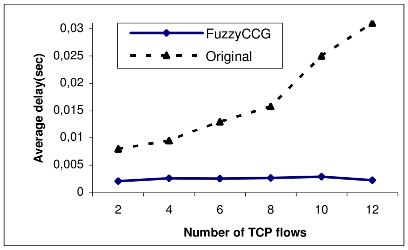


Fig. 10: Average delay in the original and FuzzyCCG models vs. number of TCP flows

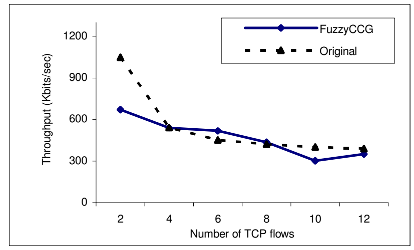


Fig. 11: Average throughput in the original and FuzzyCCG models vs. number of TCP flows

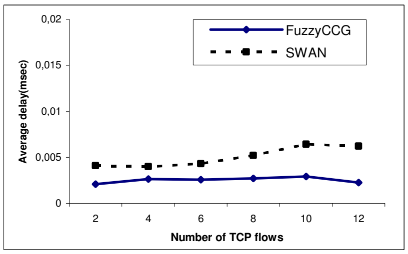


Fig. 12: Average delay in FuzzyCCG and SWAN models vs. number of TCP flows

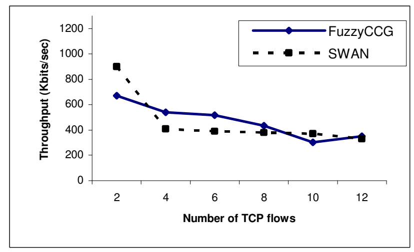


Fig. 13: Average throughput in FuzzyCCG and SWAN models vs. number of TCP flows

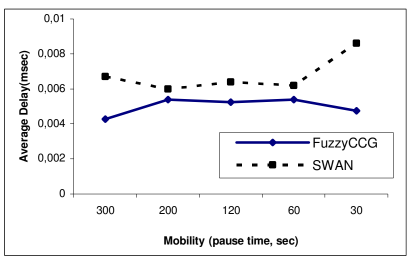


Fig. 14. Average delay in the original and FuzzyCCG models vs. mobility

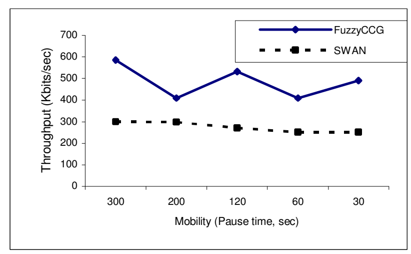


Fig. 15. Average throughput in the original and FuzzyCCG models vs. mobility

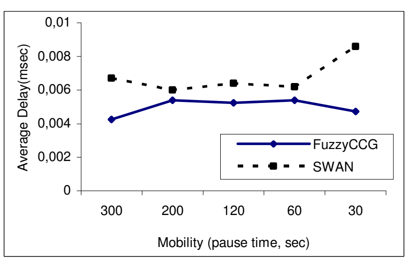


Fig. 16. Average delay in FuzzyCCG and SWAN models vs. mobility

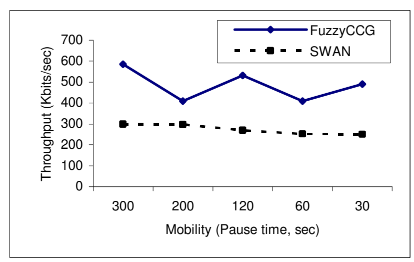


Fig. 17. Average throughput in FuzzyCCG and SWAN models vs. mobility

Figs. 10-13 explore the scalability impact of the increasing number of TCP flows on the average end-to-end delay and throughput of traffic. Fig. 10 illustrates a significant difference in terms of the average delay between FuzzyCCG and the original model. The average delay in FuzzyCCG grows slowly with the increasing number of TCP flows, and it remains between 2 and 3 msec. In contrast, the average delay in the original model grows from 7 to 31 msec as the number of TCP flows increases from 2 to 12 flows. Hence, the gain achieved by FuzzyCCG in terms of the average end-to-end delay, is by about 74-92%. Fig. 12 shows the average end-to-end delay in both FuzzyCCG and SWAN models. It is shown that the average delay is almost inferior to 3 msec in the proposed model, whereas in SWAN model the average delay is around 5 msec. This means that the achieved gain offered by FuzzyCCG is about 49% in terms of average delay.

Figs. 11 and 13 illustrate the impact of growing number of TCP flows on the average throughput of TCP traffic over the models of simulation. The average throughput of the TCP traffic in FuzzyCCG is almost the same as in the original model, as shown in Fig. 11. At lower number of TCP flows, the average throughput in the original model is superior to that in FuzzyCCG. A similar result is observed in Fig. 13 between FuzzyCCG and SWAN.

Figs. 14-17 explore the impact of mobility on the performances of FuzzyCCG. The real-time traffic is modeled in the same manner as discussed previously. The best-effort TCP flows consists of 5 web flows and 5 FTP flows. The random waypoint mobility model is implemented at each node in the network. In the beginning, the nodes are randomly placed in the area. Then, each mobile node selects a random destination and moves with a random speed up to a maximum speed of 20m/s. After reaching the destination, the node will stay there for a given "pause time" then starts to move towards another destination. This process is repeated during all simulation time.

It is observed in Fig. 14 that the average end-to-end delay in FuzzyCCG increases slowly. The average delay in the proposed model remains almost less than 5.4 msec, whereas the average delay in the original model grows from 25 to 38 msec. This means that the proposed FuzzyCCG achieves a reduction in terms of average delay by about 79-87%. On the other hand, it is observed in Fig. 15 that the throughput of TCP best-effort traffic decreases slowly in the original model as the mobility increases. The average throughput in FuzzyCCG is superior to that of the original model by about 33% for different mobility scenarios.

Fig. 16 shows the average end-to-end delay with different mobility scenarios in both FuzzyCCG and SWAN models. For different mobility scenarios, the average delay offered by FuzzyCCG is about 10-36% better than that offered by SWAN. Fig. 17 shows that for different mobility scenarios, the throughput in FuzzyCCG is better than in SWAN model by about 43%.

The previous results show that the proposed architecture provides an average end-to-end delay with low and almost stable values, which is promising result for jitter-sensitive applications.

V. CONCLUSION

In this paper we proposed an intelligent solution for the congestion control of multimedia applications. The presented approach includes a fuzzy logic technique for buffer threshold management in order to show the ability of fuzzy thresholds to adapt to the dynamic conditions over the classical inflexible thresholds. In addition, the proposed solution includes a new technique based on fuzzy Petri nets to model and analyze the QoS decision making for buffer management in wireless ad hoc networks. It is observed in the simulation results that the proposed architecture can achieve a significant reduction in terms of the average end-to-end delay in comparison to both IEEE 802.11 and SWAN models. The obtained results confirm that the intelligent-based solutions can offer a good QoS support for multimedia services.

REFERENCES

[1] S. Chakrabati, A. Mishra, "QoS issues in ad hoc wireless networks", IEEE Commun. Magazine, vol. 39, N. 2, Feb. 2001, pp. 142-148.

[2] D. Black, S. Blake, M. Carlson, E. Davies, Z. Wang, W. Weiss, "An Architecture for Differentiated Services", IETF RFC 2475, Dec. 1998.

[3] R. Braden, D. Clark, and S. Shenker, "Integrated services in the Internet architecture: an overview". IETF RFC 1363, June 1994.

[4] G.H. Ahn, A. T. Campbell, A. Veres, and L. H. Sun, "SWAN: Service Differentiation in Stateless Wireless Ad Hoc Networks," In the Proc. of IEEE INFOCOM, June 2002.

[5] S.-B. Lee, G.-S. Ahn, X. Zhang, and A.T. Campbell, "INSIGNIA: An IP-Based Quality of Service Framework for Mobile Ad Hoc Networks," Journal of Parallel and Distributed Computing, special issue on wireless and mobile computing and communication, vol. 60, no. 4, Apr. 2000, pp. 374-406.

[6] H. Xiao, W. K.G. Seah, A. Lo, and K. Chaing, "Flexible QoS Model for Mobile Ad-hoc Networks," In the Proc. of IEEE Vehicular Technology Conference, vol. 1, pp 445-449, Tokyo, May 2000.

[7] Y. L. Morgan, T. Kunz, "PYLON: An architectural framework for ad-hoc QoS interconnectivity with access domains," HICSS'03 pres, Hawaii, USA, Jan. 2003.

[8] J.L. Sobrinho and A.S. Krishnakumar, "Quality-of-Service in Ad Hoc Carrier Sense Multiple Access Networks," IEEE Journal on Selected Areas in Communication, vol. 17, no. 8, pp. 1353-1368, Aug. 1999.

[9] L. Khoukhi, S. Cherkaoui, "A Quality of Service Approach Based on Neural Networks for Mobile Ad hoc Networks," IEEE-IFIP Int. Conf. on Wireless and Optical Communications Networks, WOCN 2005, Mar. 2005.

[10] L. Khoukhi, S. Cherkaoui, "FuzzyMARS: A Fuzzy Logic Approach with Service Differentiation for Wireless Ad hoc Networks," In the Proc. of IEEE WirelessCom2005, Jun. 2005.

[11] C. R. Lin and J.-S. Liu. "QoS Routing in Ad Hoc Wireless Networks," IEEE Journal on Selected Areas in Communication, vol. 17, no. 8, 1999, 1426–1438.

[12] L. Khoukhi, S. Cherkaoui, "Flexible QoS Routing Protocol for Mobile Ad Hoc Networks," In the Proceedings of the 11th IEEE Int. Conf. Telecommunication (ICT2004), Brazil, Aug. 2004.

[13] S. Chen and K. Nahrstedt. "Distributed Quality-of-Service in Ad Hoc Networks," IEEE Journal on Selected Areas in Communications, vol. 17, no. 8, 1999, pp.1426–1438.

[14] C.-R. Lin. "On-Demand QoS Routing in Multihop Mobile Networks," In Proc. of IEEE INFOCOM 2001, pp. 1735–1744, April 2001.

[15] S. Chen and K. Nahrstedt, "On finding multi-constrained paths," IEEE International Conference on Communication, 874 -879 June 98.

[16] C.C Lee, "Fuzzy logic in control systems: fuzzy logic controller- part I and II," IEEE trans. on Systems, Man, and Cybernetics, vol. 20, no. 2, 1990, pp. 404-418.

[17] L. A. Zadeh, "fuzzy logic = computing with words", IEEE Trans. on fuzzy systems," vol. 4, no. 2, 1996, pp. 104-111.

[18] A. R. Bonde and S. Ghosh, "A comparative study of fuzzy versus Fixed thresholds for robust queue management in cell-switching networks," IEEE Trans. Networking, vol. 2, no. 4, Aug. 1994, pp. 337–344.

[19] M. B., Dwyer, and L. A., Clarke, "A compact Petri net representation and its implication for analysis," IEEE Trans. Software Engineering, 22, 1996, pp. 794-811.

[20] T., Murata, T., Suzuki, and S. M., Shatz, "Fuzzy-timing high-level Petri net model of a real-time network protocol," In the proc. of ITC-CSCC 96, Seoul, Korea, July 1996, pp. 1170-1173.

[21] S.-M., Chen, J.-S., Ke, and J.-F., Chang, "knowledge representation using fuzzy Petri nets," IEEE Trans. on Knowledge Data Engineering, 2, 1990, 311-319.

[22] G., Looney, "Fuzzy Petri nets for rule-based decision making," IEEE Trans. on System, Man, and Cybernetics, 18, 1998, pp. 178-183.

[23] T., Cao, A.C., Sanderson, "Variable reasoning and analysis about uncertainty with fuzzy Petri nets," In the proceeding of 14 th Int. conf. on application and theory of Petri nets, Troy, NY, August, pp. 126-175.

[24] S.I., Ashon, "Petri net models of fuzzy neural networks," IEEE Trans. on System, Man, and Cybernetics, 25, 1995, pp. 926-932.

[25] A., Chaudhury, D. C., Marinescu, and A., Whinston, "Net-based computational models of knowledge processing systems," IEEE Expert, 8, 1993, pp. 79-86.

[26] L. A., Zadeh, "Knowledge representation in fuzzy logic," IEEE Trans. knowledge Data Engineering, 1, 1989, pp. 89-100.

[27] L. Khoukhi, S. Cherkaoui, "A Fuzzy Petri Nets QoS Model for Wireless Ad hoc Networks, The 2005 IFIP Int. Conf. on Intelligence in Communication Systems (INTELLCOMM), October 2005.

[28] C.E. Perkins, E.M. Royer, "Ad-hoc on-demand distance vector routing," IEEE Workshop on Mobile Computing Systems and Applications, pp.90 –100. New Orleans, LA, Feb. 1999.