Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. VI (2011), No. 3 (September), pp. 462-472

Distance Based Triggering and Dynamic Sampling Rate Estimation for Fuzzy Systems in Communication Networks

C.N. Nyirenda, F. Dong, K. Hirota

Clement N. Nyirenda, Fangyan Dong, Kaoru Hirota
Department of Computational Intelligence and Systems Science
Tokyo Institute of Technology
G3-49, 4259 Nagatsuta, Midori-ku, Yokohama 226-8502, Japan.
E-mail: {nyirendac, tou, hirota}@hrt.dis.titech.ac.jp

Abstract: To reduce computational cost in fuzzy systems in communication networks, distance based triggering and sampling rate adaptation probabilities are proposed based on the concept of *probability via expectation*. The triggering probability, which is calculated by using the square of distance between subsequent input vectors, governs the rate at which the fuzzy system is triggered. The dynamic sampling rate probability, which governs the adaptation of the sampling rate, is computed by using the exponentially weighted moving average (EWMA) of the triggering probability. A stopping criterion, based on convergence tests, is also proposed to ensure that the mechanism switches off when the sampling period has converged. The triggering mechanism reduces the number of computations in the Fuzzy Logic Congestion Detection (FLCD) in wireless Local Area Networks (WLANs) by more than 45%. Performance, in terms of packet loss rate, delay, jitter, and throughput, however, remains virtually the same. On the other hand, the dynamic sampling rate mechanism leads to more than 150% improvement in sampling rate and more than 70% reduction in fuzzy computations while performance in the other key metrics remains virtually the same. As part of future work, the proposed mechanism will be tested in fuzzy systems in wireless sensor/actuator networks.
Keywords: communication networks, fuzzy systems, sampling rate.

1 Introduction

The number of fuzzy logic based applications in communication networks is increasing rapidly. This development is motivated [1] by the difficulties experienced when modeling communication networks by using conventional analytical methods. Some of the fuzzy applications include power control [2] in cellular systems; congestion control in IP netwoks [3], [4]; routing [5] and data fusion [6] in wireless sensor networks; and Quality of Service management in wireless sensor and actuator networks [7]. Input parameters are, generally, sampled at a fixed rate and the fuzzy system is triggered accordingly. In some cases, an external signal is used in order to trigger the systems. The fuzzy computations are invoked even when there are no significant differences between the subsequent input parameters, at the expense of precious CPU and memory resources. Furthermore, for systems that employ a sampling rate, the rate is chosen by trial and error such that it is difficult to tell if it is optimal.

This work proposes a distance based triggering mechanism for fuzzy systems in communication networks. This generic framework can be applied to any fuzzy system with minor customization. A preliminary version of this work was presented in [8]. In this work, the concept of *probability via expectation* [9] is used to calculate the triggering probability by using the square of the distance between two subsequent input vectors. If the new input parameter vector is very far from the previous one, the triggering probability is 1. If subsequent input vector parameters are deemed to be very close to each other, the triggering probability is 0.

A sampling rate adaptation probability, which is based on the transformed Lorentzian function [10] of the exponentially weighted moving average (EWMA) of the triggering probability, is also proposed. When the EWMA is very high, it implies that the system's input vector is under-sampled. As a result, the sampling rate is increased. On the other hand, when the EWMA is low, it implies that the system's input vector is oversampled. The sampling rate is, therefore, decreased. A stopping criterion, based on convergence tests [11], [12] of subsequences of the sequence of the sampling period, is also proposed in order to ensure that the mechanism switches off when the sampling period has converged.

These mechanisms are tested on the Fuzzy Logic Congestion Detection (FLCD) mechanism [13] in the wireless Local Area Network (WLAN) environment using simulations on the NS2 platform [14] running on the Ubuntu 9.10 OS. The computing hardware is composed of a 4GB RAM and an Intel Core i7 860 2.80GHz CPU. The impact of the reduction in fuzzy system evaluations on packet loss rate, delay, jitter and throughput is also evaluated.

The rest of the paper is organized as follows. The proposed distance based triggering and the dynamic sampling rate mechanisms are presented in 2. An overview of the FLCD approach in WLANs is presented in 3. The evaluation of the proposed mechanisms is presented in 4.

2 The Distance Based Triggering and Dynamic Sampling Rate Mechanisms

The distance based triggering and the dynamic sampling rate mechanisms are incorporated in the input mechanism of the generic fuzzy logic control framework. Every τ seconds, the new crisp inputs $x_1(t), ..., x_N(t)$ are normalized to the range $[0, 1]$. The triggering probability $P(Tr)$, which governs the system's triggering rate, is calculated by using the distance between the new normalized crisp input vector $(x_1^*(t), ..., x_N^*(t))$ and the previous normalized crisp input vector $(x_1^*(t-\tau), ..., x_N^*(t-\tau))$. If the system is not triggered, the previous crisp outputs are used for decision making or control action. When the system has been triggered, fuzzy computations [15] [16] are carried out to generate new output(s).

The dynamic sampling rate mechanism uses $P(Tr)$ to compute the sampling rate adaptation probability $P(Ra)$, based on which the value of τ is adapted to an optimal level. Both $P(Tr)$ and $P(Ra)$ are developed by using the concept of *probability via expectation* [9]. According to this concept, an event A, which, in a given case, either occurs or does not, corresponds to a set of realizations ω. This set, also denoted by A, is a subset of Ω, which denotes the sample space of all possible realizations. The *probability of A*, the expected proportion of cases in which event A actually occurs, is defined as $P(A) = E(I(A, \omega))$, where $I(A, \omega)$ is the *indicator function of A*, defined by

$$I(A, \omega) = \begin{cases} 1 & (\omega \in A) \\ \\ 0 & (\omega \notin A). \end{cases} \tag{1}$$

2.1 Distance Based Triggering Probability

The event of interest is the triggering of the fuzzy system, denoted by Tr. A realization for which the event Tr takes place is denoted by ω_1, while Ω_1 denotes the sample space of all possible realizations for which triggering is considered.

To reduce computational overhead when calculating the distance based triggering probability $P(Tr)$, the square of distance $d^2(t)$ between $(x_1^*(t), ..., x_N^*(t))$ and $(x_1^*(t-\tau), ..., x_N^*(t-\tau))$ is used, where $d^2(t) = (x_1^*(t) - x_1^*(t-\tau))^2 + ... + (x_N^*(t) - x_N^*(t-\tau))^2$. Let $v_1 : [0,1]^N \longrightarrow [0,1]$ denote the distance parameter that defines the variation of ω_1 with respect to Ω_1. This parameter is defined by using

$$v_1 = \begin{cases} \frac{d^2(t)}{\Phi} & \text{if } d^2(t) < \Phi \\ \\ 1.0 & \text{otherwise.} \end{cases} \tag{2}$$

where Φ is a normalizing constant. The triggering probability $P(Tr)$ is defined as $P(Tr) = E(I(Tr, \omega_1))$; the indicator function $I(Tr, \omega_1)$ is defined by using

$$I(Tr, \omega_1) = \begin{cases} 1 & (v_1 > R_1) \\ \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

where $R_1 \in [0,1]$ is a random number.

A high $P(Tr)$ implies that the distance between the new input vector and the previous one is large. Therefore, there is sufficient new information such that the fuzzy system has to be triggered. On the other hand, a low $P(Tr)$ implies that the change in the input vector is not significant. Therefore, the system should use the previous crisp outputs thereby preserving CPU and memory resources.

2.2 Sampling Rate Adaptation Mechanism

Apart from governing the triggering rate, $P(Tr)$ also gives information on whether the sampling period τ is optimal or not. If $P(Tr)$ is very high for long periods, it implies that the system is predominantly under-sampled such that there is a need to increase the sampling rate. Conversely, if $P(Tr)$ is very low for long periods, it implies that the inputs are more or less static. The sampling rate must be reduced because the frequent processing of the inputs is just a waste of computing resources. An inverted bell shaped function must, therefore, be employed in order to ensure that there are little or no changes to τ when $P(Tr)$ is close to 0.5, which signifies an optimal sampling rate. On the other hand, in extreme regions, τ must either be decreased or increased by an adaptation factor α. A transformed Lorentzian function is employed to generate a bell shaped function depicting this behavior. The traditional three-parameter Lorentzian function [10], from which it is derived, is defined by

$$L(z) = A_p \left[\frac{\gamma^2}{(z - z_0)^2 + \gamma^2} \right], \tag{4}$$

where z_0 is the centre; γ is the width parameter; and A_p determines the peak.

Sampling Rate Adaptation Probability

To track the variations of $P(Tr)$ at time t, the *exponentially weighted moving average (EWMA)* of $P(Tr, t)$, denoted by $\overline{P(Tr, t)}$, is determined by using

$$\overline{P(Tr, t)} = w * \overline{P(Tr, (t - \tau))} + (1 - w) * P(Tr, t), \tag{5}$$

where $w \in [0,1]$ is a weighting factor. To ensure that previous values of $\overline{P(Tr, t)}$ are discounted at a medium rate, $w = 0.5$ is used in this study.

The event depicting the adaptation of the sampling rate of the fuzzy system is denoted by Ra. A realization for which this event takes place is denoted by ω_2, while Ω_2 denotes the sample space of all possible realizations for which the adaptation of the sampling rate is considered. In a similar approach to 2.1, let $v_2 : [0,1] \longrightarrow [0,1]$ denote the rate adaptation parameter that defines the variation of ω_2 with respect to Ω_2 by using the transformed Lorentzian function. The relationship between v_2 and $\overline{P(Tr,t)}$ is, therefore, defined by

$$v_2 = L^*(\overline{P(Tr,t)}) = \left[\frac{(\overline{P(Tr,t)} - 0.5)^2}{(\overline{P(Tr,t)} - 0.5)^2 + \gamma^2} \right]. \tag{6}$$

where γ is defined by

$$\gamma = f(\overline{P(Tr,t)}) = \begin{cases} \overline{P(Tr,t)} & \text{if } \overline{P(Tr,t)} \leq 0.5 \\ \\ 1 - \overline{P(Tr,t)} & \text{if } \overline{P(Tr,t)} > 0.5 \end{cases} \tag{7}$$

The sampling rate adaptation probability is defined as $P(Ra) = E(I(Ra, \omega_2))$; the indicator function $I(Ra, \omega_2)$ is defined by

$$I(Ra, \omega_2) = \begin{cases} 1 & (v_2 > R_2) \\ \\ 0 & \text{otherwise,} \end{cases} \tag{8}$$

where $R_2 \in [0,1]$ is a random number. If $P(Ra) = 1$, τ is adjusted by using

$$\tau = \begin{cases} (1-\alpha)\tau & \text{if } \overline{P(Tr,t)} \geq 0.5 \\ \\ (1+\alpha)\tau & \text{otherwise.} \end{cases} \tag{9}$$

This mechanism will help to ensure that $\overline{P(Tr,t)} \to 0.5$. The sampling period can be initialized randomly within a particular range. This mechanism will optimize it on-line based on the variations in the inputs. This characteristic is essential for new links and in situations where new nodes or traffic patterns have been introduced on the already existing links.

Stopping Criterion for Sampling Rate Adaptation

Once the dynamic sampling rate mechanism starts running, there is a need for a stopping criterion; otherwise, this mechanism will end up consuming precious CPU and memory resources even when the sampling period has converged. The evolution of the sampling rate τ can be presented as a sequence (τ_k) based on the concept of sequences of real numbers discussed in [11], [12]. When convergence tests show that the sequence has converged, it implies that an optimal sampling period has been realized. Therefore, the sampling rate estimation mechanism must be stopped.

Definition 1. A sequence (τ_k) converges to τ if for every $\epsilon > 0$, there exists a $K \in \mathbb{N}$ such that $|\tau_k - \tau| < \epsilon$ for all $k \geq K$. The point τ is called the limit of (τ_k).

If the terms of the sequence get arbitrarily close together, a sequence is said to be *Cauchy*.

Definition 2. A sequence (τ_k) is said to be a Cauchy sequence if for every $\epsilon > 0$, there exists a $K \in \mathbb{N}$ such that $|\tau_k - \tau_m| < \epsilon$ for all $k, m \geq K$.

Every convergent sequence is a Cauchy sequence. For real numbers, the converse is also true; every Cauchy sequence is convergent. In addition, a sequence is convergent if and only if all of its subsequences converge toward the same limit. A *subsequence* of the sequence (τ_k) is a sequence of the form (τ_{k_j}), where for each $j \in \mathbb{N}$, there is $k_j \in \mathbb{N}$, and $k_j < k_{j+1}$ for all j. From these observations, we have the following theorem

Theorem 3. *If the first term of a convergent subsequence, $\tau_{k_1} = \tau_K$, then for the first p terms,*

$$\frac{1}{p-1}\sum_{j=1}^{p-1}\left|\tau_{k_{j+1}} - \tau_{k_j}\right| < \epsilon. \tag{10}$$

Proof: This follows from *Definition 1* and *Definition 2*. Because the subsequence (τ_{k_j}) is convergent, it is also a *Cauchy sequence*. Therefore for every $\epsilon > 0$, there exists a $K \in \mathbb{N}$ such that $\left|\tau_{k_j} - \tau_{k_n}\right| < \epsilon$ for all $k_j, k_n \geq K$. This implies that the average of the absolute values of the differences of subsequent p terms is also less than ϵ. □

By comparing the average of the absolute values of the differences of subsequent terms in a subsequence (τ_{k_j}) with ϵ, it is possible to determine whether the original sequence (τ_k) has converged or not. The terms of the subsequence are extracted from (τ_k) every $L * \tau(t)$ seconds, where $L \in \mathbb{N}$. To reduce computational overhead and memory requirements, p is set to 4; on the other hand, $\epsilon = 1 \times 10^{-5}$. While the focus is on detecting convergence of (τ_k), it must be pointed out that cases where this sequence does not converge should also be anticipated. In such cases, the sampling rate estimation mechanism, once activated, will remain on until the system administrator decides to stop it.

The proposed mechanisms are tested on the zero-order Takagi-Sugeno [16] inference based FLCD approach [13] in WLANs. Next, an overview of the FLCD algorithm is described.

3 An Overview of the FLCD Approach in WLANs

The FLCD mechanism is a 2-input 1-output system. It is composed of the Fuzzy Logic Control Unit (FLCU), the Congestion Notification Unit (CNU), and the CHOKe [17] Activator (CA) as shown in **Figure 1**. The FLCU uses the backlog (queue size) factor x_1 and the packet arrival factor x_2 to generate the packet marking probability p_b. The queue on the outgoing link is sampled at a period $\tau = 2$ msec in order to obtain the two inputs. The CNU either marks (if ECN is enabled) or drops packets with a probability p_b. Responsive flows such as TCP react to these events by reducing their sending rates thereby reducing congestion at the bottleneck link.

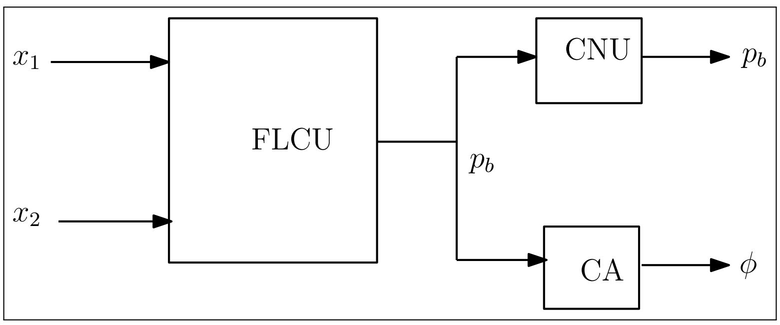

Figure 1: The FLCD Mechanism.

For purposes of fairness, in light of non-responsive flows and network anomalies such as Denial of Service (DoS) attacks and routing loops, which may flood the network as the responsive flows back off, the CA uses p_b to generate a parameter $\phi \in [0, 1]$, where $\phi = p_b^3$. The CA probabilistically picks an arriving packet picked based on the value of ϕ. This packet is compared with a randomly chosen packet from the buffer. If they have the same flow ID, they are both dropped. Otherwise the randomly chosen packet is left unchanged and the arriving packet is queued if the buffer is not full; otherwise it is dropped. As a result, more packets from non-responsive and TCP-unfriendly flows are dropped at the bottleneck link.

4 Evaluation of the Distance Based Triggering and the Dynamic Sampling Mechanisms

The distance based triggering and the dynamic sampling rate mechanisms are implemented at the input of the FLCD mechanism which is used for congestion control in the access point (AP) of the WLAN topology shown in **Figure 2**. The objective is to reduce congestion for traffic flowing from the servers on the high-speed wired/cabled network to the nodes in the bandwidth constrained wireless network. Simulations are implemented on the NS2 simulation platform [14]. In **Figure 2**, servers S1, S2 are connected to the Gateway which is connected to AP. BW1 and

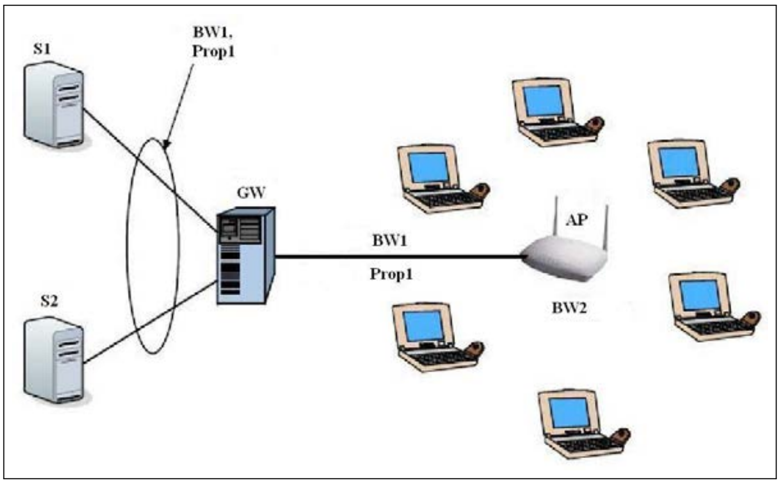

Figure 2: The used WLAN topology.

Prop1 denote the bandwidth and propagation delay between the servers and the Gateway and between the Gateway and the AP while BW2 denotes the wireless channel capacity. The wireless nodes are equidistant from the AP. For brevity, only 5 nodes are shown in **Figure 2** but more than 5 nodes are used in the simulations. Half of the nodes in the wireless network are fixed while the other half consists of mobile nodes. To depict a *high-speed topology*, BW1 and Prop1 are set to 10Gbps and 1ms respectively while the capacity, BW2, is set to 144 Mbps.

Two sets of experiments are conducted. The first one evaluates the impact of the distance based triggering mechanism on the FLCD algorithm. It is aimed at comparing the number of fuzzy system evaluations in a FLCD algorithm with and without the distance based triggering mechanism. The other objective is to find out the impact of the distance based triggering mechanism on system performance. Metrics for system performance include packet loss rate, link utilization, packet delay, and jitter. The second experiment evaluates the efficiency of the dynamic sampling rate mechanism. The objective is to find out if it really manages to guide the system toward the optimal sampling rate. Convergence times are also captured.

Simulations are configured as follows. Each simulation run takes 100 seconds. In all runs, one FTP flow and one web traffic flow are configured to flow from Server 1 to each of the nodes in the WLAN between 5 seconds and 95 seconds leaving enough time for the simulation to start up and also to shut down gracefully. The standard web traffic generator included in the NS2 platform is used with the following parameter settings: an average of 30 web pages per session, an inter-page parameter of 0.8, an average page size of 10 objects, an average object size of 400 packets and a ParetoII shape parameter of 1.002. Each web traffic flow has 4 sessions. UDP traffic is also configured to flow from Server 2 to 10 nodes in the intervals [25s-30s] and [80s-90s]. Parameters of UDP traffic are as follows: Packet Size of 1500bytes, packet interval of 12.5ms and a flow rate of 120kbytes/sec. TCP type is New Reno with a data packet size of 1000 bytes and ACK packet size of 40 bytes. The buffer size is set to 200 bytes.

4.1 Experiment 1 - Testing the impact of the distance based triggering mechanism

Only the distance based triggering mechanism is activated. The number of WLAN nodes is varied by using 10, 20, 30 up to 100 nodes. The value of the normalizing constant Φ in (2) is set to 2×10^{-2} based on several trial runs. After that, simulations are carried out. The overall results are average values over 20 independent runs, which are conducted at each and every testing point. These values are plotted in the graphs along with the error bars representing the 99% confidence intervals of the averages. The FLCD mechanism that employs the proposed triggering mechanism is labeled FLCD+D while the normal one is labeled FLCD. **Figure 3 - Figure 7** show the results.

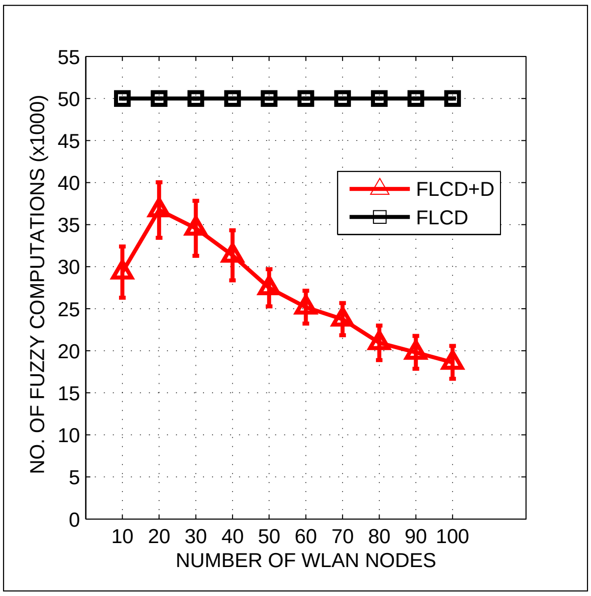

Figure 3: Number of fuzzy computations as congestion level increases

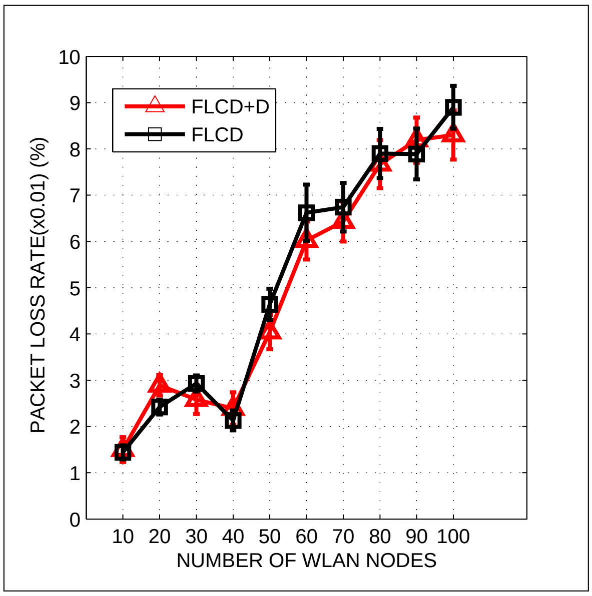

Figure 4: Packet loss rate as congestion level increases

In **Figure 3**, the distance based triggering mechanism reduces the number of fuzzy computations by more than 45% while in **Figure 4 - Figure 7**, packet loss rate, delay, jitter, and throughput remain virtually the same when the distance based triggering mechanism is employed. These results confirm the fact that a great deal of computing power is lost due to redundant computations in fuzzy systems in communication networks.

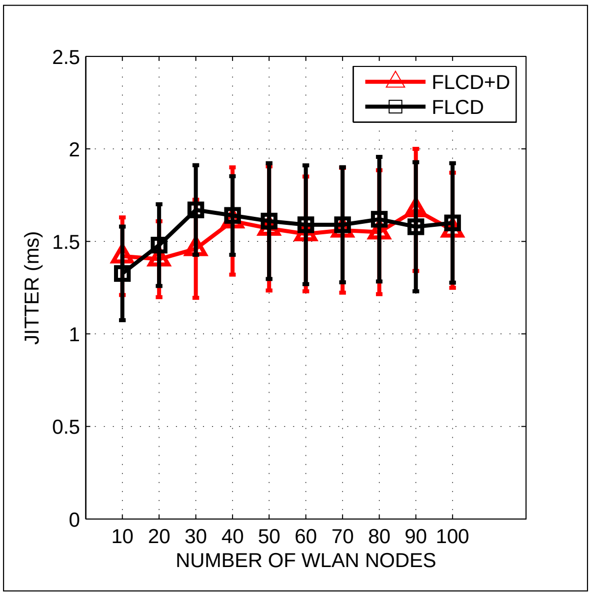

Figure 5: Packet jitter as congestion level increases

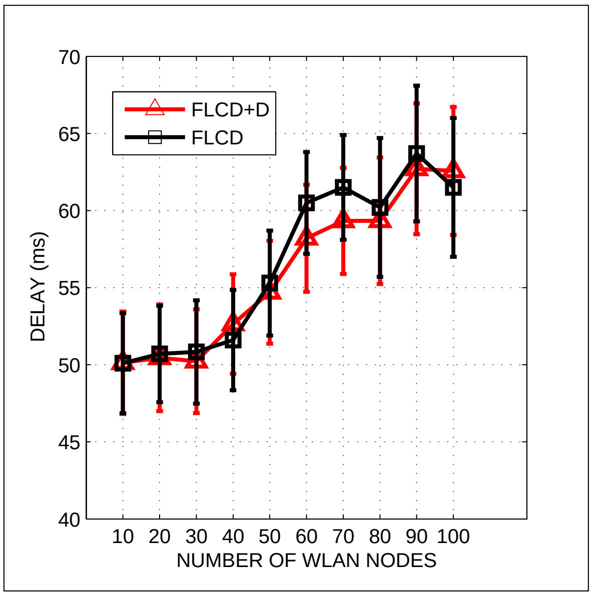

Figure 6: Packet delay as congestion level increases

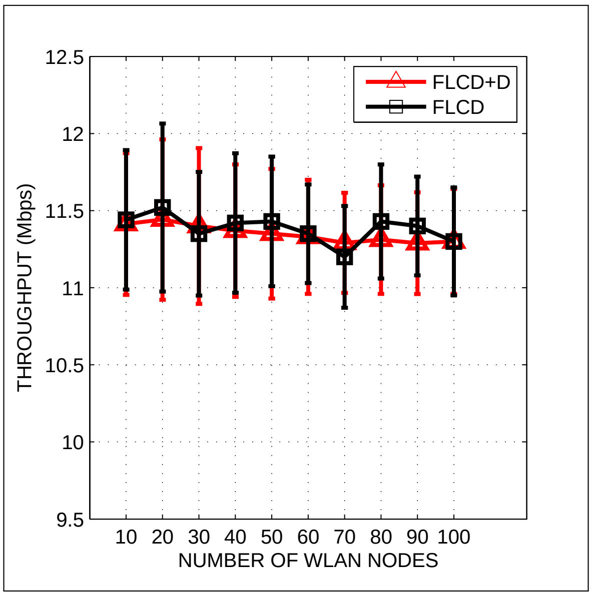

Figure 7: Throughput as congestion level increases

4.2 Experiment 2 - Efficiency of the dynamic sampling rate mechanism

Both the distance based triggering and the dynamic sampling rate mechanisms are activated. The FLCD mechanism that employs the proposed distance based triggering mechanism is labeled FLCD+DS while the normal one is labeled FLCD. The following parameter values are used: $L = 10$, $\alpha = 1 \times 10^{-3}$. These values are determined after several trial runs. Simulations are carried out with 50 nodes on the wireless network on every run. The overall results are average values over 20 independent runs, which are conducted at each and every testing point. In FLCD+DS, the sampling period τ is initialized randomly within (0,1] seconds. Based on the existing congestion control literature [3], $\tau = 2$ msec in FLCD.

Apart from the performance metrics used in Experiment 1, the averages of the converged sampling periods τ and the convergence times are also presented. To statistically establish if two corresponding averages are significantly different from each other, the one-way ANOVA test, at a significance level of 1%, is applied. **Table 1** shows the results. Columns 2 and 3 denote the corresponding average and standard deviation values for FLCD+DS and the FLCD respectively. Column 4 denotes the percentage improvement that FLCD+DS exhibits over FLCD. Statistically significant percentage improvements are shown in bold.

Table 1: Dynamic Sampling Rate Mechanism vs the Conventional approach for 50 nodes.

Metric	FLCD+DS	FLCD	Improv.
Loss rate (%)	0.055 ± 0.0047	0.057 ± 0.0051	3.5%
Delay (sec)	54.59 ± 4.26	52.91 ± 5.54	−3.01%
Jitter (sec)	1.576 ± 0.136	1.59 ± 0.123	0.88%
Throughput (Mbps)	11.47 ± 0.372	11.49 ± 0.415	0.17%
Evaluations	12687 ± 1149	50000	**74%**
τ (msec)	5.05 ± 1.634	2	**152.5%**
Conv. time (msec)	15.69 ± 5.37	–	–

The proposed mechanisms reduce the number of evaluations by more than 70%. The converged sampling period in FLCD+DS is also increased by more than 150%. Again, there is no significant difference between the two approaches in terms of the other key performance metrics; the sampling period converges within the first 20 msec. The dynamic sampling rate mechanism in FLCD+DS performs on-line optimization of τ from random values in (0,1] sec to 5.05 ± 1.634 msec. This leads to drastic reductions in fuzzy system evaluations while overall system performance remains virtually the same as in FLCD.

5 Conclusions and Future Works

To reduce computational cost in fuzzy systems in communication networks, a distance based triggering probability, which governs the triggering rate of the fuzzy system, is proposed. When the system has not been triggered, the previous output(s) are used. A dynamic sampling rate probability, based on the transformed Lorentzian function of the exponentially weighted moving average (EWMA) of the triggering probability, is also proposed. When the EWMA is tending to the extremes, the sampling period is updated in two ways. If the EWMA is higher than 0.5, it implies that the system is under-sampled such that the sampling period is decreased probabilistically. Conversely, if the EWMA is lower than 0.5, it implies that the system is over-sampled such that the sampling period is increased probabilistically. A stopping criterion is also proposed to ensure that this mechanism is switched off when the sampling period has converged.

The triggering probability reduces the number of fuzzy computations in the Fuzzy Logic Congestion Detection (FLCD) in WLANs by more than 45%. Performance, in terms of packet loss rate, delay, jitter, and throughput, however, remains virtually the same. On the other hand, the dynamic sampling rate mechanism leads to more than 150% improvement in sampling rate while the number of fuzzy computations in the FLCD mechanisn in WLANs is reduced by more than 70%. Again, performance in the other key metrics remains virtually the same.

This work shows that the triggering probability can help to alleviate redundant computations in fuzzy systems in communication networks while preserving the performance levels. It also shows that the system can optimize the sampling period just by using the distance information. As part of future work, the proposed mechanism will be implemented in fuzzy systems in wireless sensor/actuator networks.

Bibliography

[1] M. Spott, K. Leiviska, and T. Martin: Roadmap Contribution IBA C Applications in Telecommunications, Multimedia and Services, *European Network on Intelligent Technologies (EUNITE) for Smart Adaptive Systems (SAS)*, July 2004.

[2] Y.L. Chen, J.W. Wang, Y.S. Lin, and J.H. Wen: Combined Fuzzy-Based Power Control with Window-Based Transmission Rate Management in Multimedia CDMA Cellular Systems, *International Journal of Electronics and Communications*, doi:10.1016/j.aeue.2010.04.009, 2 June 2010.

[3] C. Chrysostomou, A. Pitsillides, A. Sekercioglu: Fuzzy Explicit Marking: A Unified Congestion Controller for Best-effort and Diff-serv Networks, *Computer Networks Journal (COMNET)*, Vol. 53, No. 5, pp.650-667, 9 April 2009.

[4] C.N. Nyirenda and D.S. Dawoud: Multi-objective Particle Swarm Optimization for Fuzzy Logic Based Active Queue Management, in *Proc. of the IEEE International Conference in Fuzzy Systems*, Vancouver, Canada, pp. 2231-2238, July 2006.

[5] M. Balakrishnan and E. E. Johnson: Fuzzy diffusion analysis: Decision significance and applicable scenarios, *Proc. of IEEE Military Communications Conference*, no. 1, pp. 2175-2181, October 2006.

[6] M. Marin-Perianu and P. J. M.Havinga: D-FLER: A distributed fuzzy logic enginefor rule-based wireless sensor networks, *Proc. of International Symposium on Ubiquitous Computing Systems (UCS)*, pp. 86–101, 2007.

[7] F. Xia, W.H. Zhao, Y.X. Sun, Y.C. Tian: Fuzzy Logic Control Based QoS Management in Wireless Sensor/Actuator Networks, *Sensors*, Vol. 7, No.12, pp. 3179-3191, 2007.

[8] C.N. Nyirenda, F. Dong, and K. Hirota: Euclidean Distance Based Triggering of Fuzzy Systems in Communication Networks, In proceedings of *the International Symposium on Intelligent Systems (iFAN 2010)*,Tokyo, Japan, September 2010.

[9] P. Whittle, *Probability via Expectation*, 4th ed., Springer-Verlag, New York, 2000.

[10] D.A. McQuarrie: *Mathematics for physical chemistry*, pp. 124, Univ. Science Books, 2008.

[11] J.K. Hunter and B. Nachtergaele: *Applied analysis*, World Scientific, 2001.

[12] B. Richmond and T. Richmond, *A Discrete Transition to Advanced Mathematics*, AMS Bookstore, 2009.

[13] C.N. Nyirenda, D.S. Dawoud, F. Dong, M. Negnevitsky, and K. Hirota: A Fuzzy Multiobjective Particle Swarm Optimized TS Fuzzy Logic Congestion Controller for Wireless Local Area Networks, *Journal of Advanced Computational Intelligence and Intelligent Informatics (JACIII)*, Vol.15, No.1, pp. 41-54, January 2011.

[14] *NS2 network simulator*, http://www.isi.edu/nsnam/ns/, Accessed on 28 June, 2010.

[15] E.H. Mamdani and S. Assilian, An experiment in linguistic synthesis with a fuzzy logic controller, *International Journal of Man-Machine Studies*, Vol. 7, No. 1, pp. 1-13, 1975.

[16] T. Takagi and M. Sugeno, Fuzzy identification of systems and its applications to modeling and control, *IEEE Trans. on Systems,Man and Cybernetics*, Vol.15, 116-132, 1985.

[17] R. Pan, B. Prabhakar, and K. Psounis: Choke - a stateless active queue management scheme for approximating fair bandwidth, *Proc. of INFOCOM*, pp. 942-951, March 2000.