International Journal of Advanced Computer Research (ISSN (Print): 2249-7277 ISSN (Online): 2277-7970)
Volume-4 Number-4 Issue-17 December-2014

Hierarchal Scheduling Algorithm for Congestion Traffic Control Using Multi-Agent Systems

Ma'en Saleh[1], Ahmad Aljaafreh [2] and Naeem Al-Oudat [3]*

Received: 20-October-2014; Revised: 30-November-2014; Accepted: 10-December-2014

Abstract

Congestion on roads may lead to a catastrophe, especially for those large urban areas. Accordingly, different intelligent traffic-control methodologies had been implemented based on a variety of technologies such as DSP (Digital Signal Processing), WSNs (Wireless Sensors Networks), Image processing, etc. The design process depends on different factors such as fuel consumption, waiting time, traffic volume, and vehicle density. In this paper, we propose an adaptive traffic light control design based on hierarchal scheduling algorithm (WFQ/FCFS).

The congestion control problem was modeled based on multi-agent systems, where the whole process was decomposed into a set of communicating sub agents. The traffic congestion control is based on minimizing the average total waiting time of vehicles at each lane in a single intersection. Our proposed control system is mainly designed and modeled based on packet switched networking model, where different classes of real-time traffics (video, audio) requesting the best QoS to be guaranteed. Simulation studies shows that the proposed adaptive-weighted-agent-based algorithm (AW Agent Based), the core of control design, outperforms the baseline algorithm as the variance in arrival rates increases.

Keywords

Traffic-control, scheduling, multi-agent, WFQ, FCFS, simulation, QoS.

*Author for correspondence

Ma'en Saleh, Computer and Communication Engineering, Tafila Technical University, Tafila, Jordan.

Ahmad Aljaafreh, Computer and Communication Engineering, Tafila Technical University, Tafila, Jordan.

Naeem Al-Oudat, Computer and Communication Engineering, Tafila Technical University, Tafila, Jordan.

1. Introduction

The earliest types of traffic control mechanisms are fixed-time based approaches, where both cycle time and green time duration are fixed for the traffic lights in an intersection [1]. Such approach shows an efficiency when no changes occur to the traffic status at the intersection. As a result, deploying intelligent traffic control mechanisms becomes a demand, especially for those large and complex traffic networks, where traffic congestion becomes critical. The main idea behind such intelligent mechanisms is to be adaptive and responsive to any change in the traffic status that may affect the traffic's quality-of-service (QoS) guarantees such as vehicles delay time, vehicles fuel consumption, vehicle emissions, etc. [2]. Accordingly, the intelligent traffic control mechanism deploys an on-line (real-time) monitoring system that adopts a feedback strategy for sensing any change in the traffic status, and thus recalculates the traffic control parameters to accommodate such change and provide the traffic network with the best QoS.

One of the methods followed by the traffic control process to provide the traffic flows with guaranteed QoS requirements is by applying the appropriate real-time priority scheduling algorithm such as earliest deadline first (EDF), differentiated earliest deadline first (Diff-EDF), and weighted fair queue (WFQ) [3][4]. The priority key adopted by the real-time scheduling algorithm depends on the type of the service requested, where minimizing the waiting time for the vehicles at each lane at the intersection is the most common priority key.

The modeling and analysis process for the traffic control system depends on the status of the traffic network. Conventional simulation techniques are suitable for those fixed-time based approaches, where the dynamic changes in the network is not taken into consideration while analyzing system performance. For adaptive (responsive) traffic control mechanisms, conventional simulation techniques would be inefficient, where the traffic control process should be aware of different factors such as type of intersection, traffic density, road effects, rush hour, etc. Such problem could be overcome by applying the agent-based simulation techniques, where the whole system is decomposed into a set of cooperative sub-agents with a well-defined communication protocol between them [5].

In this paper, an adaptive and responsive traffic light control design was proposed. The proposed system deploys a hierarchal scheduling algorithm (WFQ/FCFS) to achieve guaranteed QoS requirements for the traffic flows in a single intersection in terms of traffic waiting time. The priority key followed by the scheduling algorithm depends on both traffic's queue length and arrival rate. The whole process is modeled and designed using real-time agent based system. In this paper, our traffic control system is modeled as a packet networking model with different real-time flows that are requesting for appropriate QoS requirements from the service provider [6].

The rest of this paper is organized as follows. Section 2 reviews some work that is closed to the work presented here. Section 3 discusses the design of the proposed system. Agent-based traffic control system is presented in Section 4. In Section 5, simulation experiments setup and simulation results were shown and discussed. Finally paper is concluded and some directions for future work are given in Section 6.

2. Literature Review

An intensive research was performed to design and implement a variety of intelligent traffic control models based on different technologies. In [7], an intelligent traffic control model based on genetic algorithm was proposed to adaptively evaluate the optimal green time in an intersection based on machine learning.

Fuzzy-based strategies were widely deployed for intelligent traffic control systems. In [8], an adaptive neuro-fuzzy inference system was proposed to minimize the waiting time for vehicles in an intersection according to the flow rate for each lane in the intersection. An event driven traffic simulation program was implemented by Matlab to evaluate the performance of such system, which shows efficiency over traditional fixed traffic control strategies. A traffic control model based on fuzzy-neural network was proposed in [9]. In its implementation, the model adopts the queue length at each lane of an intersection to drive the control process, and thus reducing the average delay for the vehicles especially for those harsh environments.

Agent-based systems were integrated to implement traffic control systems. A multi-agent based model for intelligent traffic control was proposed in [1]. The system decomposed the traffic control problem into subagents that exchange environment status at an intersection to reduce the waiting time for vehicles at such intersection. The simulation was implemented using NetLogo-based traffic simulator that models the actions and communication between agents. In [10], a cooperative distributed multi-agent system was integrated with a dynamic route guidance to increase traffic control efficiency. The cooperative system detects the environment status based on a feedback from a WSN and performs machine learning for traffic control decisions.

Adopting scheduling algorithms for traffic control becomes essential especially for those intelligent models, where prioritizing the traffic flows is the core behind solving the road congestion problem. In [11], a real-time intelligent traffic control mechanism based on scheduling algorithms was proposed. The model depends on such scheduling algorithms to determine the moments of times when the traffic light phases have to be started according to the status of lanes at an intersection. An intelligent traffic control strategy based on CPU scheduling processes was designed and modeled to reduce the waiting time of the vehicles in an intersection [12]. The model was implemented based on three main phases: hardware design, information extraction, and design of control algorithm. The proposed algorithm supports those countdown-time traffic lights and enhances the overall process of traffic control for complex environments.

3. System Design

According to the multi-agent, our design was based on three main layers: decomposition, modeling, and communication protocol. Through decomposition, the design was decomposed into five sub-agents: source, queue, server, WFQ/FCFS hierarchal scheduler, and controller. Defining the functionalities of each sub-agent was through the modeling layer, while communication protocol defines the interaction between the sub-agents in terms of control/data message exchange.

3.1 Source Sub-Agent

The source sub-agent is the one that is responsible of generating the traffic at each lane of the intersection. Accordingly, the source sub-agent generates a traffic *y* with a rate (λ_y). An exponential distribution with mean ($1/\lambda_y$) was used to generate the inter-arrival time for each vehicle in the traffic *y*.

3.2 Queue Sub-Agent

This entity is responsible of store/retrieve vehicles. That is the vehicle arrived will be stored in the queue based on its arrival-time, while it retrieves the vehicle from the top of the queue to be served by the server sub-agent.

3.3 Server Sub-Agent

This entity serves the vehicle that was chosen by the scheduler to be served. An exponential function with a mean ($1/\mu_y$) was used to model the service time, where (μ_y) is the vehicle service rate. Different metrics affect such rate such as vehicle size, vehicle location in the lane, and lane dimensions (Bandwidth in terms of networking problems).

3.4 Hierarchal (WFQ/FCFS) Scheduler Sub-Agent

This sub-agent is decomposed into two main layers: outer and inner schedulers. The outer scheduler is the one that is responsible of serving the vehicles per lane according to the first-come-first-served (FCFS) scheduling algorithm, while the inner layer adopts the weighted fair queue (WFQ) scheduling algorithm to control the time slice given for each sub-lane in the intersection.

3.5 Controller Sub-Agent

This sub-agent is the core entity that is responsible of collecting system information from other sub-agents and evaluating system parameters needed to govern the functionalities of all sub-agents in the multi-agent system. Accordingly, it interacts with all entities in the system to ensure proper services.

4. Agent-Based Traffic Control System

The process starts when the source sub-agent at each starts to send its traffic with a well-defined sending rate (λ). Such sending rate will be generated randomly and periodically (every time period T) for each lane in the intersection. When a vehicle arrives to the queue sub-agent, the queue sub-agent performs the process of queuing (storing) the vehicle according to their interarrival times. The queue sub-agent keeps track of the queue length parameter (Q) that is the number of vehicles arrived and stored in the queue. Over every time period T, the controller sub-agent interacts with the queue sub-agent at each lane requesting for its queue-length parameter (Q) by sending control messages. Accordingly, each queue sends its Q parameter through the data communication line defined by the communication protocol. From the other side, the controller interacts with the sub-agent requesting for its traffic rate (λ) through broadcasting control messages over the interaction lines (control lines) defined by the communication protocol. Accordingly each source subagent responds by sending its sending rate (λ) to the controller. The communication protocol between the sub-agents is shown in Fig. 1.

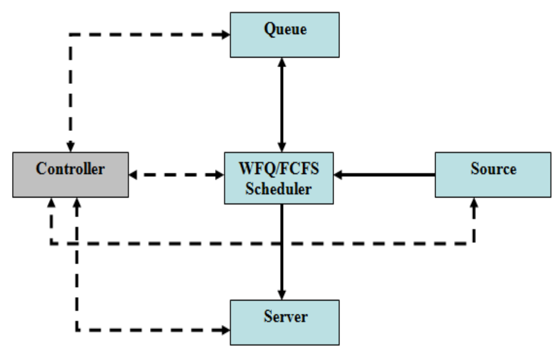

Figure 1: Traffic-Control Multi-Agent System. (Solid line: data transfer; Dashed line: interaction)

Upon receiving such parameters, the controller subagent performs the following:
1) Evaluating the serving weight, w_y, for each y, of the N lanes in the intersection, such that:

$$Q_w = \frac{Q_y}{\sum_{i=1}^{N} Q_i} \quad (1)$$

2) The controller uses the information collected from each source sub-agent regarding the traffic sending rate (λ) to evaluate the optimal cycle time (T_C) for each lane in the intersection. Such process will be performed by optimal cycle-time sub-unit inside the controller sub-agent, which had been modeled and designed using the Simulink as shown in Fig. 2. Once the optimal cycle time generated for each lane, the

controller evaluates the optimal cycle time (T_C) needed for the whole intersection at each time period (T). Let T_y^C denotes the cycle time generated by the traffic rate for lane y and N is the number of lanes in the single intersection. Then

$$T_C = \max_{1 \le y \le N} T_C^y \tag{2}$$

The controller keeps in its data base the optimal cycle time for each different traffic arrival rate in a hash-table form. Such technique will eliminate the overhead generated by the redundancy of evaluating the optimal cycle-time. Accordingly, the controller first checks if there is an entry in its database for each arrival rate before performing the process in the case of no entry. Such process will build the controller's database based on any new updates in the intersection.

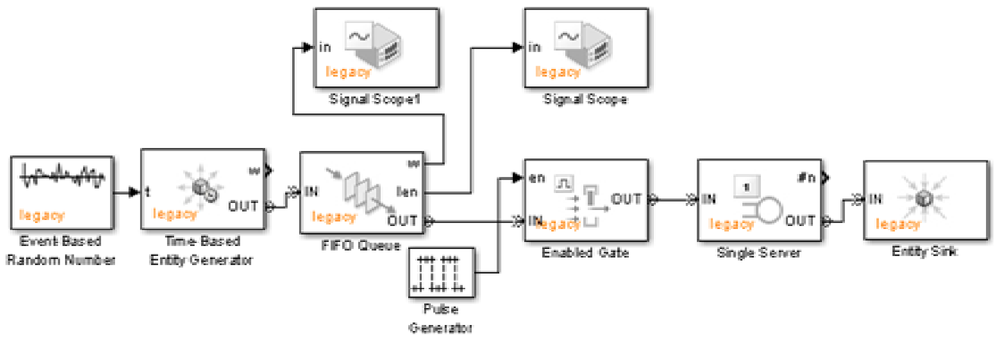

Figure 2: Simulink block diagram for a single approach

3) Once the intersection's optimal cycle time (*Tc*) was evaluated, the controller defines the green time, g_y, for each lane, *y*. Let τ denotes the yellow time needed to switch between two lanes. Then:

$$g_y = w_y(T_c - 4\tau) \tag{3}$$

The controller passes the set of green light periods for the external WFQ scheduler. Such parameter would be the time slice given by the scheduler for each lane to serve its own traffic. Upon receiving such parameter to the outer-layers scheduler, the scheduler determines the sequence, S, to be followed in serving the lanes, {a, b, c, d}. The sequence is evaluated such that the traffic congestion is minimized, where the lane with higher queue length would be served first such that:

$$S = a \rightarrow b \rightarrow c \rightarrow d$$

Where: $g_a > g_b > g_c > g_d$

This implies that: $Q_a > Q_b > Q_c > Q_d$

The outer scheduler then passes the green time (g) for each inner scheduler (FCFS). When the server subagent is idle, it requests the inner scheduler to send it a vehicle to serve. The FCFS scheduler then requests the queue sub-agent to send it a vehicle from the top of the queue. Upon receiving the vehicle to the scheduler, it passes it to the server agent to serve, which in turns change its status to busy. The server agent keeps track of two main parameters: (1) n_y: number of vehicles served during the green time for lane y; (2) t_y: total waiting time of vehicles in lane y till served. t_y is calculated as:

$$t_y = \sum_{i=1}^{n_y} \xi_i + w_i - v_i \tag{4}$$

Where ξ_i is the arrival time of vehicle *i* to the server sub-agent. w_i is the service time of vehicle *i*. v_i is the arrival time of the vehicle *i* to the queue sub-agent. Once the green time was elapsed for lane y, the outer-scheduler switches the turn to the next lane in the sequence *S*. The process of such switching needs a specific amount of time, τ that is the time for yellow light in the traffic light. The flow diagram of the controller and the design model are shown in Fig. 3 and Fig. 4, respectively. The performance of the system is evaluated by the controller through requesting the parameters t_y and n_y; from the server sub-agents. Upon receiving such parameters, it evaluates the average waiting time, t, of the vehicle during the cycle such that:

$$t = \frac{\sum_{i=1}^{N} t_i}{\sum_{i=1}^{N} n_i} \tag{5}$$

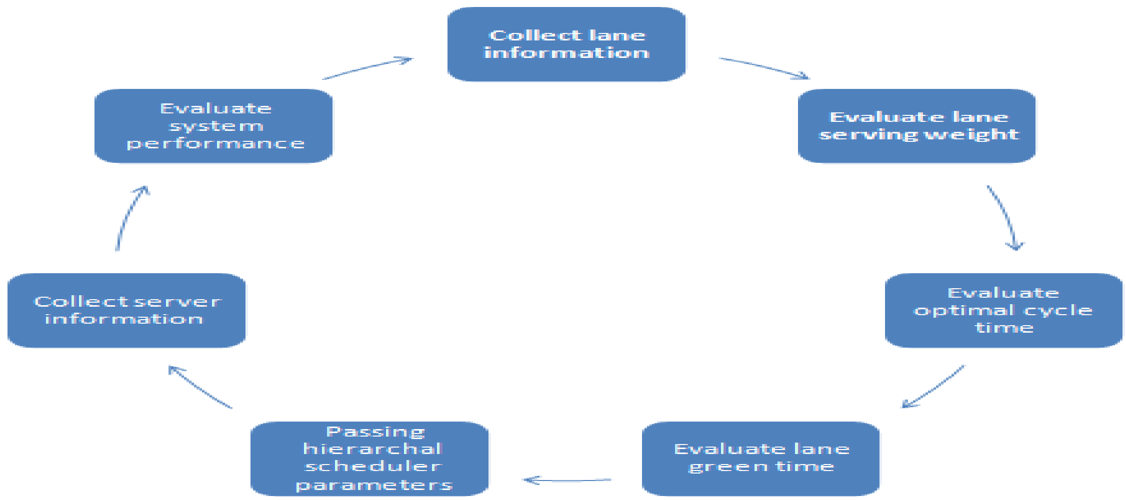

Figure 3: Adaptive controller flow chart

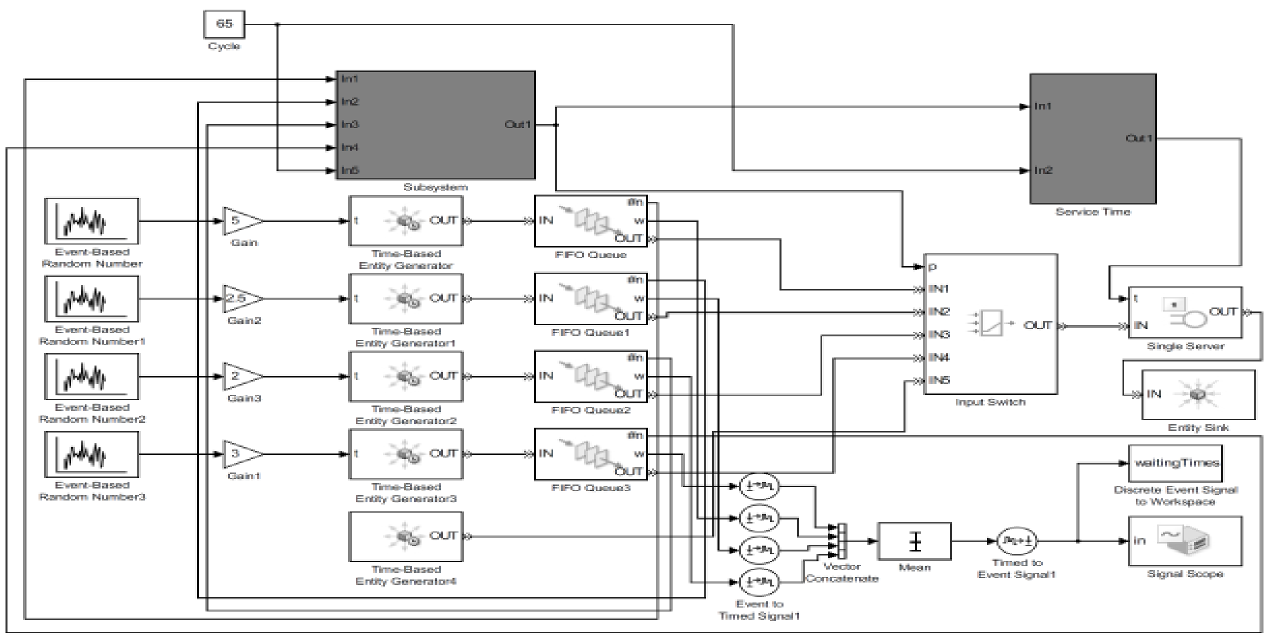

Figure 4: Simulink block diagram for an agent based approach

5. Simulation Results

In the following 51 runs with different seeds for Event-Based random number generator are used. For each point in the figures all 51 runs were conducted, then the average is calculated. Figures also show 98% confidence intervals around each point. Note that the yellow interval is kept at 4 seconds in all the simulation experiments. Fig. 5 shows the fixed-intervals-and-fixed-sequence algorithm (*AW Fixed*) as compared to the Adaptive-intervals-and-sequence algorithm. *AW Fixed* outperforms the *AW Agent based* since the fixed algorithm can use the optimal green intervals that is calculated in advance while the proposed algorithm is adaptively respond the arrival rate variation by changing green intervals and sequence of these intervals. This is not as good as using the optimal calculated intervals in *AW Fixed*.

Fig. 6 shows average waiting time vs. arrival rates variance-increase where the cycle for the *AW Fixed* is kept at 65 seconds. This figure shows the power of the proposed algorithm when the variance of arrival rate increases. *AW Agent based* outperforms *AW Fixed* as variance increases.

6. Conclusions and Future Work

In this research, an adaptive intelligent traffic control system was proposed to avoid the congestion on roads through minimizing the vehicles delay time at a single intersection. Three main contributions were highlighted in our research: (1) deploying agent-based system to model the system; (2) applying scheduling algorithms to serve the real-time traffic, and thus presenting the problem in terms of scheduling CPU processes within a number of shared resources; (3) modeling the problem as a packet networking model through showing the analogy between the two models and (4) Simulate our *AW Agent based* algorithm using the Simulink graphical programming tool. According to an extensive simulation experiments, *AW Agent based* outperforms the baseline algorithm (*AW Fixed*) as the variance of the arrival time increases.

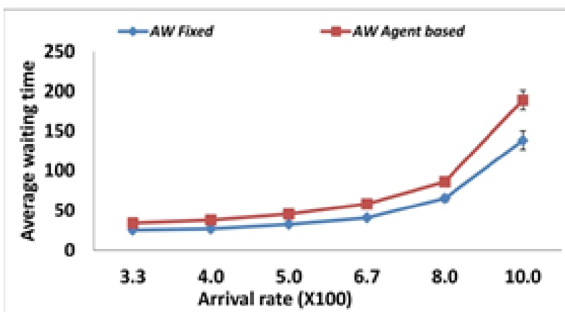

Figure 5: Average waiting time vs. arrival rates

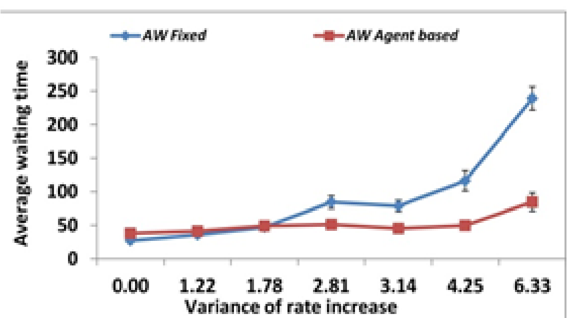

Figure 6: Average waiting time vs. variance-increase in arrival rates

As a future work, the definition of the agent based system might be changed to consider each intersection in a specific area as an agent with a layer of cooperation between them to solve the congestion problem in such area rather than solving it at the intersection area only.

References

[1] V. Hirankitti and J. Krohkaew, An agent approach for intelligent traffic-light control, First Asia International Conference on Modelling & Simulation (AMS '07), pp. 496-501, Mar. 2007.
[2] B. Zhou, J. Cao, X. Zeng, and H. Wu, Adaptive Traffic Light Control in Wireless Sensor Network-Based Intelligent Transportation System, IEEE 72nd Vehicular Technology Conference Fall (VTC 2010-Fall), pp. 1-5, Sept. 2010.
[3] A. Gupta, D. Ghosh, and P. Mohapatra, Scheduling prioritized services in multihop OFDMA networks, IEEE/ACM Trans. Netw., vol. 18, no. 6, pp. 17801792, Dec. 2010.
[4] J. Zhou, X. Zhou, Z. Chen, and X. Chen, Research of real-time control algorithm for traffic lights based on CPU process scheduling, IEEE International Conference on Anti-Counterfeiting, Security and Identification (ASID), pp. 110-114, Jun. 2011.
[5] R. Olfati-Saber, J. A. Fax, and R. M. Murray, Consensus and cooperation in networked multi-agent systems, Proc. IEEE, vol. 95, no. 1, pp.215233, Jan. 2007.
[6] M. Saleh and L. Dong, Real-time scheduling with security enhancement for packet switched networks, IEEE Trans. On Network and Service Management, vol. 10, no. 3, pp. 271 285, Sep. 2013.
[7] L Singh, S. Tripathi, and H. Arora, Time optimization for traffic signal control using genetic algorithm, International Journal of Recent Trends in Engineering, vol. 2, no. 2, Nov. 2009.
[8] C. T. Wannige and D. U. J. Sonnadara, Traffic signal control based on adaptive neuro-fuzzy inference, 4th International Conference on Information and Automation for Sustainability, pp. 301-306, Dec. 2008.
[9] D. Jia and Z. Chen, Traffic signal control optimization based on fuzzy neural network, IEEE International Conference on Measurement, Information and Control (MIC), vol. 2, pp. 1015-1018, May 2012.
[10] X. Chen, Z. Yang, and H. Wang, A multi-agent urban traffic control system cooperated with dynamic route guidance, IEEE International Conference on Machine Learning and Cybernetics, pp. 330-335, Aug. 2006.

[11] T. S. Letia, real-time approach of urban vehicle traffic control, IEEE International Conference on Automation, Quality and Testing, Robotics, vol. 1, pp. 346-350, May 2006.

[12] J. Zhou, X. Zhou, Z. Chen, and X. Chen, Research of real-time control algorithm for traffic lights based on CPU process scheduling, IEEE International Conference on Anti-Counterfeiting, Security and Identification (ASID), pp. 110-114, Jun. 2011.

Maen Saleh (M'10) received his B.S. and M.S. degrees in computer engineering from Yarmouk University in 2005 and 2007, respectively. He received his Ph.D. degree in electrical and computer engineering from Western Michigan University in 2012.
From 2006 to 2007, he was a Lecturer at the Computer Science Department at Yarmouk University. From 2007 to 2008, he was a Lecturer at the Electrical and Computer Engineering Department at Tafila Technical University. From 2010 to 2012, he was a Teaching Assistant at the Electrical and Computer Engineering Department at Western Michigan University. In 2012, he joined the faculty of Tafila Technical University as an Assistant Professor of Electrical and Computer Engineering. Dr. Maen research interests include real-time scheduling for packet switched networks, security-aware scheduling, real-time agent-based systems, and QoS for heterogeneous networks.

Ahmad Aljaafreh received his PhD in Computer Engineering in 2010 from Western Michigan University. In the same year, he joined Tafila Technical University in Tafila, Jordan. Currently, he is an assistant Professor at the Department of Communications, Electronics and Computer Engineering, Faculty of Engineering at Tafila Technical University. His research interests are in, embedded systems and machine learning.

Naeem Al-Oudat received his BSc in Telecommunication engineering from Muta University in 1992, his MSc in Computer engineering from Yarmouk University in 2005 and his PhD in Computer engineering form Iowa state university in 2012. Since 2012, He is serving as a faculty member in Communications and Computer engineering department in Tafila Technical University. His research interests include real-time distributed systems task optimizing and scheduling, educational operating systems design.
Email: naeemodat@ttu.edu.jo.