IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-6, Issue-9, Sept. 2018 E-ISSN: 2347-2693

Analysis of Scheduling Performance and Stability in Wireless Ad-Hoc Networks Using ACO, MWS and Novel ACO-MWS

B. Sindhupiriyaa[1*], D. Maruthanayagam[2]

[1]Sri Vijay Vidyalaya College of Arts & Science, Dharmapuri, Tamilnadu.India
[2]Head cum Professor, PG and Research Department of Computer Science,
Sri Vijay Vidyalaya College of Arts & Science, Dharmapuri,Tamilnadu,India.

[*]*Corresponding Author: sindhutbharathi@gmail.com*

Available online at: www.ijcseonline.org

Accepted: 19/Sept./2018, Published: 30/Sept./2018

Abstract- Multi-hop wireless networks and Routing management, it has is a vital and challenging resource allocation technique in Scheduling process. A distributed low-complexity scheduling algorithm develops into more challenging tasks, even if taking into account a physical interference model. At previous years a number of scheduling algorithms were presented, but pre-existing scheduling algorithm does not solve the drawbacks to implement in multi hop networks to overcome these issues, **proposed a new scheduling approach Ant Colony Optimization Max-Weight Scheduling (ACO-MWS)** for scheduling and routing in multi-hop wireless networks. The combination of proposed approaches such as ACO Algorithm and Max-Weight Scheduling that overcome the pre-existing scheduling problems and it accomplish maximum throughput at the distributed low complexity. The performance evolution of ACO, Max-Weight, and ACOMWS are presented in this paper.

Keywords- Wireless Ad Hoc Network, Heavy Tail, Ant Colony Optimization, Max-Weight Scheduling, Throughput, Routing

I. INTRODUCTION

In Multi-hop wireless networks, it has significant interest to surveying the scheduling issue for performing throughput/capacity optimization in wireless networks [1]–[5]. In the main purpose, this scheduling issue involves to determine that node-pairs (links) should communicate and transmit and at what modulation, execution times and which coding systems must be utilized, and at which power levels should communication take place. Even as the optimal solution of this scheduling issue has been recognized for a long period [6], so that the resultant solution has high computational complexity and it is complex to perform within multi-hop networks. One of the key considerations in a new scheduling policy intended for a queuing network is throughput optimality, which is the capability to maintain the largest set of traffic rates by a given queuing network. Max-weight scheduling [7] and Backpressure [8] are recognized to be throughput optimal by using queue length based scheduling policies. Therefore, the max-weight of scheduling policies receives much consideration in different networking circumstances, comprises satellites [10], switches [9], optical networks [12] and wireless [11]. The familiar maximum weight scheduling algorithm proves throughput optimality proposed by Tassiulas in this paper [7].

In wireless network performance, the single-hop interference model also called primary interference or the node-exclusive model. If two links or node pairs interfere with each other only it is performed within single-hop distance. The throughput-optimal policy [13] signify a Maximum Weighted Matching (MWM) policy and its complexity is approximately O (N3) [14], where N is considered the entire number of links in the wireless networks. At the same time as the single-hop interference model is used as a reasonable approximation to Frequency Hoping Code Division Multiple Access (FHCDMA) networks or Bluetooth device ([13], [16], [15],). A large class of systems knows how to be modeled using the more general K-hop interference models, wherein any two links within K-hop distance does not be activated concurrently. For instance, the pervasive IEEE 802.11 Distributed Coordination Function (DCF) of wireless networks are utilized the two-hop interference model to be effectively performed [18], [19]. The complexity of the throughput-optimal policy of [17] intended for the K-hop interference model is NP-Hard problem [15], and for this reason, it is complicated to implement. Furthermore, traditional scheduling policies are developed under LT (light-tailed) traffic assumptions. But, traffic flows of heavy tailed (HT) emerges in various network systems in recent empirical studies.

In this paper, to improve new throughput-optimal scheduling techniques perform with LT and hybrid HT traffic flows in multi-hop networks, in which classic optimal policies such as backpressure/ maximum-weight schemes, enhanced under LT assumption, are not throughput-optimal solution to any further extent. The notion of delay stability utilizes by the performance metric, check whether the steady-state expected delay is finite or not in a queue. Max-Weight policies perform with excellent stability properties, and as well to complete better delay performance under LT traffic flow. The Max-Weight policy achieves defectively using heavy tails, at the same time as a correctly modified version of Max-Weight accomplishes much better overall performance. Intended for wireless networks through multi hop traffic flows, a tight backlog bound scales as O (N / (1−ρ)) in which N carries the number of wireless nodes. A novel delay metric used for multi hop wireless networks and improve the D-BP algorithm, where a linear relation between queue delays and lengths in the fluid limits know how to be established. D-BP achieves optimal throughput performance of this linear relation. The throughput-optimality of Q-BP uses fluid limit techniques. In advance, to develop a simpler ACO approximation of D-BP performs for practical implementation.

1.1 PRELIMINARIES

- **Heavy Tail (HT):** In heavy-tailed traffic, a non-negative random variable X, even if all θ> 0, $\lim_{x\to\infty} e^{\theta x}$ Pr(X > x) = ∞, or equivalently, E[e^{zX}] = ∞, ∀z > 0. A r.v. is light-tailed (LT) traffic, even if it is not HT, or equivalently, if there exists z > 0 so that E[e^{zX}] < ∞.
- **Regulary varying distribution:** A random variable X is known regularly varying with tail index $\beta > 0$, denoted by $X \in RV(\beta)$, if Pr(X>x)$\sim x^{-\beta}$ L(x), if two real functions a(t) and b(t), a(t) ∼ b(t) denote $\lim_{t\to\infty} \frac{a(t)}{b(t)} = 1$ and L(x) is a gradually varying function.
- **Steady-state Stability:** In queuing system, if pre-exists a scheduling policy under the Markov chain of queue lengths is positive Harris recurrent (i.e., {Q (t); t ∈ Z_+} converges in distribution), after that the queuing network is steady-state stable position.
- **Strong Stability:** A queuing system is strong fully stable, when all traffic flows experience bounded average queuing delay (i.e., E [W_f] < ∞, ∀f ∈ F).
- **Network Capacity Region:** In network capacity region Φ of the queuing system is the set of all traffic admissible rate vectors by the system (i.e., λ know how to be covered by a convex combination of feasible schedules). Mathematically, Φ := { λ ∈ R_+^F | $\lambda \leq \sigma$ component wise, for some σ ∈ Co(S)}, in which Co(S) indicates the convex hull of all feasible schedules.
- **Throughput Optimality:** The throughput-optimal is one of a scheduling policy, if it can accomplish strong stability for any admissible rate vector (i.e., any rates within the network capacity region).

II. MAX WEIGHT SCHEDULING IN MULTI HOP FLOWS

In wireless ad hoc network, as directed graph represent in $G = (V, E)$, where E denotes the set of links and V denotes a set of wireless nodes. The cardinalities of V and E are N and L, respectively. Assume that, is set of multi hop flows F where flow $f \in F$ has a fixed route from a source node s (f) to a destination node d (f). The set of links and nodes signify on the route of flow f as L (f) and R (f), in that order. The packet arrivals to source nodes of all flows are i.i.d stochastic mechanism.

The queue length of flow f represent at node n at the starting of time slot t as $Q_n^f\ (t)$and the number of packets arriving at the source node of flow f as $A_{s(f)}^f\ (t)$. The data packets of any flow are delivered to the higher layer upon reaching the destination node, so $Q_{d(f)}^f\ (t)$=0. In addition, let $\mu_n^f\ (t)$ be the number of packets of flow f transmitted from node n along link (n, m) of its route which is buffered at node m if $m \neq d$ (f). Again, we assume that $\mu_n^f\ (t)$ = 1 if we activate link (n, m) on the route of flow f and $\mu_n^f\ (t)$ = 0, otherwise. Given the routes for all flows, the maximum weight scheduling algorithm is used for data delivery [20]. The maximum weight scheduling algorithm is to attain the capacity region [20]. A feasible schedule with the maximum weight at any time slot which is activated by the Max-Weight policy. The Max-Weight policy, the scheduling vector S (t) belongs to the set:

$$S(t)\ \epsilon \arg max_{(s_f\ )\epsilon S}\ \{\textstyle\sum_{f=1}^{F} Q_l(t).\ \mu_l\ (t)\}.. \quad ....(1)$$

Specifically, the scheduling is performed in every time slot as follows:

Each link *(n, m)* discovers with the intention of maximum differential backlogs as follows:

$$w_{nm}(t) = \ max_{f:(n,m)\in L(f)}\{\ Q_n^f\ (t) - \ Q_m^f\ (t)\ \}\ \ ...(2)$$

Depending upon calculated link weights, a maximum weight schedule is found as

$$\underset{\mu}{\rightarrow}{*}\ (t) = \ \text{argmax}_{\vec{\mu}(t)\in S} \textstyle\sum_{(n,m)} w_{nm}(t)\ \mu_{nm}(t) \quad ...(3)$$

One packet transmits from buffer of the flow attaining the maximum differential backlog. The queue progresses written as,

$$Q_n^f(t+1) = \ Q_n^f(t) - \ \mu_n^f(t) + \pi_n^f(t) \quad ....(4)$$

This equation holds because $\pi_n^f(t) = 1$ only if $Q_n^f(t) \geq 1$. It is also considered, $\pi_n^f(t)$ is the number of packets arriving to queue $Q_n^f(t)$ in time slot t that can be written as,

$$\pi_n^f(t) = \begin{cases} A_n^f(t), if\ n = s(f) \\ \mu_{n-1}^f(t), otherwise \end{cases} \quad ……(5)$$

2.1 Throughput Optimality of MWS

In scheduling algorithm, it is throughput-optimal under LT and hybrid HT traffic, even if it known how to achieve moment stability for any admissible rate vectors (i.e., the network capacity region) [21]. The throughput optimality of MWS is proved in this section. The queuing evolution evolves as an irreducible Discrete-Time Markov Chain (DTMC) with infinite countable states. The throughput property of MWS is using Lyapunov drift technique. Let $\vec{q}$(t) = $[q_1(t), q_2(t), \dots, q_L(t)]$, The Lyapunov function is designed as

$$V(\vec{q}(t)) = \sum_{l}^{L} \frac{L}{P_l(1-\rho)} q_l^2(t),$$

Where $\vec{q}$ = $[P_1, P_2, \dots., P_L]$ is picked from Ψ. Theorem indicates the throughput optimality of MWS that is the cornerstone for achieving delay upper bound.

Theorem: The MWS algorithm strongly stabilizes the system for any load vector $\vec{\lambda} \in \Omega$.

Proof: Using the lyapunov function defined in above equation, the drift $\Delta(V(t)) = E[V(\vec{q}(t+1)) - V(\vec{q}(t))|\vec{q}(t)]$ can be calculated by

$$\Delta(V(t)) = 2\sum_{l=1}^{L} \frac{L}{P_l(1-\rho)} E\left[\left(A_l(t) - \varphi_l'(t)\right).q_l(t)\middle|\vec{q}(t)\right] + E\left[f(t)|\vec{q}(t)\right],$$

Where $f(t) = \sum_{l=1}^{L} \frac{L}{P_l(1-\rho)}$ $[A_l(t) - \varphi_l'(t))^2]$. since $\vec{\lambda} \in \Omega$ based on above equation there exists a positive constant ε, using negative constant we have negative drift as follows

$$\Delta(V(t)) < -2\varepsilon \sum_{l=1}^{L} \frac{L}{P_l(1-\rho)} \varphi_l'(t) q_l(t) + E\left[f(t)|\vec{q}(t)\right],$$

On account of the bounded second moments of the arrival process, above final equation proves stability [6]. Additionally, it verifies can be easily from above equation that the DTMC describing the queuing system is positive recurrent and ergodic.

III. ACO (ANT COLONY OPTIMIZATION)

In ACO is considered an iterative algorithm, at each iteration, after those artificial ants are created to construct solutions from node to node on the network with the constraint not visiting any node [22]. Moreover, a certain amount of pheromone deposited by ants on the links that they traverse. The amount of pheromone $\Delta\tau$ deposited on the quality of the path found. An ant chooses to be visited the next node according to a stochastic mechanism. At each step of the solution construction, by using the pheromone to construct the quality of the results at the end of iteration. The pheromone values are updated with the intention of bias ants in prospect iterations to build solutions close to the better ones before constructed.

In ACO approach, each ant tries to discover a path in the network; it is provided that has minimum cost. Ants were initiated from a source node *s* to destination node via neighbor repeater nodes represent r_i, to reach at a final destination node represent as *d*. Even if a source has to be transferred data to the target node that is defined as base station or a base, launching of the ants is performed. After that launching, the option of the next node *r* is completed in accordance through a probabilistic decision rule:

$$P_k(r,s) = \begin{cases} \frac{[\tau(r,s)]^{\alpha}.[\eta(r,s)]^{\beta}}{\sum_{r\epsilon R_s}[\tau(r,s)]^{\alpha}.[\eta(r,s)]^{\beta}} & \text{if } k \notin tabu^r \\ 0 & \text{otherwise} \end{cases} \quad ….(6)$$

Where R_s is the receiver nodes and η (*r*, *s*) is the heuristic value, τ (*r*, *s*) is the pheromone value related to energy. Intended for node *r*, $tabu^r$ is the list of identities of received data packages formerly. Two parameters are β and α to control the relative weight of the heuristic value and pheromone trail. Arcs are connected with the support of Pheromone trails. Each *arc*(*r*, *s*) has a trail value τ (*r*, *s*) $\epsilon[0,1]$. In view of the fact that the destination *d* is a stable base station, the final node of the path is the same for each ant travel. The heuristic value of the node *r* is expressed by equation:

$$\eta(r,s) = \frac{(I - e_r)^{-1}}{\sum_{n\in R_s}(I - e_n)^{-1}}$$

Where *I* denote the initial energy of the source node, and e_r represent the current energy level of receiver node *r*. Nodes notify their neighbors about their energy levels if they sense any modification in their energy levels. In traditional ACO, a special memory M_k is held in the memory of an ant to retain the places visited by that ant (which represent nodes in WSNs). In equation (6), the identities of ants (as sequence numbers) that visited the node previously, are kept in the node's memories, instead of keeping node identities in ant's memories, so there is no necessitate to carry M_k lists in packets during transmission. This approach reduces the size of the data to be transmitted and saves energy. Each receiver node decides whether to accept the upcoming packet of ant *k* or not, by checking its tabu list in equation (6). So, the receiver node *r* has a choice about completing the receiving process by listening and buffering the entire packet. If the receiver node has received the packet earlier, it informs the transmitter node by issuing an ignore message, and switches itself to idle mode until a new packet arrives.

After all ants have completed their tour, each ant k deposits a quantity of pheromone $\Delta\tau^k$(t) given by equation (7), where $J_w^k(t)$is the length of tour $w^k(t)$, which is done by ant k at iteration t. The amount of pheromone at each connection (l(r, s)) of the nodes is given by equation. In WSNs, $J_w^k(t)$ represents the total number of nodes visited by ant k of tour w at iteration t:

$$\Delta\tau^k(t) = 1/J_w^k(t)$$

$$\tau(r,s)(t) \leftarrow \tau(r,s)(t) + \Delta\tau(r,s)(t), \qquad \forall l(r,s) \in w^k(t),\ k=1,\ldots,m \qquad \ldots\ldots\ldots(7)$$

In pheromone values are stored in a node's memory in an effective manner. Each node has information about the amount of pheromone on the paths to their neighbor nodes. After each tour, an amount of pheromone trail $\Delta\tau^k$ is added to the path visited by ant k. This amount is the same for each arc(r, s) visited on this path. This task is performed by transmitting ant k back to its source node from the base along the same path, at the same time as transferring an acknowledgement signal for the associated data package. Increasing pheromone amounts on the paths according to lengths of tours, $J_w^{(t)}$, would constantly cause an increasing positive feedback. In order to control the operation, the operation of pheromone evaporation after the tour is also accomplished by equation. A control coefficient $\rho \in (0, 1)$ is used to determine the weight of evaporation for each tour [19]:

$$\tau_{ij}(t) \leftarrow (1-\rho)\tau_{ij}(t) \qquad \ldots\ldots\ldots.(8)$$

IV. ACO-MWS ALGORITHM

In multihop networks, the MaxWeight type algorithms do stabilize the system and have better buffer-usage performance than the other algorithm. The MaxWeight algorithm assigns a weight of (queue-length X channel-rate), and schedules a collection of links that maximizes the total weight (max-weight independent set). Given a rate vector ⋋ interior to the capacity region Λ, a stationary, randomized, queue-independent policy could in principle be designed to stabilize the system, although this would require full knowledge of the traffic rates and channel state probabilities. However, it is well known that the following queue-aware *max-weight* policy stabilizes the system whenever the rate vector is interior to Λ, without requiring knowledge of the traffic rates or channel statistics: Each slot t, observe current queue backlogs and channel states $Q_i(t)$ and $S_i(t)$ for each link i, and choose to serve the link $i^*(t) \in \{1, \ldots, N\}$ with the largest $Q_i(t)S_i(t)$ product. This is also called the *Longest Connected Queue* policy (LCQ) [2], as it serves the queue with the largest backlog among all that are currently ON. **In this paper in section II already explain the** Max-Weight policy, the scheduling vector S (t) belongs to the set:

$$S(t)\ \epsilon \arg max_{(s_f)\epsilon S}\{\sum_{f=1}^{F} Q_l(t).\ \mu_l(t)\}.$$

For multihop networks, for arrival rate vectors strictly in the interior the stability region of the system that satisfy some additional constraints, if the system scale is large enough, the algorithm keeps the system stable. Max-weight policy requires more statistical knowledge to implement. In the following **the proposed work that aims to combine Max-Weight with ACO algorithms** that decide how much data is to be injected into the network. The aim of this research is to ***maximize the total utility*** of traffic injected into the network**,** and ***obtains higher throughput optimality*** then ***minimizes the delay performance*** in multihop heavy tailed networks.

The ant colony optimization algorithm has been successfully applied to many optimization combinatorial problems. The ant foraging process is very similar to the routing problem of ad hoc networks. The ant colony algorithm can be used in ad hoc networks through the pheromone mechanism; the ants search for and maintain optimal scheduling. The mechanism of evaporation updates the pheromone of each node, which can quickly adapt to the needs of the dynamic changes of ad hoc networks. ACO approach in wireless networks use adaptive learning of routing tables. Each node k in the network stores some data structures within itself which are responsible for keeping local traffic statistics, and routing table. Local traffic statistics defines a simple parametric statistical model for traffic distribution over the network as seen by node k. In fact, it keeps track of the amount of traffic flows towards each possible destination. Routing table, for each possible destination d and for each node n, stores a probability value P_{nd} which expresses the desirability of selecting n as the next node when the destination node is d. In fact it shows amount of pheromone deposited on the link (k, n). When an ant at node k heads toward a destination node d, it selects the next neighbor node n with the probability P'_{nd} where we have

$$P'_{nd} = \frac{P_{nd}+\alpha\times l_n}{1+\alpha\times(|N_k|-1)} \quad \text{where } l_n = 1 - \frac{q_n}{\sum_{n'=1}^{|N_k|} q'_n} \qquad \ldots\ldots(9)$$

Where $|N_k|$ is the number of the neighbors of node k, q_n is the length of the queue associated with the link connecting k to n and α is the weight of the importance of the heuristic function with respect to the pheromone deposit. When an ant reaches the destination node, it can then evaluate the goodness of the path. The goodness of the path can be defined according to an application's requirement.

Algorithm:

The network topology model is the wireless graph, (V, E), where V is a network node and E is the link between two nodes. At time t, there are (t) ants. The total number of ants in the network is $m = \sum_{i=1}^{n} a_i(t)$; $P_{ij}(t)$ is the probability of choosing link E_{ij} for ant K at time t.

$$P_{ij}^{k}(t) = \begin{cases} \frac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{j \epsilon allowed_k}[\tau_{ij}(t)]^{\alpha}[\eta_{ij}]^{\beta}} & , j \epsilon\ allowed_k \\ 0 & \text{else} \end{cases} \quad \ldots\ldots(10)$$

where $\tau_{ij}(t)$ is the strength of the pheromone in the link E_{ij}; α is a parameter to measure the trajectory of pheromones; η_{ij} is visibility between node i and node j, which is generally defined as $1/d_{ij}$ (d_{ij} is the distance between node i and node j); β is a parameter that measures visibility; and allowed$_k$ is a collection of nodes that have not been visited. The pheromone update formula on each path in ad hoc networks is as follows:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij} \quad \ldots(11)$$

where ρ is the pheromone volatilization coefficient, which is a constant between 0 and 1, and $\Delta\tau ij$ is the increment of the pheromone of ants passing through links i and j.

$$\Delta\tau_{ij} = \sum_{k=1}^{l} \Delta\tau_{ij}^{k}. \quad \ldots\ldots(12)$$

In ad hoc networks, there are two main reasons for path breaking. One is the movement of nodes on the communication path, and the other is the nodes withdrawing from the network because of energy depletion. Thus, we select relatively reliable nodes and links. Then the path stability (PS) factor is introduced to judge the stability of the path. The max-weight policy is very important because of its simplicity and its general stability properties.

Stability Region: An arrival rate vector $\lambda = (\lambda_1, \ldots, \lambda_F)$ is in the stability region Λ of the multi-hop switched queueing network described above if there pre-existing $\zeta_{f,i,j} \geq 0$, $f \in F$, $i, j \in N$ such that the following set of constraints is satisfied:

- Flow efficiency constraints

$$\zeta_{f,i,j} = \zeta_{f,i,s_f} = \zeta_{f,d_f,i} = 0, \qquad \forall_i \in N, \qquad \forall_f \in F;$$

- Routing constraints

$$\zeta_{f,i,j} = 0, \qquad \forall (i,j) \notin L_f, \qquad \forall_f \in F;$$

- Flow conservation constraints

$$\sum_{j \in N} \zeta_{f,i,j} + \lambda_f . 1_{\{i=s_f\}} = \sum_{j \in N} \zeta_{f,i,j}, \quad \forall_i \neq d_f, \qquad \forall_f \in F;$$

- Link capacity constraints

$$\sum_{j \in N} \zeta_{f,i,j} < 1, \qquad \forall (i,j) \in L.$$

If an arrival rate vector is in the stability region, after that pre-exists a policy that stabilizes the network, in the sense of stability Definition. The stability region depends on the routing constraints, the link capacities and the network topology; however, it does not on higher order statistics of the arriving traffic. In single-hop networks, the two metrics are equivalent progress: a traffic flow is delay stable if and only if the queue buffering the traffic of that flow is delay stable. However, in multi-hop networks the situation could be more difficult. For instance, a traffic flow can be delay unstable at the same time as some queues of that flow are delay stable.

Lemma 1: The multi-hop switched queuing network described above under a stabilizing policy. If queue *(f, i)* is delay stable, for all $i \in N_f$, then traffic flow $f \in F$ is delay stable.

Lemma 2: Let f be a traffic flow with fixed routing. If queue *(f, i)* is delay unstable, for some $i \in N_f$, then traffic flow f is delay unstable.

V. SIMULATION RESULTS

In simulation result, to analyze the performance of this work, the event-driven network simulator is performed (NS2 version 2.34). Evaluate the performance of all the approaches in a larger grid network topology with 40 links and 25 nodes as shown in Figure 1, even if links and nodes are represented by circles and lines, respectively, with link capacity. The capacity of each link has beside the link and carefully assigned to avoid traffic symmetry, to establish 9 multi hop flows are represented by arrows.

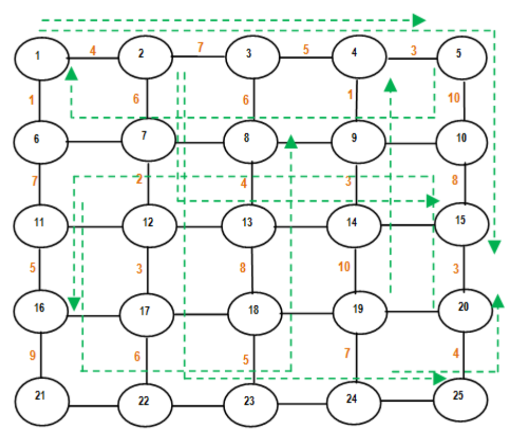

Figure 1: Grid network topology consists of 25 nodes with 40 links

Let consider, uniform traffic in which each flow has independent packet arrivals at each time-slot following Poisson distribution with the same mean rate $\lambda > 0$. to select $\varepsilon = 0.07$ for ACO, MWS and ACOMWS. Each scheduling approach along with the ACOWMS, ***we measure average packet delays under different offered loads to analyze their performance limits.***

Average Queue Length

Average queue lengths under different offered loads to examine the performance limits of scheduling schemes in Figure 2. In simulation result, it represents an average of 10 simulation runs with independent stochastic arrivals, each run lasts for 10^6 time slots. Because the optimal throughput region is defined as the set of arrival rates under which the queue lengths remain finite, to consider the traffic load, under which the queue length increases speedily, as the boundary of the optimal throughput region. ACOWMS and ACO accomplish the same throughput region as MWS, therefore supporting the theoretical results on throughput performance in Figure 2.

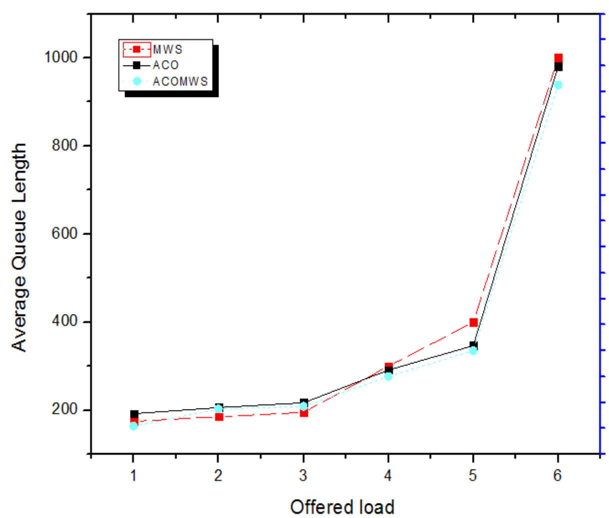

Figure 2: Performance (Average Queue Length) of scheduling algorithms for multihop traffic.

Figure 2 illustrates **ACOWMS** Average Queue Length minimizes compare **MWS** and slightly differ from **ACO.**

Average packet delay

The ***ACOMWS approach have lower packet delays to compare than the ACO and WMS algorithms*** even if traffic load is light (e.g., ⋋< 0.) as shown in Figure 3. Since the scheduling decisions depends upon the shadow queue lengths rather than the actual queue lengths, queues with very small (or even zero) queue length can be activated. On the other hand, the effect tends to reduce with heavier traffic load because the queue lengths are likely to be large. The results also illustrate that the proposed scheme consistently outperform the WMS, ACO algorithms when ⋋ > 0.15. Note that with $\varepsilon = 0.07$, the shadow traffic rate vector is outside the optimal throughput region when ⋋ > 0.15/ (1+ 0.07) ≈0.45, however, interestingly, the schedules chosen based on the shadow queue lengths can stabilize the data queues even if (which is still feasible).

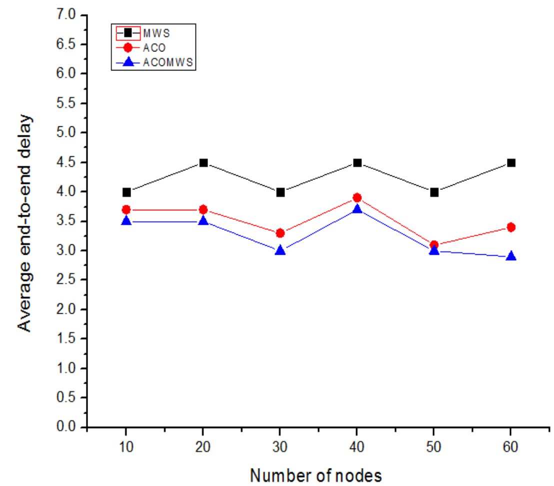

Figure 3: Performance (Average Packet Delay) of all the scheduling schemes in a grid network.

End-to-end delay

The total latency handles between the source and destination experienced by a legitimate packet is provided by end-to-end delay performance to evaluate the time periods experienced as processing, queuing, transmission, propagation and packet delays. The ***ACOWMS simulation results in 10% less delay than the ACO model and 20% less delay than the WMS model in Figure 4.*** The end-to-end delay slowly reduces if the number of nodes increases. Due to the 25 -nodes scenario the nodes are spread over a 750 *X* 750 m^2 area and is a possibility of increase in distance between adjacent nodes. Even if the network size is high scaling, more adjacent nodes are available to perform as intermediate nodes. If the size increases beyond 100, a chance of more packets drops because of collision.

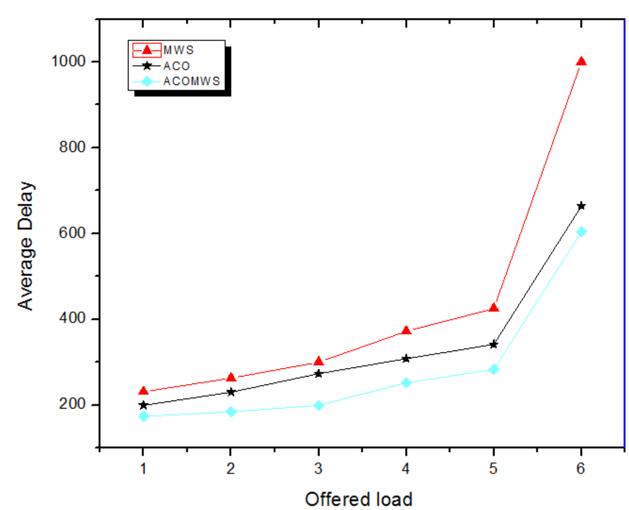

Figure 4: Number of nodes vs. average end-to-end delay (s).

Delivery Ratio and Bandwidth

Graphs vary number of nodes with packet delivery ratio and bandwidth utilization, in Figure 5 and 6, respectively. The ***proposed ACOMWS has staged developments if compared to the other two models.*** For that reason the improved bandwidth utilization can be that the other two models were not taking the bandwidth into consideration while computing the route. Even if the number of nodes is 75 we can see the maximum utilization, and it could slightly reduce after that.

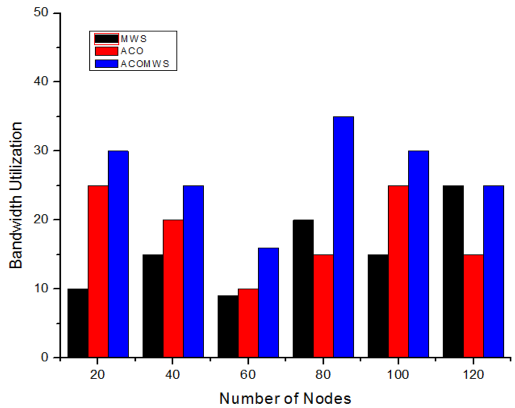

Figure 5: Number of nodes vs. Packet delivery ratio

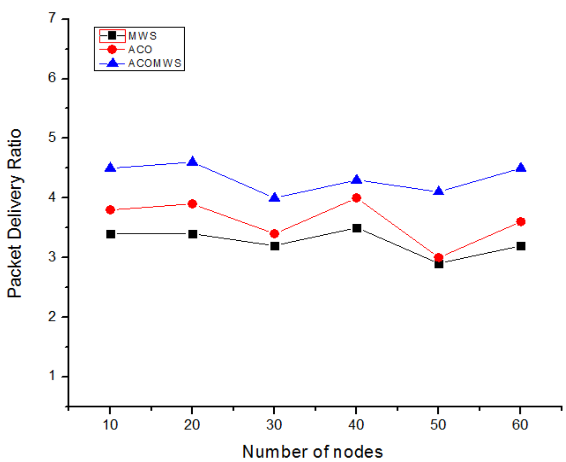

Figure 6: Number of nodes vs. available bandwidth

Routing Overhead

The routing overheads computed under varying pause times and varying number of nodes as shown in Figure 7 respectively. In view of the fact that more control packets are needed at the route discovery of the ACO phase and periodical update so that the extra control packets needs to perform route selection in the MWS phase. The routing overhead of the ACOWMS is slightly higher than that of other protocols. ***The routing overhead*** recognizes how ***to be reduced through piggybacking*** the pheromone information on data packets, even if appropriate traffic exists in the exact opposite direction. Due to the periodic updates, the ACOWMS needs a certain amount of routing overhead, but ***when the pause time increases, finally the overhead is reduced*** due to the relatively static nature of the topology.

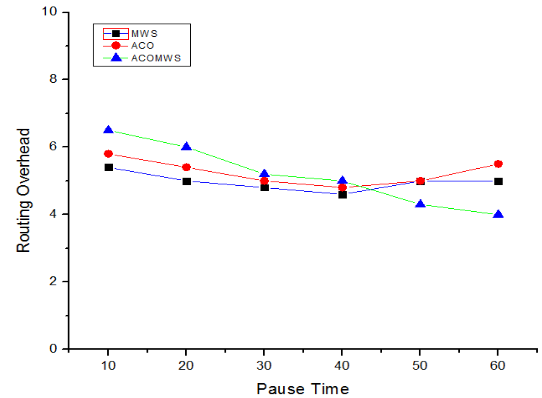

Figure 7: Node pause time (s) vs. routing overhead

To compare and evaluate the scheduling performance of MWS, ACO and the ACOMWS algorithm implement in a simple linear network. It consists of 6 links and 7 nodes in Figure 8, where nodes represent by links and circles are represented by dashed lines with link capacity in respective manner. The seven flows signify by arrows in which each flow is from node 1 to node i+1 through all the nodes in between. An uniform traffic performs between all flows contain packet arrivals at each time-slot next Poisson distribution with the same mean rate $\lambda > 0$ and to run our simulations with modifying traffic load with $\lambda < 0.5$ is feasible.

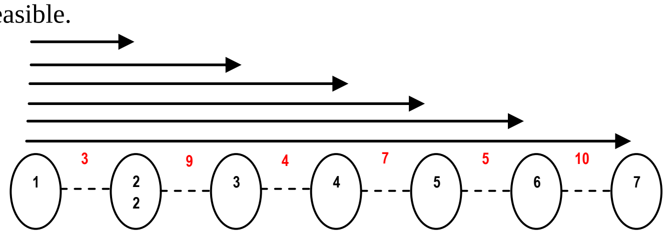

Figure 8: Linear Topology

Average Delay

The average delays under different offered loads to observe the performance limits of different scheduling approaches in Figure 9. The simulation result represents that lasts for 10^7 time-slots. As the optimal throughput region $\Lambda^*$is described as the set of arrival rate vectors under which queue lengths and therefore delays remain finite, consider the traffic load, under which the average delay increases quickly, as the boundary of the optimal throughput region. The entire schemes achieve the same boundary (i.e., $\lambda < 0.5$), and that maintains our theoretical results on throughput optimality in Figure 9. Additionally, ***the proposed schemes achieve considerably better delay performance*** than the other algorithms **(MWS and ACO)** as well as back-pressure algorithm. In the back-pressure algorithm, the queue lengths must build up along the route a flow obtains from the destination to the source, and generally, previous hop link has a larger queue length so that it leads to poor delay performance.

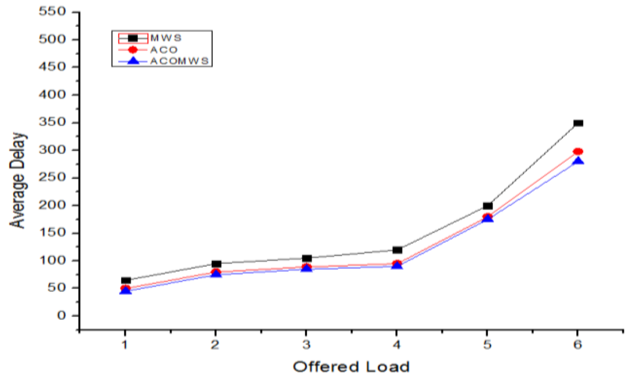

Figure 9: Average delay of MWS, ACO, and ACOMWS in a linear network topology.

Average Queue Length

In that MWS, ACO and ACOMWS have finite average queue length for $0 \leq \varepsilon < \frac{1}{7} = 0.143$ and therefore attain the maximum throughput in figure 10. Alternatively, the average queue length enhances linearly with ε under ACOMWS starting from $\varepsilon = 0$ and $\varepsilon = 0.04$, in respective manner. This involves that ACOMWS are throughput-optimal in this setting, at the same time as **ACOMWS achieves better throughput** ($\varepsilon < 0.04$) **with other algorithms**.

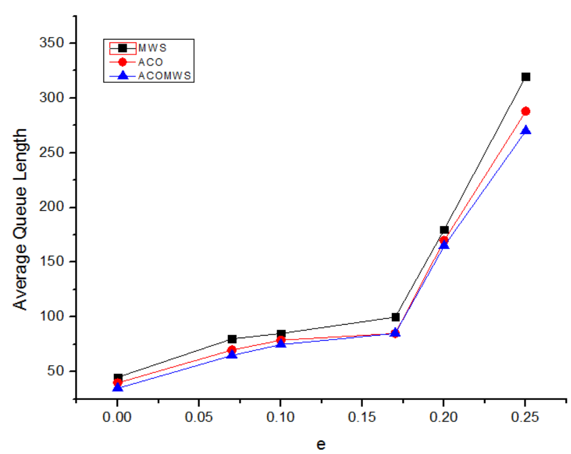

Figure 10: Average Queue Length comparison of MWS, ACO and ACOMWS

End-to-End Delay

The values of end- to- end delay for the scheduling algorithms ACO, MWS and ACOMWS simulated at different number of nodes shown in Figure 11. Higher end-to –end delay values imply that the routing protocol is not fully efficient and causes a congestion in the network. To compare other two algorithms considered ACOMWS exhibits lesser values of end-to-end delay. ***The ACOMWS shows a better performance than ACO and MWS*** and this implies for wireless networks. By means of different number of nodes circumstances also examined while the number of nodes enhanced, even if end-to-end delays are also increased.

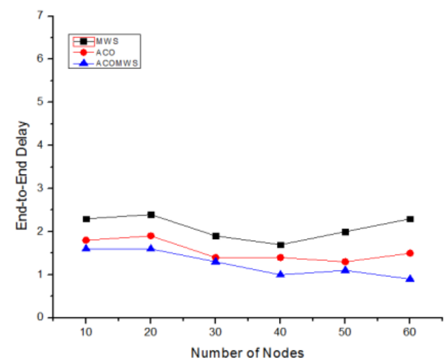

Figure 11: End-to-End Delay under various nodes

Delivery Ratio and Bandwidth

Bandwidth utilization and delivery ratio of ACO, ACOMWS and MWS scheduling algorithms show in figure 12 and 13. ACOMWS performs in results phase as well at different nodes. As bandwidth considers being a limited resource in the network, so that ***ACOMWS will automatically develop the revenues*** of the service providers. The packet delivery ratio is a comparative view for ACO, MWS and ACOMWS shown in Figure 13. ***ACOMWS demonstrates an enhanced delivery as compared with the other two algorithms***. For that reason the higher PDR ratio of ACOMWS know how to be attributed to its first-rate performance in large networks with low traffic. It finds scheduling on-demand, and effectively uses available bandwidth.

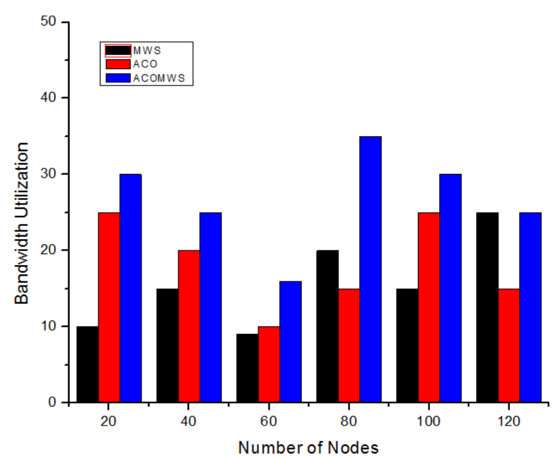

Figure 12: No of nodes vs. bandwidth

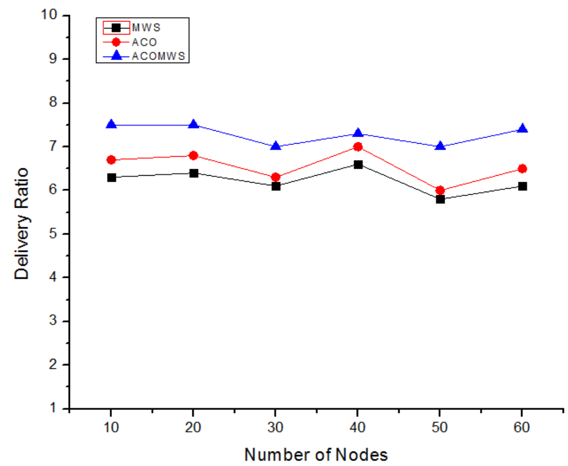

Figure 13: Packet Delivery Ratio vs. No of nodes

Routing Overhead

Routing overhead intended for ACOMWS was comparing with other MWS and ACO algorithms, as illustrate in figure 14. The routing overhead used for ***ACOMWS was performed better than the MWS and ACO.*** The routing overhead was low at less number of nodes. Its value was approximately equal with less number of nodes. The overhead enhanced with number of nodes, to compare the increasing value of routing overhead was performed more than the other two algorithms.

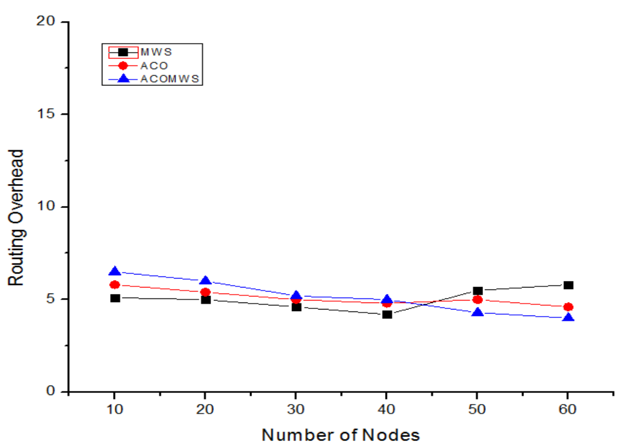

Figure 14: Routing Overhead Vs No. of node for ACO, MWS, ACOMWS

VI.CONCLUSION

Max-Weight is a familiar scheduling algorithm to perform in the presence of light-tailed traffic, in single-hop queuing networks. In Max-Weight scheduling policy, it is badly performed in the presence of HT traffic. To improve combined scheduling policies using an ACO based method Max-Weight scheduling, namely ***ACOMWS, to maximize the throughput optimization*** in wireless multi hop networks. To combine scheduling algorithm accomplish highest throughput rate in low complexity. From this paper, the throughput performance of ACO, MWS, and ACOMWS schedulers evaluate using ns-2. In simulation results, it illustrate that ***ACOMWS perform better than other scheduling algorithms including ACO and MWS.*** The proposed scheduling algorithm is proved the optimal solution and generate efficient throughput at low complexity and to achieve throughput optimality.

REFERENCES

[1]. G. Gupta and N. B. Shroff, "Delay analysis of scheduling policies in wireless networks," Asilomar Conference on Signals, Systems, and Computers, Oct. 2008.
[2]. A. Stolyar, "Large deviations of queues under QoS scheduling algorithms," Allerton 2006, Sept. 2006.
[3]. M. Neely, "Delay analysis for maximal scheduling in wireless networks with bursty traffic," IEEE INFOCOM 2008.
[4]. M. Neely, "Order optimal delay for opportunistic scheduling in multiuser wireless uplinks and downlinks," Allerton 2006, Sept. 2006.
[5]. M. Neely, "Delay analysis for max weight opportunistic scheduling in wireless systems," Allerton 2008, Sept. 2008.
[6]. L. Bui, R. Srikant, and A. Stolyar, "Novel architecture and algorithms for delay reduction in back-pressure scheduling and routing," IEEE INFOCOM 2009.
[7]. L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," IEEE Transactions on Automatic Control, vol. 37, no. 12, pp. 1936–1948, Dec 1992.
[8]. A. Ganti, E. Modiano, J.N. Tsitsiklis (2007). Optimal transmission scheduling in symmetric communication models with intermittent connectivity. IEEE Transactions on Information Theory, 53(3), pp. 998-1008.
[9]. L. Georgiadis, M. Neely, L. Tassiulas (2006). Resource allocation and cross-layer control in wireless networks. Foundations and Trends in Networking, 1(1), pp. 1-144.
[10]. A. Gut (2005). Probability: a Graduate Course. Springer Texts in Statistics, Springer- erlag.
[11]. B. Ji, C. Joo, N.B. Shro_ (2013). Throughput-optimal scheduling in multihop wireless networks without per-ow information. IEEE/ACM Transactions on Networking, 21(2), pp. 634-647.
[12]. B. Ji, C. Joo, N.B. Shro_ (2013). Delay-based Back-Pressure scheduling in multihop wireless networks. IEEE/ACM Transactions on Networking, 21(5), pp. 1539-1552.
[13]. D. Bertsimas, D. Gamarnik, J.N. Tsitsiklis (2001). Performance of multiclass Markovian queueing networks via piecewise linear Lyapunov functions. The Annals of Applied Probability, 11(4), pp. 1384-1428.
[14]. X. Lin and N. B. Shroff, "The Impact of Imperfect Scheduling on Cross-Layer Congestion Control in Wireless Networks," IEEE/ACM Transactions on Networking, vol. 14, no. 2, pp. 302–315, April 2006.
[15]. L. Tassiulas and A. Ephremides, "Stability Properties of Constrained Queueing Systems and Scheduling Policies for Maximal Throughput in Multihop Radio Networks," IEEE Transactions on Automatic Control, vol. 37, no. 12, pp. 1936–1948, December 1992.

[16]. B. Hajek and G. Sasaki, “Link Scheduling in Polynominal Time,” IEEE Transactions on Information Theory, vol. 34, no. 5, September 1988.

[17]. S. Sarkar and L. Tassiulas, “End-to-end Bandwidth Guarantees Through Fair Local Spectrum Share in Wireless Ad-hoc Networks,” in IEEE CDC, December 2003.

[18]. P. Chaporkar, K. Kar, and S. Sarkar, “Throughput Guarantees in Maximal Scheduling in Wireless Networks,” in the 43rd Annual Allerton Conference on Communication, Control and Computing, September 2005.

[19]. X. Wu and R. Srikant, “Bounds on the Capacity Region of Multihop Wireless Networks Under Distributed Greedy Scheduling,” in IEEE INFOCOM, April 2006.

[20]. G. Sharma, N. B. Shroff, and R. R. Mazumdar, “On the Complexity of Scheduling in Wireless Networks,” in ACM Mobicom, September 2006.

[21]. L. Tassiulas and A. Ephremides, “Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks,” IEEE Trans. Automatic Control, vol. 37, no. 12, pp. 1936-1948, Dec. 1992.

[22]. Dorigo M, Birattari M, St□utzle T, Bruxelles D, Roosevelt AF. Ant colony optimization artificial ants as a computational intelligence technique. IEEE Comput Intell Mag 2006; 1: 28-39.

Authors Profile

B.Sindhupiriyaa received her M.Phil Degree from Periyar University, Salem in the year 2015. She has received her M.C.A Degree from Anna University,Coimbatore in the the year 2011. She is pursuing her Ph.D Degree at Sri Vijay Vidyalaya College of Arts & Science (Affiliated Periyar University),Dharmapuri, Tamilnadu, India. Her areas of interest include Mobile Computing and Wireless Networking.

Dr.D.Maruthanayagam received his Ph.D Degree from Manonmaniam Sundaranar University,Tirunelveli in the year 2014. He received his M.Phil Degree from Bharathidasan University, Trichy in the year 2005. He received his M.C.A Degree from Madras University, Chennai in the year 2000. He is working as HOD Cum Professor, PG and Research Department of Computer Science, Sri Vijay Vidyalaya College of Arts & Science, Dharmapuri, Tamilnadu, India. He has 17 years of experience in academic field. He has published 4 books, More than 25 papers in International Journals and 32 papers in National & International Conferences so far. His areas of interest include Computer Networks, Grid Computing, Cloud Computing and Mobile Computing.