ISSN: 0974-5823 Vol. 7 No. 5 May, 2022

International Journal of Mechanical Engineering

EFFICIENT ROUTING SCHEDULING IN MOBILE AD HOC NETWORK USING ANY COLONY OPTIMIZATION WITH MAX WEIGHT SCHEDULING ALGORITHMS

Dr.B.SINDHUPIRIYAA,

Assistant Professor, Department Of Computer Science and Computer ApplicationsPadmavani Arts & Science College for Women, Salem, Tamilnadu.

Abstract: The Scheduling is one of the decision-making processes which become the most important role in MANET Environment. Scheduling approaches has been recognized to be a key issue for capacity/throughput optimization in wireless networks. In existing scheduling methods, some efficient factors related to high throughput and bandwidth not been considered in the scheduling operation. Several factors in the scheduling process have been mechanically decreased the performance level. To overcome such drawback in the existing techniques, new efficient optimization algorithms have been emerged in the field of Multi hop wireless Networks. The proposed ACOMWS algorithm will be an incorporate of the two heuristic techniques namely, ACO and MWS. The combination of these techniques, proposed ACOMWS method will efficiently choose the to the optimal resources based on Path Stability (PS) factor, link capacity constraints, flow efficiency constraints, routing constraints, Max-weight independent set in the interference graph.

Keywords: Mobile Ad hoc network, Ant Colony optimization, Greedy Algorithms, Max Weight Scheduling

I.INTRODUCTION

In Mobile Ad Hoc Networks (MANET), the multiple tasks of nodes as terminals, and routers, and mobility induced transmission of route packets, and multi-hop forwarding of packets, which may generate unique queue dynamics. The alternative choice of scheduling technique to find out which queued packet to process next will have a major effect on the overall end-to-end performance even if the traffic load ishigh. So it describes different scheduling algorithms were discussed [12]. In recent times, the MANET utilize a easy priority scheduling algorithm for simulation in which data packets are scheduled in FIFO (First In First Out) order, and all routing packets (RERR, RREP and RREQ) are indicated priority over the transmission of data packets at the network interface queue. Different routing protocols make use of different methods of scheduling. The drop-tail policy is utilized as a queue management algorithm in all scheduling algorithms for buffer management that provide high priority to manage the data packets, when the buffer is full. But the non-priority algorithm, all the other scheduling algorithms give top priority to control packets than to data packets. At present, only priority scheduling is allowed to utilize in MANETs [12]. Different scheduling approaches are formulated by using considering fairness, distance metrics, and applying the multiple tasks of nodes as both data sources and routers. The scheduling approaches that provide higher load to data packets with shorter geographic distances or smaller numbers of hops to their targets to improve the average throughput and decrease the average delay [13]. The main two classifications of network traffic are data packets and control packets, and that may contain different types of scheduling in Ad Hoc networks.

In the wireless network and wire-line Network, the delay stability is one of the significant metric considered. The delay analysis of packets plays an essential job in the network. Decomposition of packets partition into multiple paths, check whether if any two nodes assemble at same point bottleneck occurred. ***To overcome these issues utilizesa new delay efficient scheduling policy.*** For recognizing the behavior of the each path in the network lower bound analysis is used. Different policies are used for scheduling the packets that provides better optimality. Therefore, the new approaches are needed to address the delay trouble in multi-hop wireless networks. The most important goal of the work is to design a network as well as a new scheduling approach that provides minimizing the delay, better operation of resources in the network and increasing optimality. A comparative result analysis of proposed scheduling algorithm (ACOMWS) is evaluated with other algorithms (***Greedy, and GMWS)***.

Greedy Scheduling Algorithm, **Greedy Max Weight Scheduling (GMWS)** have to be applied to optimal scheduling operation better operation of resources in the network and increasing throughput. In this research paying more attention on scheduling approaches includes Greedy Scheduling Algorithm, Greedy Max Weight Scheduling (GMWS). The above techniques are achieved throughput optimal solution **only if the network remains stable condition** otherwise it has unsuccessful conditions.

Ant colony optimization (ACO) algorithm is based on the Bio-inspired algorithms; it has utilized to enhance the routing algorithms for MANETs. Routing in MANETs is a challenging due to MANETs dynamic features, its limited power energy and bandwidth. ACO is particularly like a well-known swarm intelligence approaches and it has acquired the inspiration from real ants. In such case, they are wandering around their nests for searching the food. In the lead of discovering food they will return back to their nests at the same time as deposit pheromone trails next to the paths, its next step depending on the quantity of deposited pheromone on the path to the next node. The ants may not handle anything except small control packets, which have the task to search and gather whole information about it and a path towards their destination. The main problem for searching shortest paths maps rather than the issue of in this concept, at least half of the capacity region is archived by using a greedy scheduling policy. The full capacity region in several different networks accomplishes using Greedy scheduling. Greedy scheduling policies empirically perform nearly in addition to an optimal scheduling policy. Additionally, the greedy partitioning algorithm makes use of dividing the entire wireless network into many single - queue systems and obtain bound on expected delay performance. Throughput maximization on multi hop can be mathematically states as below.

routing in networks.

Maximize $\sum^{T}$ $i=1$

IC_i

The main scope of this research has to be providing better scheduling method by solving the drawbacks that currently exist in the literature survey. The proposed work has considered a few efficient factors like i) High throughput ii) High Bandwidth iii) Minimize end to end delay iv) Good Packet Delivery Ratio v) Average Queue Length. The proposed **Ant Colony Optimization Max Weight Scheduling (ACOMWS)** method professionally does

Let T and C be the number of tasks and cores in the system. A greedy algorithm intended for the difficulty of run-time multi hops scheduling. The proposed algorithm is composed of following sequential steps performed by the greedy scheduler before every scheduling epoch.

- Assume that $C_i = 0\ \forall i \in T$.
- All tasks in T sort arrange in ascending order by

stabilize the system and have better buffer-usage performance than the other algorithm. The Scope of the

using comparator [$IC_i + 1 - IC_i$

] and store in a queue.

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***, obtains higher throughput optimality and minimizes the delay performance*** in multi-hop heavy tailed networks.

II. SCHEDULING ALGORITHMS2.1.Greedy Algorithm

In Greedy algorithm, it has a set of algorithms which have one general characteristic, making the best option locally at every step without considering any future plans. Therefore, a set of options to select the current best option is provided by the essence of greedy algorithm. [53]. On the other hand, some certain problems know how to easily be solved by using greedy algorithm like Interval Coloring, Knapsack Problem, Huffman Code and Interval Scheduling (Activity Selection) Problem. In greedy approach, a node makes a decision to the transmission path on the position of its neighbors; the source assesses the location of the destination node with the organize of its neighbors and propagates the information to the neighbor which is closest to the final destination.

The process is repeated until the whole packet arrives at the deliberate destination. During the closeness of the concept isproposed and related to several metrics in this area and that the popular metrics is the projected line and the Euclidean distance joining the destination and the relay node .Intended for the retransmissions process, the untrustworthy neighbors are not taken into action. In the multi hop network area, the main use of greedy policy is consecutively admits links to a schedule in a greedy fashion.

Virtually assign a core to Task j in front of the queue and update the corresponding IC_j using performance prediction models.

- Reposition the Task j according to the updated IC in the sorted queue using a binary search insertion. j
- Repeat Step 3 and Step 4 till all cores are allocated.
- Readjust real core allocations from last scheduling epoch to reflect the new optimal core allocations.
- Execute tasks with the optimal core allocations.

The greedily searching algorithm proceeds as below figure

2.1 for a (K, X) bottleneck which yields the maximum lowerbound for computing operation in the multi hop networks. The flows transferring through the selected bottleneck are removed and that process is repeated until the entire flows are utilized. Therefore the wireless network decomposes into numerous single queue systems.

The system average delay can be effortlessly computed. Remind that the decomposition process obtained by the greedy algorithm but it is not the optimal decomposition. Finally, the optimal decomposition can be alternately obtained by using a dynamic programming technique with the cost of increased computation complexity. The drawback of the greedy algorithms is the transmission may possibly be unsuccessful in that case if the current holder of the message has no neighbors closer to the receiver apart from itself. Even if there is a possible path between the two extremes, for example, when an obstacle is comes into existence.

1: Z ← {1, 2 . . .N}

2: BOUND ← 0

3: **repeat**

4: Find the (K, X)-bottleneck which maximizes E[DX]

5: BOUND ← BOUND + E[DX]

6: Z ← Z \ i : i ∈ X

7: **until** Z = Φ

8: **return** BOUND

Figure 2.1: Computing the Lower Bound

2.2. Greedy Max-Weight Scheduling (GMWS)

The combination of greedy techniques with Max-Weight Scheduling, it accomplishes more throughputs at low complexity [54]. The GMWS algorithm is naturally low- complexity different to MWS. The scheduling difficulty of multi-hop wireless networks find out set of links such that

$$\mathrm{r} = \underset{r\in C_o(R)}{\arg\max} \sum_{l=1}^{L} r_l q_l$$

Let r_l be the rate of the link l and q_l denotes the implicit price (i.e. length of the queue) of link l. This problem can be interpreted to the matching issue in node-exclusive interference model. The weight of link l is $w_l = q_l c_l$. After that rewriting above equation as

$$\mathrm{x} = \underset{r\in C_o(\pi)}{\arg\max} \sum_{l=1}^{L} x_l w_l$$

Let be π consists of a feasible set of matches performing under node-exclusive interference model. Greedy Max- weight knows how to be viewed as an approximation to MWS. A familiar sub-optimal algorithm is GMWS, which is also called Longest Queue First (LQF). The operation of GMWS is quite natural; in this scheduling the opportunity to transmit is given to a link with higher weight prior to the links with lower weight (i.e. shorter queue length). By using this model a wireless network by using a graph G= (V, E), where the set of nodes denote as V and E the set of links. Nodes were represented in wireless transmitters and receivers, and a link between two nodes directly communicates with each other.

For any link∈, the set figure 2.2 of its interfering links define as I(l) = { l' ∈ E | l' interferes with l} Assume that if l' interferes with l ,after that l also interferes with l'. A schedule of graph represent G= (V, E), it has a subset of links M ⊆ E that considers scheduled/activated at the same time with the interference constraint, that is no two links M interfere with each other. Assume that all links have unit capacity, for instance, a scheduled link transmits one packet in one time slot. A schedule is to be maximal even if no link can be added to it without violate the interference constraint. The capacity region of the network consists of the set of each and every arrival rates for performing with exist scheduling algorithm which stabilize the queues, where the queues are bounded in some deterministic sense or appropriate stochastic based on the arrival model utilized.

The GMWS algorithms operates as follows: Begin with an empty schedule; first select the link l with the largest backlog; add l into the schedule, and disable other links in I

(l); subsequently, choose the link l' with the largest backlog from the remaining links, add l' into the schedule, and disable other links I (l'); this process continues until the entire links are either disabled or selected. The entire chosen links {l, l',….} is scheduled during time slot.

GMS(E) := ∅

E' : = E WHILE ($'\neq\emptyset$)

Pick a globally heaviest link *l :*

$l \in$ arg max$l' \in E'$ (l')

$l' \in E'$

GMS(E) : = GMS(E) ∪ {l}

E' : = E'\ {l}\ I {l}ENDWHILE

Figure 2.2: Greedy Max-Weight Scheduling algorithm A scheduling algorithm achieves the maximum throughput,or is throughput optimal solution, when the capacity regionremains stable for the entire arrival rates in the network.

2.3. ACO-MWS Algorithm (ACOMWS)

In multi-hop networks, the Max-Weight type algorithms do stabilize the system and have better buffer-usage performance than the other algorithm. MWS algorithm performs as a weight of (queue-length X channel-rate), and schedules a collection of links that increases the total weight (max-weight independent set). A rate vector interior to the capacity region, a randomized, stationary, queue- independent policy in principle be designed to stabilize the system, although this requires full acquaintance of the channel state probabilities and traffic rates. The following queue-aware *MW* policy become stable the system when the rate vector is interior to, without requiring knowledge of the channel statistics or traffic rates: Each time slot *t*, analyze current queue backlogs and channel states *Qi(t)* and *Si(t)* for each link *i*, and choose to serve the link *i*(t) {1, . . . ,N}* withthe largest *Qi(t)Si(t)* product. This is known as the LCQ (*Longest Connected Queue* policy) [2], as it serves the queue with the largest backlog among all that are currently ON. Max-Weight policy, the scheduling vector S (t) belongsto the set:

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

Max-weight policy requires more statistical knowledge to implement. T**he proposed technique that aims to achieve Max-Weight with ACO algorithms** decide how much data is to be injected into the network. The aim of this research isto maximize the total utility of traffic injected into the network**,** and ***obtains higher throughput optimality*** then***minimizes the delay performance*** in multi-hop heavy tailed networks. ACO algorithm has successfully applied to several optimization techniques and combinatorial issues. Ant foraging process is to discover the routing issues of MANETs. ACO can be used by the pheromone mechanism and maintain optimal scheduling in ad hoc networks. The

mechanism of evaporation updates the pheromone of eachnode, which can rapidly adapt to the requires of the dynamic

$\Delta r_{ij} = \sum^{l}$

$\Delta\tau^{k}$. $k=1$ ij

changes of ad hoc networks. In wireless networks, ACO approach utilize adaptive learning of routing tables. Each node *k* in the network stores some data structures, which is responsible for keeping routing table and local traffic statistics. Local traffic statistics describes a simple parametric statistical model for traffic distribution over the network as seen by node *k*. In such case, it maintains trackof the amount of traffic flows towards each possible destination. Intended for each possible destination represents d and also node *n*, stores a probability value Pnd in which communicates the popularity of selecting *n* even if the destination node represent as *d*. The amount of pheromone deposited on the link (*k, n*). When an ant at node *k* heads toward a destination node *d*, it chooses the next neighbor node *n* with the probability $P'nd$

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

Where |*Nk*| is the number of the neighbors of node *k*, qn 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 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 $\lambda = (\lambda_1, \ldots, \lambda_F)$ isin the stability region Λ of the multi-hop switched queuing network described above if there pre-existing $\zeta f,i,j \geq 0$, fF, i, $j \in N$ such that the following set of constraints is satisfied: $\in$

- Flow efficiency constraints

$$\zeta f,i,j = \zeta f,i,sf = \zeta f,df,i = 0, \quad \forall_i \in N, \quad \forall_f \in F ;$$

- Routing constraints

$\zeta f,i,j = 0,$ $\forall(i,j) \notin Lf, \quad \forall f \in F;$

- Flow conservation constraints

$$\sum \zeta f,i,j + \lambda f.\, 1\{i=sf\} = \sum \zeta f,i,j, \forall_i \neq d_f, \quad \forall_f \in F;$$

ant reaches the destination node, it can then evaluate the

$j \in N$

$j \in N$

goodness of the path. The goodness of the path can bedefined 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

- Link capacity constraints

$\sum \zeta f,i,j < 1,$ $\forall(i,j) \in L.$

$j \in N$

If an arrival rate vector is in the stability region, after thatpre-exists a policy that stabilizes the network, in the sense of

in the network is $m = \sum_{i=1}^{n}$

$a_i(t)$; (t) is the probability of

stability Definition. The stability region depends on the

choosing link j for ant K at time .

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

Where (t) is the strength of the pheromone in the link Eij; α is a parameter to measure the trajectory of pheromones; ηij is visibility between node i and node j, which is generally defined as $1/dij$ (dij is the distance between node i and node

j); β is a parameter that measures visibility; and allowed$k$ isa collection of nodes that have not been visited. The pheromone update formula on each path in ad hoc networks

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 networkdescribed above under a stabilizing policy. If queue *(f, i)* is delay stable, for all $i \in N_f$, then traffic flow f F is delay stable.

∈

is as follows:

Δr_{ij}

$r_{ij}(t+1) = (1-\rho)r_{ij}(t) +$

∈

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

Where ρ is the pheromone volatilization coefficient, which is a constant between 0 and 1, and Δrij is the increment of the pheromone of ants passing through links i and j.

III. EXPERIMENTAL RESULTS

With the intention of simulation process the Greedy, Greedy with Max Weight Scheduling and ACO with MWS for handling a comprehensive simulation environment (Network Simulator-2). The network simulator covers a huge number of applications of multiple kinds of protocols of variable network types consisting of variable network fundamentals and traffic model. For the implementation process, both the stand alone the max weight scheduling algorithm and backpressure algorithm, Network Simulator (NS-2) is most well-liked simulation software and it is a separate eventpacket level simulator. To imitate performance of network simulator is a complete wrap up of tools like creating network topologies, log events that takes place under any load to examine the events and comprehend the network. NS-2 was and preserved by USC and developed by UC Berkeley. In scientific environment, network simulator is one of the most popular simulators and it is based on the twolanguages such as OTcl and C++. OTcl is defined an object oriented version of Tool Command language.

In simulation process, it considers three topologies such as atree topology, These nodes are located over 500m × 500m terrain and it denotes S1, S2 and R1, R2 which are likely source-receiver pairs in the tree and diamond topologies. The tree topology of S1 figure 3.1 representation is initiated from node A and ends on node B, and S2 representation is initiated from node A and ends on node D.

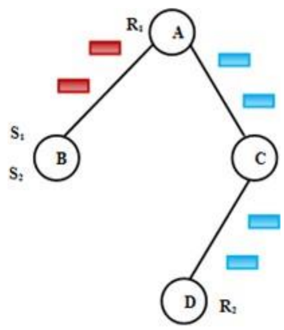


Figure 3.1 Tree topology

Parameters for the simulation scenario:

Simulator: NS-2 Network structure: tree

Node initial placement: random baseMobility model: random waypoint Node communication range: 250 m

Medium access mechanism: IEEE 802.11bTraffic source model: CBR

Packet capacity: 1000 B Channel capability: 1 mbps

Buffer size at each node: 1000 packets

Node speed: 10 m/s Pause time: 0-480 s Simulationtime: 900 s

The MAC layer is performed by using IEEE 802.11b. In terms of wireless channel, it simulates a Rayleigh fading channel among average channel loss rates 0, 20, 30, 40, 50%. This process have been repeated each 100sec simulation for 10 seeds. The channel capacity is 1 Mbps, packet sizes are set to 1000Bytes, and before that the buffer size at each node is set to 1000 packets. To compare routing schemes such as max weight scheduling and ant colony optimization, in terms transport-level throughput. With the simulation results is provided to reveal the performance of the scheduling policy with the support of GMWS can be carried out by the execution of the similar algorithm and taking the result of the algorithms and analyze the performance between them.

Average Queue Length: The traffic load vector is randomly selected by multiplying a factor. The scaling factor represents by using the x-axis. Intended for each scheduling policy, a certain threshold rapidly enhances under the total queue length. The boundary of the capacity region of *ACOMWS* is the largest that it shows in figure 3.2. On the other hand, the performance gap between *GMWS and Greedy* is relatively small. Particularly, the capacity boundaries of *GMWS and Greedy* are almost the same.

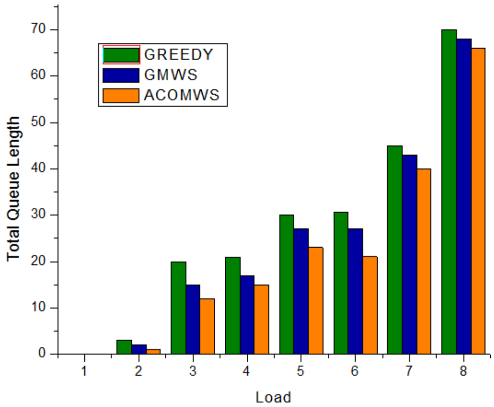
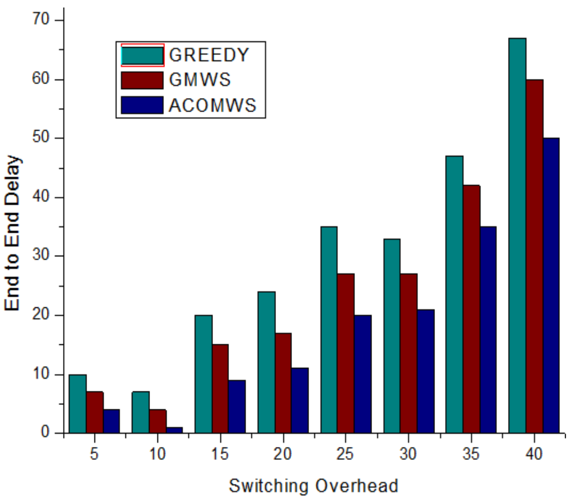


Figure 3.2: Average Queue Length

Average Delay: End-to-End delay varies in switching overhead δ in Figure 3.3.



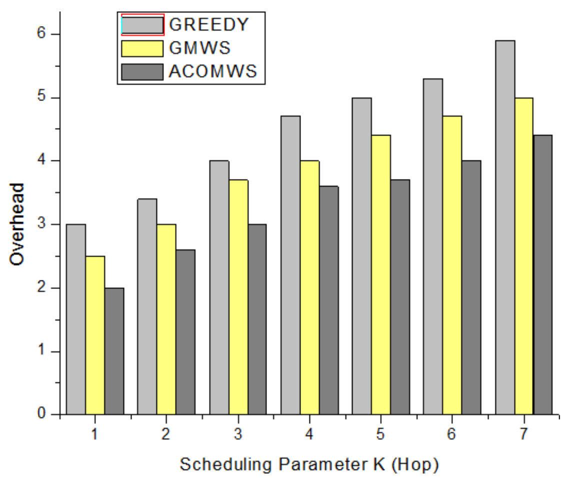


Figure 3.3: Average Delay

Figure 3.4: Average Communication Overhead

So that the throughput of ACOMWS is significantly higher, even if the end-to-end delay emerges to be related between Greedy and GMWS is a significant development over MWS. Moreover, MWS proves a huge improvement over GMWS. ACOMWS observation is the drop in delay as switching delay δ changes from 0.4 to 0.5 and 0.6 to 0.7. Other than, the throughput considerably dropped at those intervals, and it does not indicate improvement those drops in delay for end-to-end delay.

Overhead: The effect of *k (hops)* on the average overhead is shown in Figure 3.4, which shows that our approach can reduce the overhead by up to 75% when $k = 1$ and up to 50% when $k = 4$. Also, the overhead reduction drops if the value of *k* increases that demonstrates in figure 3.4. The most important reason is that a larger value of *k* incurs a larger Operating Set, in which more control messages will be exchanged between nodes. Additionally, the overhead of this approach as well increases when the number of edges in the conflict graphs increases, because the additional network interference needs the scheduling algorithm to produce morecontrol messages for information exchange. Also when the number of edges increases, the same value of *k* leads to a larger operating set and involves more nodes into the rescheduling process.

Throughput: The throughput performance is to comparewith different algorithms for varying switching overhead δ in figure 3.5.

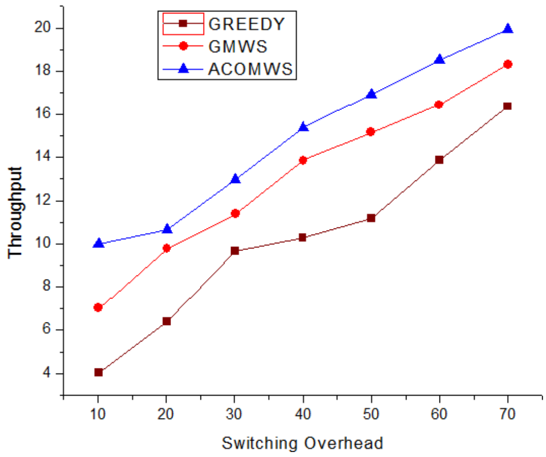


Figure 3.5: Average throughput

The throughput ratio analyzes between the total received packets and the total sent packets. The throughput of GMWS and Greedy algorithms (both implemented without consider about switching overhead) decreases dramatically even if δ are larger than 0.3 and 0.1in respective as shown in figure 3.5 However, the centralized algorithm ACXPMWS has provided the same performance with varying δ. Even if the throughput of Greedy and GMWS shows a decrease computation speed, compared to ACOMWS, it shows dramatic enhancement in stability process. In order to demonstrate the improvement of throughput, the ACOMWS algorithms can accomplish seven to twelve times of high throughput than other algorithms shown in figure 3.5.

Packet Delivery ratio: The scheme (ACOMWS) provides a better packet delivery ratio value better than the other scheduling algorithms in figure 3.6. Through the decrement of no of nodes, the proposed method constructs an effective usage of the batteries in the MANET. The packet delivery ratio increases by providing the packets, and that consumes higher transmitting priorities, more energy. In case of packets whose energy transmission path level is low, which are prioritized with high transmitting priority. The Basic Scheduling accomplishes better delivery ratio at network energy level of 60% in figure 3.6.

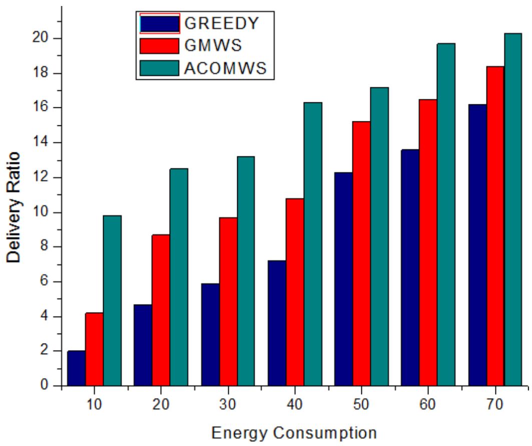


Figure 3.6: Packet Delivery Ratio

The proposed scheduling algorithm create more predictable behavior, in multi hop nodes are often dedicated to the Ad hoc networks and are not preempted by outside work. This enables schedulers to compute the accurate high throughput and minimize packet delay in between multi hop nodes, when their running characteristics are known. ACOMWS also used for measurement information about the current utilization of nodes to determine which ones are not routing overhead before transmitting packets via undefined networks. Hence, the proposed ACOMWS scheduling algorithm is capable of finding the optimal throughput for better packet transmission even if heavy traffic flow and achieve good bandwidth with maximum packet transmission in before allotted time to the desired destination nodes.

As a result, our proposed algorithm ACOMWS to improve the throughput optimality, packet delivery ratio, reduce the routing overheads and also minimizing the delay in multi- hop wireless networks. In this research, greedy and greedy max weight algorithms (GMWS) are effectively compared and performed over the real-time simulation environment that exhibits the efficiency of ACOMWS. By using ACOMWS satisfies a tradeoff between the throughputs and minimize end-to-end delay bound while assuring the high data rate requirements for delay stability of traffic flows. Finally, conclude that the ACOMWS provides better performance as compare to other algorithms.

IV. CONCLUSION

In this paper, the proposed method ACOMWS has achieved the optimal throughput, minimize packet delay, end to end delay and reduce overhead as well as increased bandwidth. The drawbacks of existing methods (GREEDY and GMWS) solved by considering some efficient factors in a scheduling operation. We investigated chosen jobs had been allocated to the best selected resources of each iterations. This process is repeated until all jobs have been scheduled and a complete solution has been built. This proposed technique can find an optimal throughput plus minimize packet delay and network for each machine to receive packets before allotted even though more traffic. The scheduling algorithm is proposed for achieving high bandwidth, high throughput, and prevent packet delay performance in Wireless Ad hoc Networks. An important direction for future research is to consider queuing networks with correlated traffic. In previous some evidence recommends that traffic in real- world networks exhibits strong correlations, and phenomena like long-range dependence traffic and self-similarity with service for leading to large delays, long periods of time at conflicting queues.

V. REFERENCES

[1]. Byung-Gon Chun, Mary Baker," Evaluation of Packet Scheduling Algorithms in Mobile Ad Hoc Networks", MobileComputing and Communications Review,Volume 6,Number 3

[2]. P. Bhagwat, P. P. Bhattacharya, A. Krishna, and S. K. Tripathi, "Using channel state dependent packet scheduling to improve TCP throughput over wireless LANs," ACM Wireless Networks, vol. 3, no. 1, pp. 91–102, 1997.

[3]. Sridhar K N, MunChoon Chan. Channel-aware Packet Scheduling for MANETs.2008 IEEE, 978- 1-4244-2100.

[4]. A. L. Stolyar, "Maximizing Queueing Network Utility subject to Stability: Greedy-Primal Dual Algorithm," Queueing Systems, vol. 50, no.4, pp. 401-457, 2005.

[5]. 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.

[6]. Long Bao Le, Krishna Jagannathan, and Eytan Modiano, "Delay Analysis of Maximum Weight Scheduling in Wireless Ad Hoc Networks", NSERC Postdoctoral Fellowship and by ARO Murigrant number W911NF-08-1-0238.

[7]. S.Sindhupiriyaa, Dr.D.Maruthanayagam, Comparative of Delay Tolerant Network Routings and Scheduling using Max-Weight, Back Pressure and ACO. International Journal on Recent and Innovation Trends in Computing and Communication Volume: 6 Issue: 7, 2018.

[8]. Yogesh Hole et al 2019 J. Phys.: Conf. Ser. 1362 012121