INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

OPEN ACCESS
Received: 17-01-2023
Accepted: 24-08-2023
Published: 13-10-2023

Citation: Joshi SS, Parvati VK (2023) A Novel Packet Scheduler Policy for Multi-Hop Wireless Networks Using Diameter Based Consensus Algorithm. Indian Journal of Science and Technology 16(38): 3186-3194. https://doi.org/10.17485/IJST/v16i38.105

* **Corresponding author**.

sachinjoshi055@gmail.com

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

A Novel Packet Scheduler Policy for Multi-Hop Wireless Networks Using Diameter Based Consensus Algorithm

Sachidanand S Joshi[1]*, Vasudev K Parvati[2]

1 Assistant Professor, Information Science and Engineering, SDM College of Engineering & Technology, Dharwad, Karnataka, India
2 Associate Professor, Information Science and Engineering, SDM College of Engineering & Technology, Dharwad, Karnataka, India

Abstract

Background: Over the time, the wireless communications have become more universal. Development of packet scheduling algorithms in wireless networks can proficiently improve delivery of packets via wireless links. Method of packet scheduling can promise quality of service and enhance transmission rate in wireless networks. **Objectives:** To develop an efficient packet scheduler policy to reduce the average End to End delay among the flowsand to fix the number of queues by finding diameter and TB-DVA. **Methods:** The timed-buffer distributed voting algorithm (TB-DVA) is a kind of voting algorithm that used to determine the maximum distance reported by all the nodes. We have used NS-3 simulator for simulation model and average throughput and average packet delay parameters are considered; we have compared the novel TB-DVA based Packet Scheduler policy with the traditional AODV Packet Scheduler policy. Mention the dataset considered. How comparison made with gold standards **Findings:** Comparing the new packet scheduler policy to the standard AODV packet scheduler policy, the simulation revealed that the new packet scheduler policy provides lower average end-to-end latency and high throughput. In comparison to the traditional AODV packet scheduler policy, average end-to-end delay is reduced by 7.14 percent and average throughput is enhanced by 9 percent. **Novelty:** The farthest node is determined by a distributed vote across all nodes.

Keywords: Wireless Networks; Packet Scheduler Policy; AODV; MultiHop

1 Introduction

With the development of wireless technologies, wireless networks in recent years have been widely used in public places such as libraries, hotels, schools and airports due to the greater flexibility, increased efficiency, and reduced wiring costs [1]. Wireless networks refer to any kind of computer network that does not involve any cables or wires. It is a technique that helps industrialists and telecommunications networks to save the cost of cables for networking in specific premises in their installations [2].

As one of the key functions in wireless communication networks, link scheduling determines which links should be activated at what time. The problem is challenging, in particular, in multi-hop networks due to non-linear interference relationship between wireless links[3]. Packet scheduling refers to the decision process used to select which packets should be serviced or dropped. Dropping of packets will be based on network characteristics like bandwidth, packet arrival rate, deadline of packet and packet size. Scheduling will be done in scheduler. A scheduler will find it difficult to handle all the packets coming in if the packet rate is high, if the bandwidth is too low or the packet size is large. So, the scheduler will select certain packets based on various algorithms[2].

A DEEC protocol variant-based, energy-efficient clustering heterogeneous protocol was provided by[4] for distributed WSNs. In a MATLAB simulation, authors compared the proposed protocol to IDEEC based on metrics such as alive and dead nodes during network life, sensor network throughput, number of packets received by BS and CH, packet delivery ratio, and algorithm overhead. IoT-DEEC surpasses IDEEC in all metrics and is more efficient at enhancing network lifetime by reducing energy usage. Hence, authors propose a new energy consumption CH selection algorithm for heterogeneous WSNs.

Wireless network coordination and control was examined by[5]. Authors highlight the main technical challenges to reduce the gap between industry and smart manufacturing using real-world use scenarios. Author's control-over-wireless technology addresses these issues. Cyber-physical test bed experiments and theoretical closed-loop stability analysis enable smart manufacturing scenarios. This paper ends with questions and research ideas.GA-LSPI-based WSN centralized routing protocol was developed for lifetime and energy optimization[6]. Sink GAs generate network graph routing tables in polynomial time. After sensor nodes expire, the sink's LSPI learns the lifetime and energy-optimal routing path at each network graph level. This maximizes the sink's inaccessibility to alive sensor nodes while lowering network energy consumption.

Spanish-funded ALLIANCE report highlights new architecture includes monitoring, trust, the SDN/NFV cross-layer domain, RINA, and KDN orchestration. This project fixed 5G/B5G networking difficulties[7]. Energy-efficient Adaptive Route Decision Sink relocation procedure using Cluster Head Chain Cycling was proposed by[8]. The strategy lowers energy consumption. The proposed solution minimized transmission energy, data aggregation efficiency, and network longevity. WSN may develop a strategy for fault tolerance.

A dynamic WSN efficient node stable routing (ENSR) protocol to guarantee the stability of data transmission between source and destination nodes was proposed by[9]. Simulation results demonstrate that the stable routing protocol eliminates unnecessary routing and outperforms conventional routing technologies in terms of network efficiency and dependability. A data routing system was developed by[2] to utilize a small number of sensor nodes to cover a vast region, thereby reducing node deployment costs and facilitating the tracking of land mine explosions.

There are many ways to provide the Quality of Service. On one extreme provide enough bandwidth that enables us to meet the service level agreement (SLA) specified in terms of latency, bandwidth and Jitter. On another extreme, use the available resources in steadfast that will enable to adhere to the SLA. Since wireless Adhoc networks are scarce resources, the latter option is appropriate. In this work, we have used a later approach wherein we have changed the default scheduler policy of AODV protocol. This enabled us to reduce the latency of the flows between pair of nodes which are distant apart and thereby reducing the average delay for all the flows. Using an AODV as a routing system, we have made an attempt to minimize the average delays for flows in a wireless ad hoc network.

In the present study, we have developed a consensus approach to determine the maximum distance of the deployed wireless network based on the number of hops. We have also proposed an admission and scheduling method for port queues that changes the way packets are picked out of the queue for transmission.

2 Methodology

In this work, we have made an attempt to compute the diameter of the deployed wireless network which is necessary to obtain the Quality of Service, considering the delay as a parameter. The purpose is to schedule the longer flow packets ahead of the packets belonging to shorter distance flows. The basic idea is to reduce the increase in delay for the flows to travel farther by prioritizing them over the flows that need to travel less. Since we cannot go for the entire hop count space, we have defined ranges based on hop count to achieve the diameter.

2.1 Diameter Finding Algorithm Using TB-DVA

To deliver Quality of Service (QoS) in terms of reduced average end-to-end latency, we've built a packet scheduler that ensures the delay of packets requiring a longer journey is less than that of packets requiring a shorter journey. To achieve this purpose, we needed to establish the diameter of the wireless network that had been deployed. This is an attempt to determine the diameter of the wireless network that has been installed. In distributed systems, replication and majority voting are well-known and frequently employed consensus-building techniques. Current research focuses on safeguarding voting systems against malicious assaults.

Regarding our task, we must determine the diameter of the wireless network that has been deployed. The Secure Distributed Voting Protocol Timed-Buffer Distributed Voting Algorithm (TB-DVA) was utilized[(3)] . It is compared to various different distributed voting techniques in order to illustrate its unique contribution to fault tolerance and security, both of which are essential components of distributed systems. Once the system has stabilized, all nodes will broadcast the most distant node. Using a voting process, the node with the greatest distance from the other nodes is then determined.

Algorithm Diameter-voting ($Node_i$) provides a way for updating the max-count from the ith node. $Node_i$ recomputed the Max-Count for all of his reachable destinations according to the interval setting of 3 seconds, 5 seconds, or 10 seconds. Then, it compares the newly calculated number to the committed Max-Count value. The following algorithm describes subsequent steps:

2.2 Algorithm Diameter-voting ($Node_i$)

The diameter-majority algorithm is scheduled to wait for certain time after the diameter-voting algorithm is initiated. We set it for 2 Seconds. After that it checks all the values received for Max-Count and selects maximum of them and commits that value. This is explained in the following algorithm.

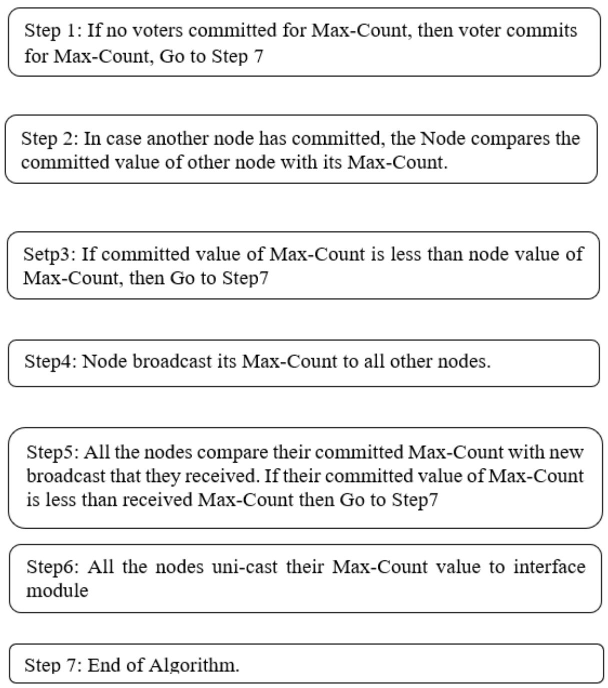


Fig 1. Algorithm 1

2.3 Algorithm Diameter-Majority

In this study, the network diameter will be determined by developing a packet scheduler. Once the network's diameter is determined, a set number of queues will be maintained. As described previously, the diameter-voting process is scheduled to run for a predetermined amount of time. This interval is defined by the duration of the entire simulation. If the simulation duration is 100 milliseconds, we specify intervals in multiples of ten, such as 20 milliseconds, 30 milliseconds, or 40 milliseconds, and at the end of each interval period, each node broadcasts its known neighbourhood distances to the voter node. This negates the impact of newly added or removed nodes in existing networks. Once, each node broadcasts its distance from the voting node. The diameter is decided using the majority vote technique, as described in the algorithms.

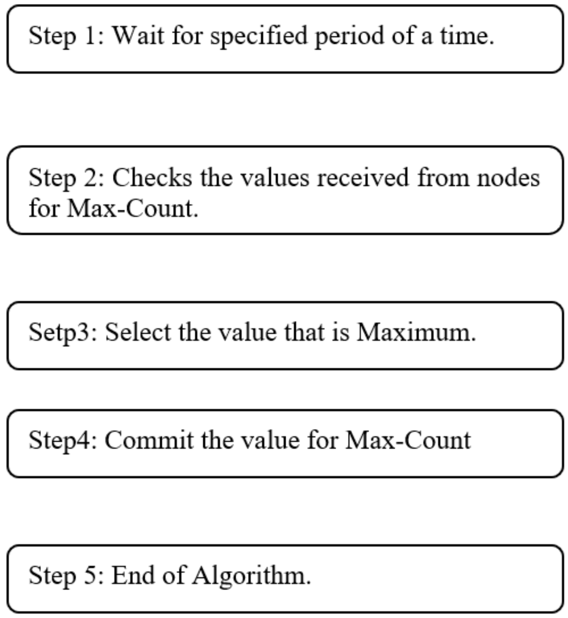


Fig 2. Algorithm 2

2.4 Packet scheduler for meeting the flows deadline

A packet-switched environment and an integrated service environment are two prevalent methods for addressing different service requirements. A path for a flow is booked in an integrated service environment by reserving the resources along the path. During packet scheduling, packets are prioritized based on service requirements, and intermediary nodes arrange them based on their respective priorities. When scheduling packets in delay-constrained systems such as wireless Adhoc networks, a basic scheduling strategy may not be sufficient to meet the flow deadlines for diverse flows. Instead, a combination of a scheduling algorithm and statistical multiplexing will provide an efficient technique for meeting flow deadlines.In this section, we show how to meet flow deadlines by statistically multiplexing packets at intermediate nodes.

We implemented our proposed packet scheduler on the nodes of the deployed wireless ad hoc network. This is referred to in future references as a "modified AODV deployment." The source achieves this by computing the deadline for individual packets using the deadline value for the corresponding flow. Because flow is designed as a logic unit that represents a series of packets, each packet within the flow uses the same deadline value. However, when their respective arrival times are considered, their respective deadlines may vary slightly. Here, it is assumed that the source is informed of the flow deadline at the outset. The intermediate switches then reorganize and schedule the packets by collecting them in ascending order of their associated deadline information. This is achieved by the switch maintaining a collection of virtual output queues (VOQs) for each input port. Numerous scheduling methods, including first-come, first-served (FCFS), round robin (RR), rigorous policy, and earliest deadline first, have been devised for packet scheduling (EDF). We chose EDF as the optimal scheduling method for real-time applications; therefore, that's why we used it.We demonstrate how each component of our proposed method operates.

2.5 Packet classifier

Priority is determined by the number of packets to be transmitted by the packet classifier. This might be deduced from Figure 3. According to the routing table for node 3, node 1 is two hops away from node 2, and node 2 is one hop away. This concept is given precedence. As indicated previously, we partition packets at intermediate nodes based on their diameter and distance to travel. Because we must keep the number of queues constant, we perform the modular arithmetic on the diameter as shown in the following section.

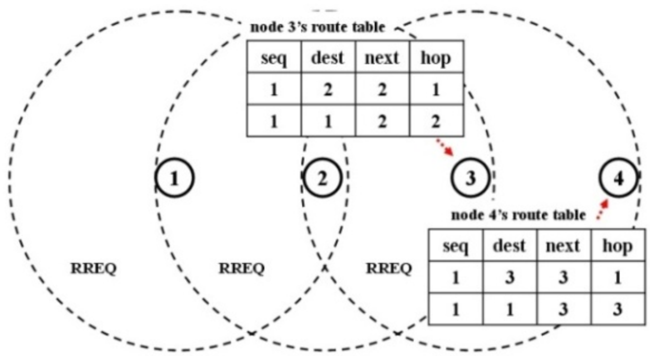


Fig 3. Packet classifier based on hop count

2.6 Scheduler

At intermediate switches, the classifier splits packets into distinct input queues based on their principal priority, the information level. There are two reasons for using multiple input queues. First, it is observed that the bandwidth of interconnection networks has increased during the past many years. However, the amount of packets utilized by wireless ad hoc network applications has remained relatively constant (especially control messages). The second cause is the blockage of the head of line (HOL). If the packet at the head of the queue's target output port is busy, subsequent packets in the queue will experience HOL blocking delays.

In formula (1), if the diameter of node$_i$ is'd' and the number of queues chosen is 'r,' then the number of hop ranges is d/r.

Each queue $_i$ will include packets with hop counts within the specified range

$$\frac{d}{r}\cdot i$$

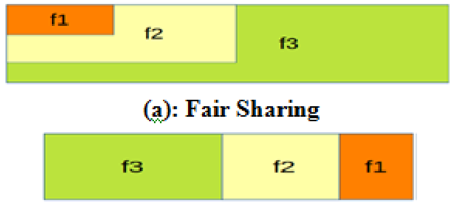
$$\left(\frac{d}{r}\cdot(i+1)\right)+1$$

For example, if the determined diameter is 16 and it is intended to maintain four queues at each node, then Queue0 will contain all packets at distances 1 to 4, Queue$_0$ will contain all packets at distances 5 to 8, and Queue$_1$ would contain all packets at distances 9 to 12. Queue$_2$ will include all packets between distances 13 and 16.Once packets are separated based on hop count, they are scheduled by providing the greatest priority to packets that need to travel the farthest, i.e., scheduling packets in Q $_{i+1}$ before Q$_i$ for all i. Referring back to our earlier illustration, packets in Queue$_4$ are scheduled first, followed by packets in Queue$_3$, Queue$_2$, Queue$_1$, and finally packets in Queue$_0$. Since we are scheduling longer-distance packets ahead of shorter-distance packets, we are minimizing the delay for longer-distance flows while perhaps increasing the delay for shorter-distance flows, thereby decreasing the average latency and enhancing quality of service (QoS).Using our second component, the diameter finding algorithm, we will determine the diameter of the network as part of the packet scheduler implementation. Once we have determined the network diameter, we will maintain a fixed number of queues. To do this, a modulus operation is performed on the diameter over the number of queues, thereby determining the hop length for each queue. Once the packets are segregated into the queues based on number of hops they need to travel, we will schedule the queues in such a way that, the packets need to travel more will be pushed ahead of the others.

In other words, packets in queue$_3$ are scheduled ahead of queue$_2$, which is scheduled ahead of queue$_1$ and so on. Doing this way we expect to meet the deadlines of the flow by scheduling packets need to travel most ahead of packets needed to travel less.

2.7 Mathematical Model

Let's consider there are n flows, f_1, f_2, $f_3 \ldots f_n$ and their corresponding flow sizes be λ_1, λ_2, $\lambda_3 \ldots \lambda_n$. We assume λ_i be flow length and we approximate it to the number of hops to make the quantitative estimate of the improvement in reducing the delay and there by meeting the deadline.



(a): Fair Sharing

(b): Earliest Deadline first

Fig 4. Demonstration of Fair sharing and EDF scheduling algorithm

We assume that flow and their flow length are increasing order of their magnitude, that is, $\lambda_1 < \lambda_2 < \lambda_3 < \ldots < \lambda_n$ and also assume that $\lambda_2 = \eta\ \lambda_1$, $\lambda_3 = \eta\ \lambda_2$, $\lambda_k = \eta\ \lambda_{k-1}$, and so on and finally $\lambda_n = \eta\ \lambda_{n-1}$ (meaning flow sizes are monotonically increasing their number of hops in their path).

This arrangement allows us to write any flow length $\lambda_k = (\eta)^{k-1}\lambda_1$.

Further, we assume that processing time at each queue is constant and is equal to δ.

Thus, we will evaluate the performance gain for the longest flow $f_n$with our proposed work.

1) In unconstrained scheduling: If we do not make use of any of the scheduling policy, then the time incurred in scheduling is proportional to the length of the flow and we could approximate the time incurred during the scheduling as shown in the Figure 4 (a)

Particularly for the flow f_n, delay incurred will be calculated as

$$d_n(\text{ Old }) = \sum_{i=1}^{n} \sum_{j=1}^{n-i+1} \delta \lambda_\text{k}$$

But, we use our proposed work, by bringing the longest flow, fn, we can reduce the second component of the summation, and thus incurred delay will be

$$d_n(New) = \sum_{i=1}^{n} \delta \lambda_\text{k}$$

As shown in Figure 4 (b), the deadline associated with the longest flow is reduced comparative to increase in delay for shorter flows.

Consider the dn(New), dn(Old), we can deduce that scheduling the longest flow ahead of shorter flows, we can reduce the delay associated with the longer flows. As can be shown it is only the parameter we can optimize in the routing. Further, it can be easily deducing the reduction in delays for the flows fn-1, fn-2 and so on.

2.8 Implementation

Using TB-DBA, we will determine the network's diameter in order to create a packet scheduler. Once we have determined the network diameter, we will maintain a fixed number of queues. To do this, a modulus operation is performed on the diameter over the number of queues, thereby determining the hop length for each queue. For instance, if we wish to maintain 5 queues and the width of the deployed network is 16, then each queue is designed to accommodate hop distances in the range of 3 (16/5 = 3). Specifically, packets 1 to 3 hops away will be placed in queue 1, packets 4 to 6 hops away will be placed in queue 2, packets 7 to 9 hops away will be placed in queue 3, packets 10 to 12 hops away will be placed in queue 4, and packets 13 to 16 hops away will be placed in queue 5. Although the last queues will serve as placeholders for the packets (with a number of hops in the range of 4 compared to the other queues' packets in the range of 3), we continue to do so to keep our implementation as simple as possible.

Once the packets are separated into queues based on the number of hops they need to traverse, we will schedule the queues so that the packets that require more hops will be prioritized. In other words, queue-5 packets are prioritized above queue-4 packets, which are prioritized over queue-3 packets, and so on. We expect to achieve the flow's deadlines by prioritizing the packets that must go the farthest above those that must travel the least.

3 Results and Discussion

Table 1. Simulation Model

Number of nodes	**50**
Simulation area	300 x 1500
Transmission power	7.5 mWatt
Simulation time	100 Seconds
Node speed time	10 - 90 Meters / Second
Node pause time	0 to 90 Seconds
Protocols Referred	AODV, Modified –AODV
Simulator Used	NS - 3.30

This work utilises NS3 for simulation; we modified the NS3 AODV implementation to incorporate our ideas. In future references to our work, we will refer to this modified version as Modified AODV. Because the code that runs to determine the diameter of the network consumes additional CPU cycles, we compare the CPU utilisation of the AODV and Modified AODV algorithms. In this work, we determined the diameter of the deployed wireless network. This was necessary for our network since we desired to achieve quality of service while accounting for delay. We prioritize long-distance flows over shorter-distance flows to reduce the delay caused by long-distance flows. Given that we cannot cover the full hop count space, we sought hop count-defined ranges. This was attained by expanding the diameter. As anticipated, the code that computes the diameter in modified AODV will require additional CPU cycles due to its separation from the algorithm that classifies and schedules packets based on the number of hops they must traverse. To determine the rise in CPU utilization in the improved AODV, we measured the increase in this metric in different circumstances. The results are encouraging since, despite the fact that improved AODV consumes more CPU cycles than the original implementation, these additional CPU cycles are minimal.We conclude by emphasizing that this increase in CPU utilization is modest compared to packet prioritization based on hop count. This reduces by a factor of the diameter or number of queues the overall workload involved with categorizing and prioritizing packets simply on the basis of hop count. Thus, if the diameter of the network is 16, the improved AODV utilizes only five priority levels as opposed to sixteen.

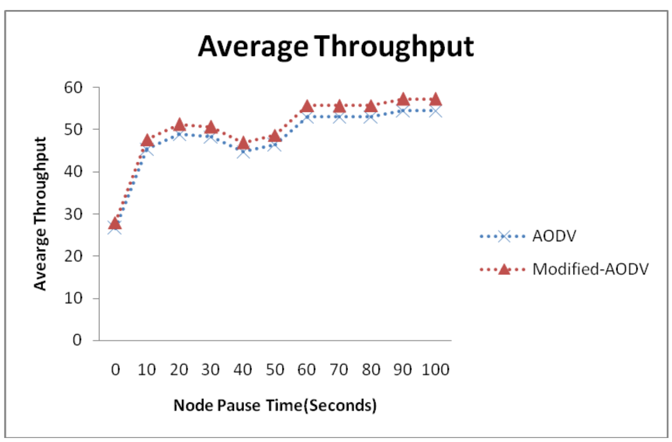


Fig 5. Throughput for 50 nodes with 10 Sources with constant node speed time 10 m/s

Positive results were observed for both throughput and delay. The number of dropped packets decreased since data transmissions prioritized the most distant destination. In modified AODV, the decrease in delay for longer flows was greater than the rise in delay for shorter flows, resulting in a reduction in average delay in comparison to AODV. This decrease in delay resulted in an increase in throughput.

Our primary focus is to reduce the average end-to-end delay and increase the throughput of the Adhoc Network by choosing its diameter. The utilization of replication and majority voting procedures is widely recognized and frequently implemented in order to attain fault tolerance in distributed systems. A secure public voting facility that is resistant to malicious attacks. In order to achieve this objective, the Timed-Buffer inquiry is now organizing the implementation of a secure distributed voting protocol

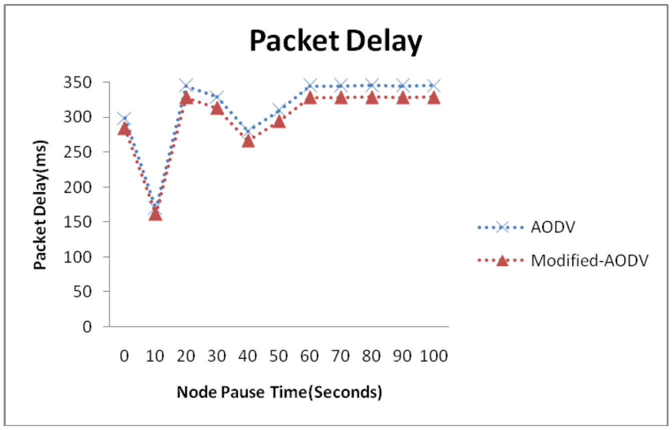


Fig 6. Delay for 50 nodes with 10 Sources with constant node speed time 10 m/s

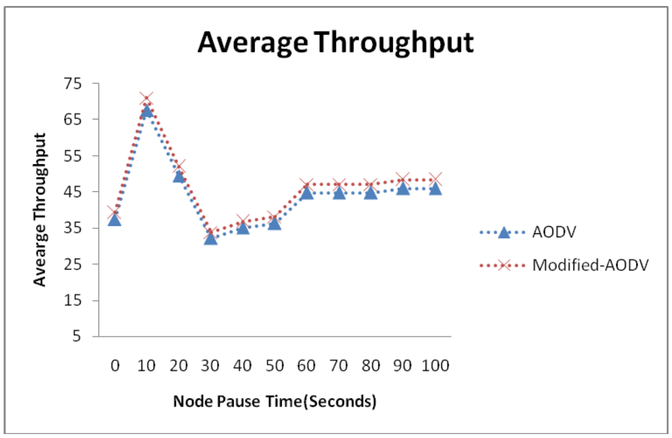


Fig 7. Throughput for 50 nodes with 20 Sources with constant node speed time 10 m/s

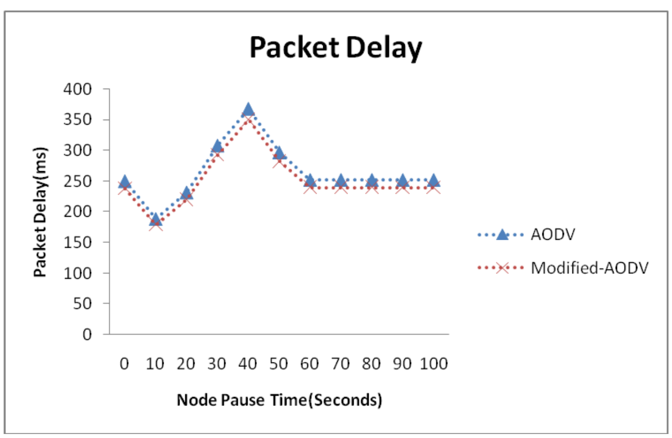


Fig 8. Delay for 50 nodes with 20 Sources with constant node speed time 10 m/s

known as the Timed-Buffer Distributed Voting Algorithm (TB-DVA). This study aims to compare the proposed technique with several distributed casting vote systems, emphasizing its distinct contributions to fault tolerance and security, which are critical requirements for ensuring system dependability. The algorithm was utilized to ascertain the overall diameter of the installed Adhoc network. In this paper, we propose a scheduling method and admission control policy for the IEEE 802.16b wireless access standard. The proposed system demonstrates practicality and adherence to the IEEE802.16 standard, since it effectively accommodates numerous traffic classes while ensuring quality of service (QoS). To the best of our knowledge, this algorithm represents the initial instance of its kind. Simulation experiments provide empirical evidence that the suggested system incorporates Quality of Service (QoS) support, hence guaranteeing that the majority of flow deadlines are met by the proposed solution. A correlation was shown between traffic characteristics, quality of service (QoS) needs, and network performance.

4 Conclusion

The diameter of the operational network is determined as part of this study's scope. Compared to the AODV standard packet scheduler policy, the new packet scheduler policy was found to provide high throughput while also having a lower average end-to-end latency. This was discovered through simulation. The discovery of this was aided by contrasting the two policies. The new algorithm delivers a 7.14 percentage point reduction in average end-to-end delay and a 9 percentage point improvement in average throughput when compared to the traditional AODV packet scheduler policy. Additionally, the new approach achieves a 9 percentage point increase in average throughput.

References

1) Karygiannis T, Owens L. Wireless Network Security 802.11, Bluetooth and Handheld Devices, NIST Special Publication. 2002. Available from: https://www.govinfo.gov/content/pkg/GOVPUB-C13-701c05fec7eec9b362e0dac65ba19f97/pdf/GOVPUB-C13-701c05fec7eec9b362e0dac65ba19f97.pdf.
2) Kalaivanan K, Idayachandran G, Vetrivelan P, Henridass A, Bhanumathi V, Chang E, et al. An Energy-Efficient T-Based Routing Topology for Target Tracking in Battery Operated Mobile Wireless Sensor Networks. *Sensors*. 2023;23(4):2162–2162. Available from: https://doi.org/10.3390/s23042162.
3) Hardekopf B, Kwiat K, Upadhyaya S. A Decentralized Voting Algorithm for Increasing Dependability in Distributed Systems. 2001. Available from: https://www.researchgate.net/publication/228557120_A_decentralized_voting_algorithm_for_increasing_dependability_in_distributed_systems.
4) Hamdy H, El-Sayed ZM, Hashem MA, Saleem. New Proposed Wireless Sensor Network Protocol for Internet of Things Applications", Sohag Journal of sciences. *Sohag Journal of Sciences*. 2023;2023(1):83–89. Available from: https://sjsci.journals.ekb.eg/article_259689_61739f300eeb80cdb9bbb307714c818a.pdf.
5) Baumann D, Mager F, Wetzker U, Thiele L, Zimmerling M, Trimpe S. Wireless Control for Smart Manufacturing: Recent Approaches and Open Challenges. *Proceedings of the IEEE*. 2021;(4):441–467. Available from: https://doi.org/10.48550/arXiv.2010.09087.
6) Obi E, Mammeri Z, Ochia OE. A Centralized Routing for Lifetime and Energy Optimization in WSNs Using Genetic Algorithm and Least-Square Policy Iteration. *Computers*. 2023;12(2):22. Available from: https://doi.org/10.3390/computers12020022.
7) Careglio D, Spadaro S, Cabellos A, Lazaro JA, Barlet-Ros P, Gené JM, et al. Results and Achievements of the ALLIANCE Project: New Network Solutions for 5G and Beyond. *Applied Sciences*. 2021;11(19):9130. Available from: https://doi.org/10.3390/app11199130.
8) Sudha1 M, Shanmugapriya P, Rami Q, Malik A, Alkhayyat. Adaptive Route Sink Relocation Using Cluster Head Chain Cycling Model in WSN. *Intelligent Automation & Soft Computing*. 2023;36. Available from: https://www.techscience.com/iasc/v36n2/51131/pdf.
9) Shukry S. Stable routing and energy-conserved data transmission over wireless sensor networks. *EURASIP Journal on Wireless Communications and Networking*. 2021;(1):36. Available from: https://doi.org/10.1186/s13638-021-01925-3.