SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V10I9P104

Volume 10 Issue 9, 28-37, September 2023

Original Article

Buffer-Aware Route-Finding Mechanism for Enhanced Communication in Infrastructure-less Wireless Networks

Mohammad Khaja Nizamuddin[1], D HariKrishna[2], Syed Raziuddin[3], M. Anusha[4], Arshad Ahmad Khan Mohammad[5]

[1,5]*Department of CSE, GITAM Deemed to be University, Hyderabad, India.*
[2]*Department of CAI, KKR & KSR Institute of Technology, Guntur, India.*
[3] *Department of CSE, Deccan College of Engineering and Technology, Hyderabad, India.*
[4]*Department of CSE, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Andhra Pradesh, India.*

[1]*Corresponding Author : mohammed.k.nizam@gmail.com*

Received: 06 July 2023 Revised: 18 August 2023 Accepted: 07 September 2023 Published: 30 September 2023

Abstract *- Wireless infrastructure-less networks are designed to provide access irrespective of geographical positioning, featuring key attributes such as mobility, heterogeneous resource utilization, peer-to-peer communication, adaptability, and self-forming capabilities. These networks find crucial applications in health management, military operations, and disaster relief, where efficient communication is imperative. The linchpin of communication in such networks lies in routing, which entails creating a path from source to destination and forwarding data in the form of packets. However, efficient routing becomes paramount when considering the constraints of limited buffer capacity within these networks. In this paper, we introduce a novel buffer-aware route-finding mechanism, leveraging the knapsack algorithm to optimize the packet processing ability of nodes based on their residual buffer size. The primary focus of this protocol is to mitigate the packet loss issue stemming from buffer overflow, a common challenge in infrastructure-less networks. We meticulously execute, evaluate, and validate our proposed mechanism using the NS-2 simulator. Our results unequivocally demonstrate the significant enhancement in network performance achieved through implementing our approach, notably reducing packet loss. This innovative buffer-aware routing solution holds immense promise for improving the reliability and efficiency of communication in infrastructure-less wireless networks, thereby contributing to the advancement of various critical applications.*

Keywords *- Infrastructure-less networks, Buffer-aware routing, Wireless communication, Knapsack mechanism, Packet loss mitigation.*

1. Introduction

The fundamental objective of wireless infrastructure-less networks is to establish ubiquitous internet connectivity, transcending geographical boundaries. These networks are characterized by many wireless devices equipped with diverse and heterogeneous resources seamlessly interconnected through radio communication. Communication within this network paradigm occurs directly between devices if they fall within each other's radio range; otherwise, they rely on intermediate nodes to facilitate communication.

Notably, widgets are free to enter and exit the network at will, resulting in a dynamic and unpredictable network topology. These intrinsic attributes render infrastructure-less networks self-forming, adaptable, and autonomous, making them efficient and cost-effective in terms of implementation time. The versatility of infrastructure-less networks finds applications in critical domains such as military operations, disaster recovery efforts, and health management. Given the sensitivity and demand for efficient communication in these contexts, it is imperative to address the unique challenges posed by such networks. Communication within this network architecture hinges on the core processes of route computation and subsequent packet forwarding along the computed path. While the primary objective of routing is to add a viable approach, it must also adhere to the quality of service requirements dictated by specific applications.

In the context of wireless infrastructure-less networks, designing efficient routing paths becomes an intricate challenge due to the network's inherent characteristics, making it a thriving area of research [1]. Buffer awareness emerges as a pivotal factor during route computation, considering the decentralized nature of the network, the absence of a central coordinator, and the presence of resource-constrained devices [2-4]. In the data forwarding phase of the routing mechanism, packets are temporarily buffered in intermediate nodes before proceeding along their path. The risk of packet loss looms when the node buffer reaches its capacity or when packets remain in the buffer for a duration exceeding their defined lifetime. Therefore, the efficient selection of intermediate nodes during route computation becomes paramount. The literature presents many routing mechanisms to mitigate packet loss attributed to buffer constraints and minimize packet delay within node buffers. These mechanisms are often categorized into congestion control and non-control routing, with the latter being the prevalent choice for contemporary wireless infrastructure-less networks [5].

This paper presents a novel and highly efficient routing protocol tailored to minimize packet loss from constrained buffers in wireless infrastructure-less networks. Our route-finding metric revolves around optimizing the packet processing capacity of nodes concerning their residual buffer space. We address the critical issue of packet loss attributed to buffer overflow, a prevalent concern in such networks. The remainder of this paper unfolds as follows: Section 2 delves into the intricacies of buffer-aware routing in wireless infrastructure-less networks. Section 3 elucidates the methodology for determining a node's buffer-specific optimized packet processing capability. Section 4 outlines our performance evaluation criteria and presents the outcomes of our comprehensive analysis. Finally, we conclude with a summary of our contributions and offer recommendations for future research directions.

2. Buffer-Aware Routing in Wireless Infrastructure-less Network

Wireless infrastructure-less networks represent a dynamic and decentralized communication paradigm characterized by multi-hop, peer-to-peer connectivity. In these networks, intermediate nodes take on the role of routers, enabling communication between nodes that are out of direct radio range. Within this context, the management of node buffer space emerges as a critical factor profoundly influencing network performance. The effectiveness of these networks in terms of packet delivery and reduced latency hinges on the status of an intermediate node's buffer space. If a node's buffer can comfortably accommodate incoming packets, it positively impacts network performance. Conversely, buffer congestion due to occupied space can lead to packet loss, posing challenges to seamless communication.

Unlike traditional infrastructure-based networks, where congestion management occurs primarily at higher OSI layers, such as transport layer protocols, wireless infrastructure-less networks face unique challenges. These networks lack a centralized infrastructure, making the application of traditional congestion control approaches impractical. Consequently, congestion can manifest at any intermediate node during data transmission, resulting in increased packet loss, delayed network responses, and compromised performance.

In wireless infrastructure-less networks, effective communication between nodes relies on intermediate nodes maintaining buffers. These buffers briefly hold packets to ensure smooth synchronization between communicating entities. The internal structure of a node's buffer consists of several key elements, as shown in Figure 1.

1. Input Buffer Queue: Incoming packets are placed in the input buffer queue upon arrival at the input interface. They remain there until they reach the edge of the queue, at which point a relevant judgment is made.
2. Decision Module: A decision module selects a packet from the input buffer queue and evaluates it based on the requirements of the underlying routing protocol.
3. Output Buffer Queue: Processed packets are then placed in the output buffer queue, where they await their turn for transmission following MAC protocol instructions.

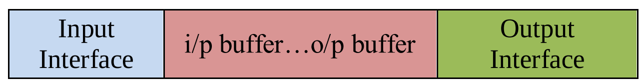

Fig. 1 Node buffer's internal structure

Congestion in wireless infrastructure-less networks occurs when the incoming packets' rate exceeds the input buffer's processing capacity. In such instances, a queue forms in the input buffer, potentially leading to packet loss if the total is exceeded. The routing protocol's route discovery metric often reflects this congestion condition.

When an intermediate node's resources fall below a specific threshold, it becomes a bottleneck, affecting the network's reliability, performance, energy consumption, and overall lifespan. In wireless infrastructure-less networks, an intermediate node assumes the role of a "bottleneck" when it cannot efficiently manage the incoming data traffic due to factors like limited processing capacity, constrained buffer space, or an overwhelming influx of data packets. When an intermediate node reaches this bottleneck state, it struggles to handle the incoming traffic effectively, resulting in packet loss, delays, and a significant deterioration in network performance.

Essentially, a bottleneck intermediate node disrupts the seamless data flow within the network, adversely affecting its overall reliability and operational efficiency, making it a critical concern in network optimization and management. Node 4 becomes the bottleneck intermediate node in the network, as shown in Figure 2.

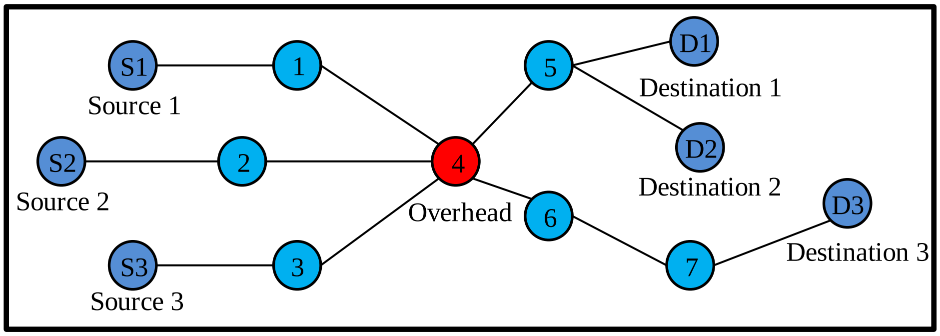

Fig. 2 Node 4 becomes the bottleneck intermediate node

Performance evaluations were conducted to assess the efficacy of existing proactive and reactive routing mechanisms in dense wireless infrastructure-less network environments. Findings indicate that protocols based on distance vector routing tend to redirect excessive traffic to specific node buffers, resulting in a high incidence of packet loss. Moreover, the route-finding metrics failed to prevent intermediate nodes from becoming bottlenecks in congested scenarios.

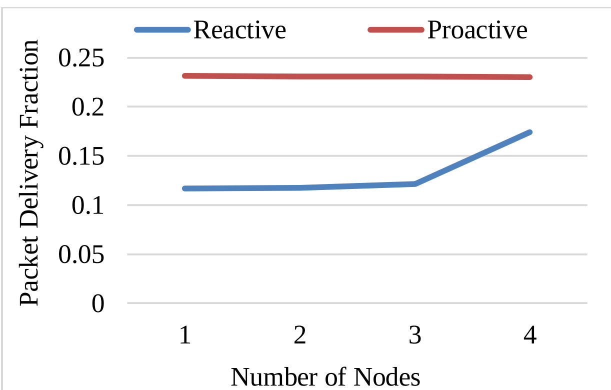

Fig. 3 Packet delivery fraction in wireless infrastructure less networks during bottleneck situation in network

Figures 3 and 4 present the outcomes of our comprehensive performance evaluation concerning proactive and reactive routing mechanisms within a densely populated wireless infrastructure-less network. Our results unequivocally demonstrate that the employed protocols exhibit a significant packet loss issue, primarily attributable to their reliance on distance vector routing. These protocols divert substantial network traffic toward a specific node's buffer, contributing to elevated packet losses. Remarkably, the route-finding metric embedded in these protocols proves ineffective in preventing specific nodes from assuming the bottleneck role within the network.

Figure 3 illustrates the packet delivery percentage of proactive and reactive distance vector-based routing protocols within a busy, congested network environment, with one of the intermediate nodes acting as a bottleneck.

Furthermore, Figure 4 vividly portrays the extent of packet loss incurred by a proactive and reactive routing system based on distance vectors in a congested network scenario, wherein one of the intermediary nodes assumes the role of a bottleneck node, reaffirming the pressing need for innovative routing solutions.

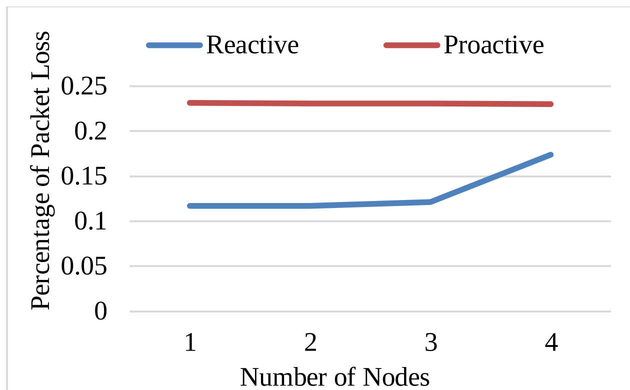

Fig. 4 Percentage of packet loss in wireless infrastructure less networks during bottleneck situation in network

Central to enhancing network performance is carefully considering an essential element: a node's buffer space [5]. In Mobile Ad Hoc Networks (MANETs), buffer overflow is one of the most prevalent culprits behind packet loss, stemming from the burdensome traffic loads exerted on node buffers. Traditional congestion control techniques, primarily deployed at the transport layer, are devised to regulate buffer overflow. However, the distinctive characteristics of wireless infrastructure-less networks render applying such techniques impractical. Hence, a proactive strategy for detecting and controlling congestion is imperative. This issue is closely intertwined with medium access and packet forwarding, aggravated by the surge in traffic volume. A well-designed routing protocol becomes indispensable to address this challenge effectively, as depicted in Figure 5. The figure elucidates the underlying causes of packet drops from intermediate nodes within the network, emphasizing the importance of network-layer congestion management through buffer-aware routing protocols.

Routing protocols can be categorized into two main groups: congestion control routing and congestion non-control routing. Most routing mechanisms employed in wireless infrastructure-less networks fall into the latter category. Unlike established networks, the dynamic nature of wireless infrastructure-less networks incurs higher overheads and delays, making complete congestion prevention challenging. In light of these constraints, the Internet Engineering Task Force (IETF) has recommended the utilization of average queue size computations for computation and monitoring purposes, aiming to alleviate congestion and consequent packet loss. In pursuit of this goal, we have devised an innovative buffer-aware route-finding mechanism, leveraging the optimization prowess of the knapsack algorithm [6]. Our protocol strategically addresses the packet loss stemming from buffer overflow by dynamically adapting to the residual buffer space of individual nodes, thereby mitigating congestion-induced performance degradation and ensuring more efficient network communication.

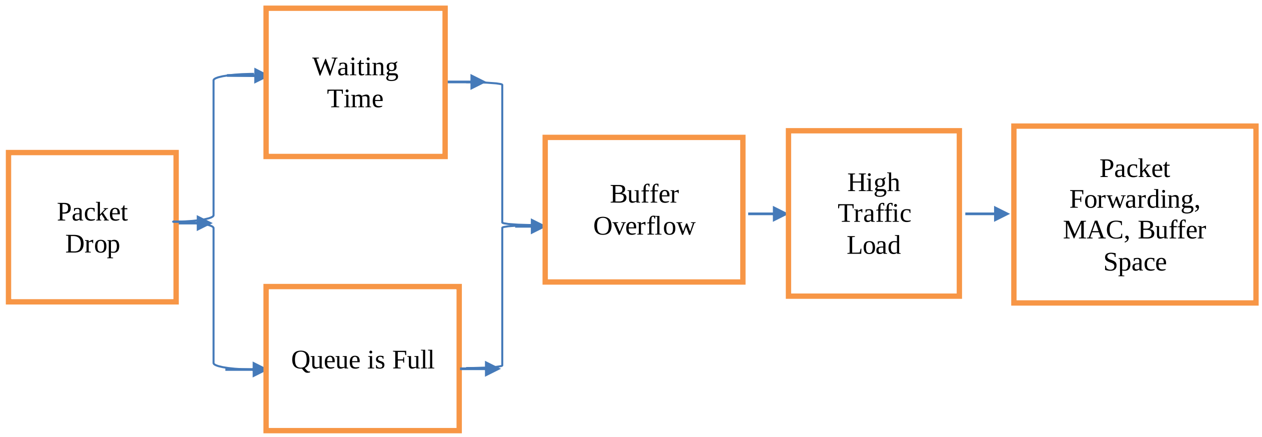

Fig. 5 Reason behind the packets drop from intermediate nodes

Recognizing the importance of node buffer space in enhancing network performance, we address the issue of buffer overflow, a common cause of packet loss in Mobile Ad Hoc Networks (MANETs). While traditional congestion control techniques primarily function at the transport layer, these approaches are less suited to the unique characteristics of wireless infrastructure-less networks.

As a result, we propose an innovative buffer-aware route-finding mechanism, incorporating the optimization technique of the knapsack algorithm. This mechanism proactively identifies and addresses potential congestion issues, reducing packet loss caused by buffer overflow.

Our route-finding metric is intelligently tailored to consider each node's residual buffer capacity, contributing to more efficient network communication. Congestion control is a vital aspect of optimizing the performance of wireless infrastructure-less networks. While routing protocols are typically categorized into congestion control and non-control groups, the latter is more prevalent in dynamic wireless networks.

We propose a novel buffer-aware route-finding mechanism to mitigate congestion and reduce packet loss, leveraging the knapsack optimization technique. By proactively managing buffer space and addressing congestion at the network layer, our approach offers a promising solution to enhance communication reliability, reduce packet loss, and improve the overall efficiency of wireless infrastructure-less networks.

3. Optimal Packet Processing Capacity (OPPC)

In the context of optimizing packet processing within wireless infrastructure-less multi-hop networks, where mobile nodes are equipped with buffers of size Q bytes, and communication occurs through packet-based exchanges, several critical conditions must be met to determine a network node's Optimal Packet Processing Capacity (OPPC). These conditions are pivotal in ensuring seamless communication without incurring packet loss due to buffer constraints:

1. Packet Volume Constraint: The total volume of communication packets handled by a node must remain within the confines of its buffer size, Q bytes. This condition safeguards against buffer overflow, which can lead to packet loss.
2. Maximizing Packet Processing: The network strives to process as many packets as possible within the capabilities of each node. This involves making efficient use of available resources to maximize packet throughput.
3. Strategic Packet Handling: To achieve optimal communication, nodes are tasked with intelligently selecting which packets to process and which to disregard. Ignoring a substantial portion of incoming packets is essential to manage congestion effectively.

In this network architecture, intermediary nodes (I_n) serve as pivotal points through which packets (P_i) must pass during multi-hop communication. These intermediary nodes are equipped with buffers of size Q bytes, and packets are temporarily stored within these buffers as they traverse the network. These three fundamental conditions are carefully considered and addressed to calculate the node's optimal packet processing capacity.

The network can achieve efficient, low-loss communication by adhering to these conditions while effectively managing buffer limitations. In the realm of wireless infrastructure-less networks, particularly within intermediary nodes (I_n), a sophisticated mechanism is employed to gauge and manage the flow of incoming and outgoing packets. This mechanism involves calculating averages using exponentially weighted moving averages, leading to essential insights for congestion control. Here's how it works:

3.1. Average Packet Arrival Rate ($\boldsymbol{R}$)

The average number of packets (R) that arrive at an intermediate node during a specific time period (t) is determined using an exponential weighted moving average formula:

$$R = \alpha R\_c + \beta R\_p \tag{1}$$

Here, R_c represents the average number of packets received during the previous time intervals, and R_p represents the average for the current time interval. The parameters α and β are weighted constants with values ranging from 0 to 1. This calculation provides a dynamic assessment of packet arrival rates.

3.2. Average Packet Transmission Rate (T)

Similarly, the average number of packets sent from a node during a certain time period is computed using the same exponential weighted moving average mechanism:

$$T = \alpha T\_c + \beta T\_p \tag{2}$$

Here, T_c signifies the average packets sent during the previous time intervals, and T_p represents the average for the current time interval, just like in the case of R, the parameters α and β play a crucial role in calculating this transmission rate. In network congestion management, comparing these rates, R and T, is pivotal. When a node's packet receiving rate (R) surpasses its packet transmission rate (T), congestion begins to develop. This situation implies that more packets are arriving at the node's buffer than it can transmit, potentially leading to packet drops and congestion-related issues. Conversely, when a node's packet receiving rate remains lower than its transmission rate, it can effectively process and transmit packets without drops, as the buffer can accommodate the incoming flow. This equilibrium ensures efficient packet flow through the node's buffer, mitigating congestion concerns. The intermediate nodes in infrastructure-less wireless networks encounter a significant obstacle due to their limited buffer size. In contrast to scenarios with indefinitely large buffers, these nodes are limited by their buffer capacity, which can result in packet loss and network congestion. Queuing theory addresses this issue, with equations 3 and 4 playing a crucial role in quantifying the buffer queue size, representing the number of packets held in the buffer during a specific period.

3.3. Packet Queue Calculation (Q_p)

Equation 3, representing Q_p, calculates the packet queue size. The determination of a particular outcome is contingent upon the ratio between the average rate at which packets arrive (R) and the disparity between the average rate at which packets are transmitted (T) and the arrival rate (R).

$$Q\_p = R \,/\, (T - R) \tag{3}$$

The equation mentioned above quantifies the accumulation of packets within the buffer, taking into account the arrival and transmission rates.

3.4. Queue Size (Q_S)

Equation 4, represented as Queue Size (Q_S), computes the packet queue size in bytes. It is obtained by multiplying Q_p by a constant factor (k).

$$Queue\ Size\ (Q\_S) = Q\_p * k\ bytes \tag{4}$$

This calculation helps understand the buffer occupancy in bytes, vital for congestion management. In practical terms, an intermediate node (I_n) within the network typically has a finite buffer capacity ($Q\ bytes$). To efficiently handle the flow of packets from various sources within a given time interval, it must adhere to the following condition: the communication packets arriving at the intermediate node's buffer must either be entirely processed and transmitted or discarded.

Partially processing packets is not viable, as it can lead to inefficiencies and potential congestion issues. By precisely calculating the queue size and understanding the limits of the buffer capacity, intermediate nodes can make informed decisions about how many packets to process and transmit, ensuring that the network operates optimally while avoiding packet drops and congestion to the extent possible.

Indeed, here's a breakdown of how to determine the optimal number of packets (n) to process through an intermediate node with a limited buffer capacity (Q) effectively.

3.5. Packet Selection and Buffer Management

In the context of an intermediate node with a buffer capacity of Q bytes, there are multiple incoming packets, denoted as $P(1,)P(2,) \ldots \ldots \ldots P(n)$.

Each of these packets has a specific size, represented as (x) bytes, and occupies an amount of buffer space (Q_s) within the node's buffer. Equation 5 calculates the remaining buffer size (Q_res) after forwarding a packet (i).

$$Q\_res = Q - Q\_s \tag{5}$$

This equation shows how much buffer space is left after a packet has been processed and forwarded.

3.6. Optimal Packet Selection

To determine the optimal number of packets (n) to process through the intermediate node's available buffer (Q), you need to choose a subset of packets (T) from the set $\{1, 2, 3, 4, 5, \ldots, n\}$. This subset represents the packets that the intermediate node will process. The goal is to maximize the sum of the sizes of these selected packets ($\sum (P\_i)(i \in T)$ while ensuring that the sum of the buffer space they occupy ($\sum Q\_s$ ($P_$i) ($i \in T$)) does not exceed the available buffer capacity (Q).

In other words, to find the combination of packets from the set that maximizes the utilization of the available buffer space without exceeding its limit. This selection process ensures efficient use of the intermediate node's buffer while preventing congestion and packet drops. The key is to balance processing as many packets as possible (to maximize network efficiency) and managing the buffer space effectively to prevent overflow or congestion.

This optimization process helps improve the performance of wireless infrastructure-less networks, especially in scenarios where buffer capacity is limited. Certainly, break down the cycle of using the knapsack algorithm to calculate the Optimal Packet Processing Capacity (OPPC) for a given residual buffer size in a wireless infrastructure-less network:

3.7. Knapsack Algorithm Setup

- The knapsack algorithm finds the optimal combination of packets to process within the available residual buffer size (Q). This algorithm generates a two-dimensional array (L) with dimensions from $0\ to\ n$ (representing the packets) and $0\ to\ Q$ (representing the buffer space).
- The algorithm considers whether to include each packet (indexed as i) in the set of packets to be processed. The objective is to maximize the total size of the selected packets while staying within the buffer space limit.

3.7.1. Initialization

Initialize the L array as follows:

- $L[0, Q\_s\ (P\_i) \leq 0] = 0$
  $for\ all\ possible\ values\ of\ i\ and\ Q\_s\ (P\_i) \leq Q$
- $L[0, Q\_s\ (P\_i) > 0] = -\infty$
  $for\ all\ possible\ values\ of\ i\ and\ Q\_s\ (P\_i) > 0$

These initializations set up the boundary conditions for the knapsack algorithm. Any attempt to include a packet in the solution that would exceed the available buffer space is considered an illegal condition and assigned a value of -∞.

3.8. Calculating Optimal Packet Processing Capacity (OPPC)

The algorithm's core calculates the Optimal Packet Processing Capacity (OPPC) for each combination of packets and buffer sizes.

For each packet $i\ (1 \leq i \leq n)$ and each possible buffer size $Q\_s\ (P\_i)$ $(0 \leq Q\_s\ (P\_i) \leq Q)$, the algorithm calculates $L[i, Q\_s\ (P\_i)]$ based on the recurrence relation:

$$L[i, Q\_s\ (P\_i)] = \max(L[i-1, Q\_s\ (P\_i)], L\_i + L[i-1, Q - Q\_s\ (P\_i)])$$

- L_i represents the size of packet i.
- The max function selects the optimal value for $L[i, Q\_s\ (P\_i)]$ among the options:
- The value without including packet i $(L[i - 1, Q\_s\ (P\_i)])$.
- The value of including packet i (L_i) plus the optimal value of the remaining buffer space $(L[i - 1, Q - Q\_s\ (P\_i)])$.
- This calculation is performed for all valid combinations of packets and buffer sizes.

3.9. Keeping Track of Packet Selection

To determine which packets are selected as part of the solution, a Boolean auxiliary array called $Keep[i, Q\_s\ (P\_i)]$ is used. It keeps track of whether an active node chooses to send packet i from the communication array $(Keep[i, Q\_s\ (P\_i)] = 1)$ or not $(Keep[i, Q\_s\ (P\_i)] = 0)$.

3.10. Result

The final result is the value of $L[n, Q]$, which represents the Optimal Packet Processing Capacity (OPPC) for the given residual buffer size Q. This algorithm efficiently determines how many and which packets should be processed by an intermediate node within its available buffer space, ensuring an effective and optimized use of resources while preventing congestion and packet drops in a wireless infrastructure-less network.

Certainly, Algorithm 1 is an effective algorithm designed to compute the node's optimal packet processing capacity, ensuring that intermediate nodes in a wireless infrastructure-less network can efficiently manage their buffer space and prevent packet drops caused by buffer constraints. Here's a detailed breakdown of the algorithm:

Algorithm 1: Computation of optimal packet processing capacity of node in its residual buffer

Input
- l: The number of packets (l ≤ n).
- Q_s (P_i): Buffer space occupied by each packet (0 ≤ Q_s (P_i) ≤ Q).
- n: The total number of packets.
- Q: The available residual buffer size.

Initialization
- Initialize a two-dimensional array L with dimensions $[n+1][Q+1]$ to store intermediate results.
- Initialize a Boolean array. Keep track of packet selection.
- Initialize $K = E$ (a variable for the optimal packet processing capacity).
- Set $L[0][Q\_s\ (P\_i)] = 0$ for all $Q\_s\ (P\_i)$ from 0 $to\ Q$.

Main Loop
- Iterate through packets (i = 1 to n).
- For each packet and buffer size (Q_s (P_i) = 0 to Q):
- Check if including packet i would not exceed the available buffer space and if including it results in a better value for L[i][Q_s (P_i)].
- If the conditions are met, update L[i][Q_s (P_i)] and set Keep[i][Q_s (P_i)] to 1 (indicating packet i is included).
- Otherwise, retain the previous value of L[i][Q_s (P_i)] and set Keep[i][Q_s (P_i)] to 0 (indicating packet i is not included).

Backtracking
- Start from the last packet (i = n) and the residual buffer size Q.
- Trace back through the Keep array to identify which packets were selected.
- Output the packet index if Keep[i][Q_s (P_i)] is 1, subtract Q_s (P_i) from the residual buffer size Q, and repeat for the previous packet (i-1).
- Continue until you reach i = 1.
- Return the final value of L[n][Q] as the Optimal Packet Processing Capacity (OPPC).

Algorithm 1 efficiently calculates the OPPC, ensuring that intermediate nodes can manage their buffer space effectively and prevent packet drops caused by buffer constraints. This algorithm helps maintain network performance and minimize congestion in wireless infrastructure-less networks by selecting an optimal combination of packets to process.

Algorithm 2, as described, provides a mechanism for determining whether a network node should actively participate in routing based on its computed optimal packet processing capacity (OPPC) concerning threshold values. This algorithm helps manage network congestion by selectively involving nodes in the routing process. Here's a detailed explanation of Algorithm 2:

Algorithm 2: Technique for determining the residual position of the intermediate node

Input
- OPPC: The node's computed optimal packet processing capacity.
- OPPC_min: The minimum threshold value for OPPC.
- OPPC_max: The maximum threshold value for OPPC.

Procedure
1. Determine the node's OPPC using Algorithm 1, as computed based on its residual buffer size and packet characteristics.
2. Evaluate the OPPC value against the defined threshold values.
3. If OPPC is greater than or equal to OPPC_max, the node has sufficient buffer capacity to participate in routing actively. In this case, mark the node as actively engaged in routing.
4. Otherwise, if OPPC falls within the range between OPPC_max and OPPC_min, the node can serve as a standby node, ready to participate if needed but not actively routing traffic.
5. If OPPC is less than OPPC_min, it suggests that the node does not have adequate buffer space to handle routing tasks effectively. In this case, the node should not be involved in route computation.

4. Performance Analysis

The proposed algorithm aims to prevent packet drops and congestion in the network by dynamically determining the involvement of nodes in routing based on their buffer capacity. By using the computed OPPC and comparing it to defined thresholds, the algorithm adapts to varying network conditions, ensuring efficient resource utilization.

Nodes with higher OPPC values actively participate in routing, while nodes with lower OPPC values act as standby or avoid routing altogether, optimizing network performance and resource allocation.In summary, Algorithm 2 offers a dynamic approach to managing node participation in routing, helping to maintain network efficiency and minimize congestion by considering the capacity of individual nodes' buffers. The proposed model for preventing packet-dropping nodes from affecting the routing process considers various intermediary nodes with different properties. Additionally, other network packets are considered, along with the size and typical buffer size used by nodes to process packets. Threshold values for $OPPC\_max$ and $OPPC\_min$ are calculated based on network sensitivity and initial network conditions. Here are some threshold values for the suggested methods in the analytical model:

1. Threshold for OPPC_max: This threshold represents the maximum allowable OPPC value a node's buffer capacity should meet to participate in routing actively. It is calculated under buffer = Q/2, which implies that the node's buffer is half full. This condition is applied to assess how nodes with moderately loaded buffers should be considered for routing.
2. The OPPC_min threshold indicates the minimum OPPC value required for a node to serve as a standby node in the routing process. It is calculated under Buffer = Q/4, which means the node's buffer is one-fourth full. Nodes meeting this threshold can assist in routing if needed but are not actively routing traffic.

These threshold values are essential for adapting the routing process to varying network conditions and ensuring efficient resource utilization. By considering different buffer levels and their impact on OPPC, the model aims to balance utilizing available resources effectively and preventing packet drops and congestion in the network. These thresholds provide a flexible mechanism for managing node participation in routing and optimizing network performance, especially in scenarios with varying traffic loads.

Table 1. Intermediate nodes with their attributes

Parameter	Node-1	Node-2	Node-3
Energy	7 j	6 j	5 j
Receiving Power	300 MW	300 MW	300 MW
Transmitting Power	600 MW	600 MW	600 MW
Buffer Size	2560 bytes	3584 bytes	5120 bytes
R	45	49	34
(T)	0.02	0.016	0.02

Threshold values

$$OPPC_{max} = 625\ Kbytes$$
$$OPPC_{min} = 312.5\ Kbytes$$
$$Lifetime = 200\ ms$$
$$Q = Buffer\ size\ of\ node$$

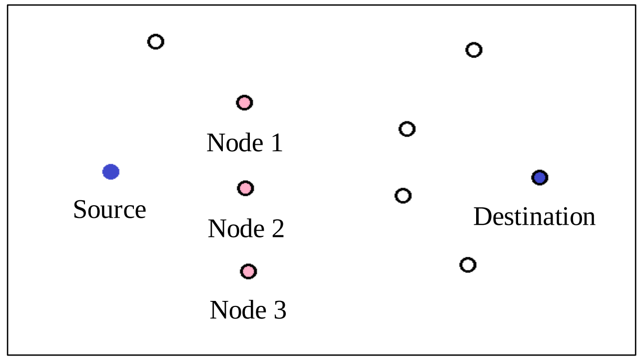

Fig. 6 Network with source and three intermediate nodes

Figure 6 illustrates a portion of the network featuring a source node and three intermediate nodes labelled nodes 1, 2, and 3. This depiction allows for a closer examination of each intermediate node's specific characteristics, which are integral to understanding the suggested algorithm. Some of the critical attributes of each intermediate node include:

1. Input and Output Packet Arrival Rates: These rates determine how quickly each node receives and transmits packets. They are crucial for assessing the node's processing capabilities and role in the routing process.
2. Buffer Capacity: The buffer capacity of each intermediate node signifies the maximum number of packets it can store at any given time. It's a vital factor in preventing congestion and packet loss within the node.
3. Battery Life: Battery life is critical, especially in wireless networks. It reflects how long the node can operate without recharging or replacing batteries. This information is essential for long-term network sustainability.
4. Power Transmitted and Received: The power levels for transmitting and receiving signals are essential for assessing the node's energy consumption and ability to communicate with neighbouring nodes.

The suggested algorithm can make informed routing decisions by considering these characteristics for each intermediate node. It can optimize packet processing based on buffer capacity, consider energy efficiency based on battery life, and ensure that nodes operate within their processing capabilities. This comprehensive approach helps in enhancing network performance and reliability. Intermediate nodes with their attribute values are shown in Table 1.

Table 2 indicates that node two is selected by your method as an intermediate node because it has lower traffic in its buffer. This choice is made because node 2 has an adequate buffer size, the shortest packet waiting time in its buffer, and its packet processing capability surpasses the threshold value.

This demonstrates the effectiveness of your algorithm in selecting intermediate nodes based on these critical factors.

Table 2. Calculated optimized residual status of the intermediate nodes

Node	Residual Buffer	OPPC	Packet Waiting Time
Node-1	1088 bytes < B_i	620 Kb< $OPPC_{max}$	0.0625 < TTL
Node-2	4608 bytes < B_i	875 Kb> $OPPC_{max}$	0.2 = TTL
Node-3	2280 bytes <B_i	750 Kb> $OPPC_{max}$	0.0909 < TTL

A network simulator [10] was employed to evaluate the mechanism's performance further. The simulation involved 200 nodes with varying resources distributed within a radio communication range of 1000 m*1000 m. The nodes determined their routes using an existing distance vector routing protocol, and the source node transmitted packets using the CBR application.

The simulation parameters are detailed in Table 3, and performance metrics included throughput, packet loss, and packet delivery fraction. The results, as depicted in Figures 7-10, were compared to those achieved using residual buffer-aware routing [12]. These comparisons provide valuable insights into the effectiveness of your proposed mechanism in enhancing network performance and reducing packet loss.

Table 3. Simulation parameters

Network Parameters	Value
Radio Area	100-300 m
Noses	200
Simulation Time	100 s
Mobility	10-40 M/s
Routing	Distance Vector, Residual, Optimized
Communication	Two Ray Ground
Energy	100 j
Traffic	CBR
Communication Area	1000 m*1000m

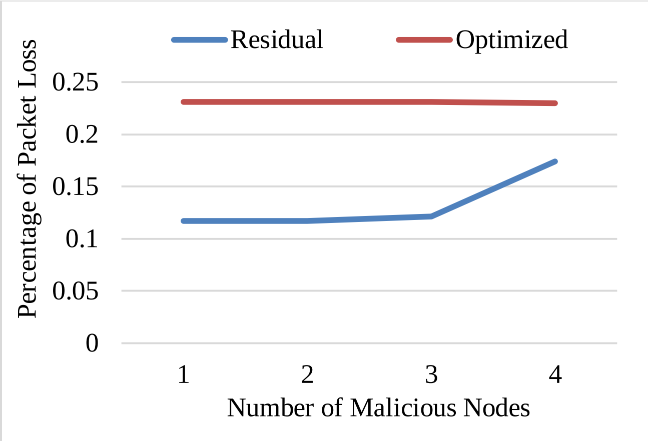

Fig. 7 Packet deliver fraction of the optimized buffer aware routing in comparison with the residual buffer aware routing

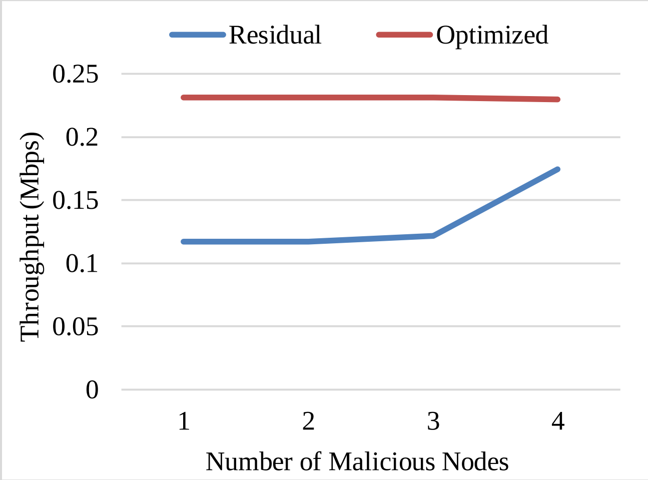

Fig. 8 Throughput of the optimized buffer aware routing in comparison with the residual buffer aware routing

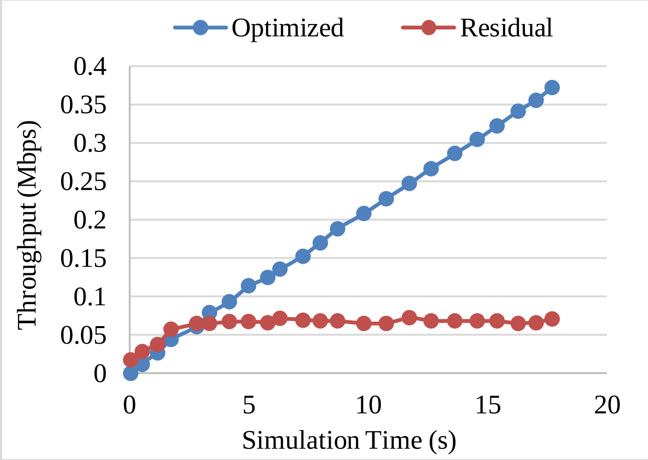

Fig. 9 Throughput of the optimized buffer aware routing in comparison with the residual buffer aware routing

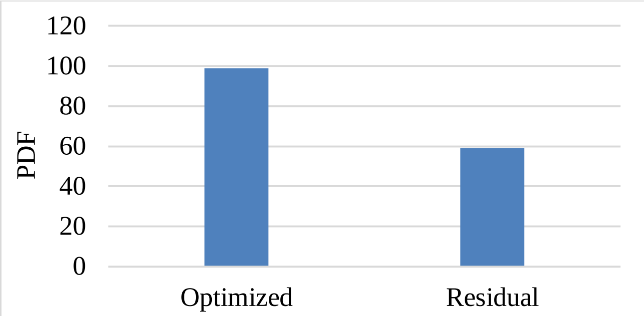

Fig. 10 Overall packet delivery fraction comparison proposed work with the residual buffer aware routing

The performance results in Figures 7-10 prove that your proposed mechanism has substantially improved the network's packet delivery and throughput performance. Our approach has effectively enhanced network efficiency by addressing the issue of packet-dropping nodes due to buffer overflow.The proposed method leverages the node's optimized packet processing capability as a metric for route computation, ensuring that intermediate nodes do not drop packets due to buffer constraints. This enables the identification of nodes prone to dropping packets due to buffer limitations, allowing them to be excluded from future communications. This indirect enhancement of network performance is particularly valuable in peer-to-peer wireless infrastructure-less networks, where each node's contribution is crucial to overall network efficiency and reliability.

5. Conclusion

Wireless infrastructure-less networks enable access in diverse geographical locations characterized by mobility, heterogeneous resources, adaptability, self-formation, and peer-to-peer communication. These networks support a wide range of applications that require efficient communication, making the design of efficient routing protocols, especially concerning buffer management, essential.

In this context, we have introduced a novel buffer-aware route-finding mechanism based on the knapsack algorithm. This mechanism effectively addresses the problem of packet loss resulting from buffer overflow in wireless infrastructure-less networks. Our proposed mechanism has been implemented, evaluated, and validated using the NS-2 simulator. The results demonstrate that our approach significantly enhances network performance by reducing packet loss, contributing to more reliable and efficient communication in these networks.

References

[1] Mona N. Alslaim, Haifaa A. Alaqel, and Soha S. Zaghloul, "A Comparative Study of MANET Routing Protocols," *The Third International Conference on e-Technologies and Networks for Development (ICeND2014)*, pp. 178-182, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[2] R. Sujee, and K.E. Kannammal, "Routing Protocols Based on Network Structure in Wireless Sensor Networks - A Survey," *SSRG International Journal of Computer Science and Engineering*, vol. 2, no. 4, pp. 17-24, 2015. [CrossRef] [Publisher Link]

[3] Nileshkumar R. Patel, and Shishir Kumar, "Wireless Sensor Networks' Challenges and Future Prospects," *2018 International Conference on System Modeling & Advancement in Research Trends (SMART)*, pp. 60-65, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[4] M. Supriya, and T. Adilakshmi, "Secure Routing Using ISMO for Wireless Sensor Networks," *SSRG International Journal of Computer Science and Engineering*, vol. 8, no. 12, pp. 14-20, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[5] T. Senthil Kumaran, and V. Sankaranarayanan, "Early Congestion Detection and Adaptive Routing in MANET," *Egyptian Informatics Journal*, vol. 12, no. 3, pp. 165-175, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[6] Silvano Martello, and Paolo Toth, "A New Algorithm for the 0-1 Knapsack Problem," *Management Science*, vol. 34, no. 5, pp. 633-644, 1988. [CrossRef] [Google Scholar] [Publisher Link]

[7] J. Stuart Hunter, "The Exponentially Weighted Moving Average," *Journal of Quality Technology*, vol. 18, no. 4, pp. 203-210, 1986. [CrossRef] [Google Scholar] [Publisher Link]

[8] Chien-Hsing Huang, Pao-Chu Chen, and Chi-Chang Chen, "A New Method of Target Tracking with Wireless Sensor Networks," *International Journal of Computer Trends and Technology*, vol. 68, no. 4, pp. 175-182, 2020. [CrossRef] [Publisher Link]

[9] Vladimir V. Kalashnikov, *Mathematical Methods in Queuing Theory*, Springer Science & Business Media, 1993. [Google Scholar] [Publisher Link]

[10] Teerawat Issariyakul, and Ekram Hossain, *Introduction to Network Simulator 2 (NS2)*, Introduction to Network Simulator NS2, Springer, Boston, MA, pp. 1-18, 2009. [CrossRef] [Google Scholar] [Publisher Link]

[11] Edo John et al., "A Framework for Scalability on Wireless Connections in Cellular Networks," *International Journal of Computer Trends and Technology*, vol. 68, no. 5, pp. 30-36, 2020. [CrossRef] [Publisher Link]

[12] Kamal Mistry, Sanjay Srivastava, and R.B. Lenin, "Buffer Aware Routing in Interplanetary Ad Hoc Network," *2009 First International Communication Systems and Networks and Workshops*, pp. 1-10, 2009. [CrossRef] [Google Scholar] [Publisher Link]

[13] Sairam Kaleru, and Srinivasa Rao Dhanikonda, "Empyrical Study of Load Balancing Algorithms in Cloud Computing," *Pramana Research Journal*, vol. 8, no. 10, pp. 202-207, 2018. [Google Scholar] [Publisher Link]

[14] Annabathula Phani Sheetal et al., "Load Balancing and Parallel Computation Model for Performance and Accuracy over the Cluster of Nodes," *International Information and Engineering Technology Association*, vol. 27, no. 2, pp. 343-348, 2022. [CrossRef] [Google Scholar] [Publisher Link]