applied sciences

MDPI

Article

TMPAD: Time-Slot-Based Medium Access Control Protocol to Meet Adaptive Data Requirements for Trusted Nodes in Fog-Enabled Smart Cities

Ahmad Naseem Alvi [1], Mumtaz Ali [1], Mohamed Saad Saleh [2,*], Mohammed Alkhathami [2], Deafallah Alsadie [3], Bushra Alghamdi [2] and Badriya Alenzi [2]

[1] Department of Electrical and Computer Engineering, COMSATS University Islamabad, Islamabad 45550, Pakistan; naseem_alvi@comsats.edu.pk (A.N.A.); mumtazaliciit@gmail.com (M.A.)
[2] Information Systems Department, College of Computer and Information Sciences, Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh 11432, Saudi Arabia; maalkhathami@imamu.edu.sa (M.A.); 444008873@sm.imamu.edu.sa (B.A.); bmalenzi@imamu.edu.sa (B.A.)
[3] Information Systems Department, Umm Al-Qura University, Makkah 21955, Saudi Arabia; dbsadie@uqu.edu.sa
* Correspondence: msmsaleh@imamu.edu.sa

Abstract: The popularity of fog-enabled smart cities is increasing due to the advantages provided by modern communication and information technologies, which contribute to an improved quality of life. Wireless networks make them more vulnerable when the network is under malicious attacks that cause a collision in the medium. Furthermore, diverse applications of smart cities demand a contention-free medium access control (MAC) protocol to meet adaptive data requirements. In this work, a time-slot-based medium access control protocol to meet adaptive data requirements (TMPAD) for IoT nodes in fog-enabled smart cities is proposed. TMPAD proposes a trust mechanism to differentiate malicious and legitimate data requests. In addition, it accommodates more legitimate data-requesting nodes to transfer their data during a session by applying the technique for order performance by similarity to ideal solution (TOPSIS) and 0/1 knapsack algorithm. The performance of TMPAD is compared with well-known techniques such as first come first serve (FCFS), shortest job first (SJF), and longest job first (LJF) in different prospective scenarios. The results show that TMPAD scrutinizes more data-requesting nodes in slot allocation, allowing more data transmission in a session, with better mean trust value, as compared to other algorithms.

Keywords: MAC protocol; smart cities; fog; IoT network

Citation: Alvi, A.N.; Ali, M.; Saleh, M.S.; Alkhathami, M.; Alsadie, D.; Alghamdi, B.; Alenzi, B. TMPAD: Time-Slot-Based Medium Access Control Protocol to Meet Adaptive Data Requirements for Trusted Nodes in Fog-Enabled Smart Cities. *Appl. Sci.* **2024**, *14*, 1319. https://doi.org/10.3390/app14031319

Academic Editor: Yutaka Ishibashi

Received: 8 January 2024
Revised: 30 January 2024
Accepted: 31 January 2024
Published: 5 February 2024

1. Introduction

Secure and ease in human lifestyle have led to a healthy increase in the adoption of smart cities in the last decade. In a smart city, its residents enjoy smart services, such as IoT-based healthcare services, intelligent farming, live surveillance, smart industry, and intelligent transportation systems [1,2]. Improved communication and advancement in information technologies help in provisioning quality implementation of these smart city applications by ensuring secure and reliable data delivery [3–6].

Smart city applications save their data on cloud servers for analysis and take proactive measures to avoid any disturbance. However, due to remotely placed cloud servers with increased propagation delay and with the emergence of fog-computing nodes, which are created by placing computing nodes in the near vicinity of smart city applications, fog-computing nodes are now becoming a permanent part of IoT-based smart cities, as shown in Figure 1.

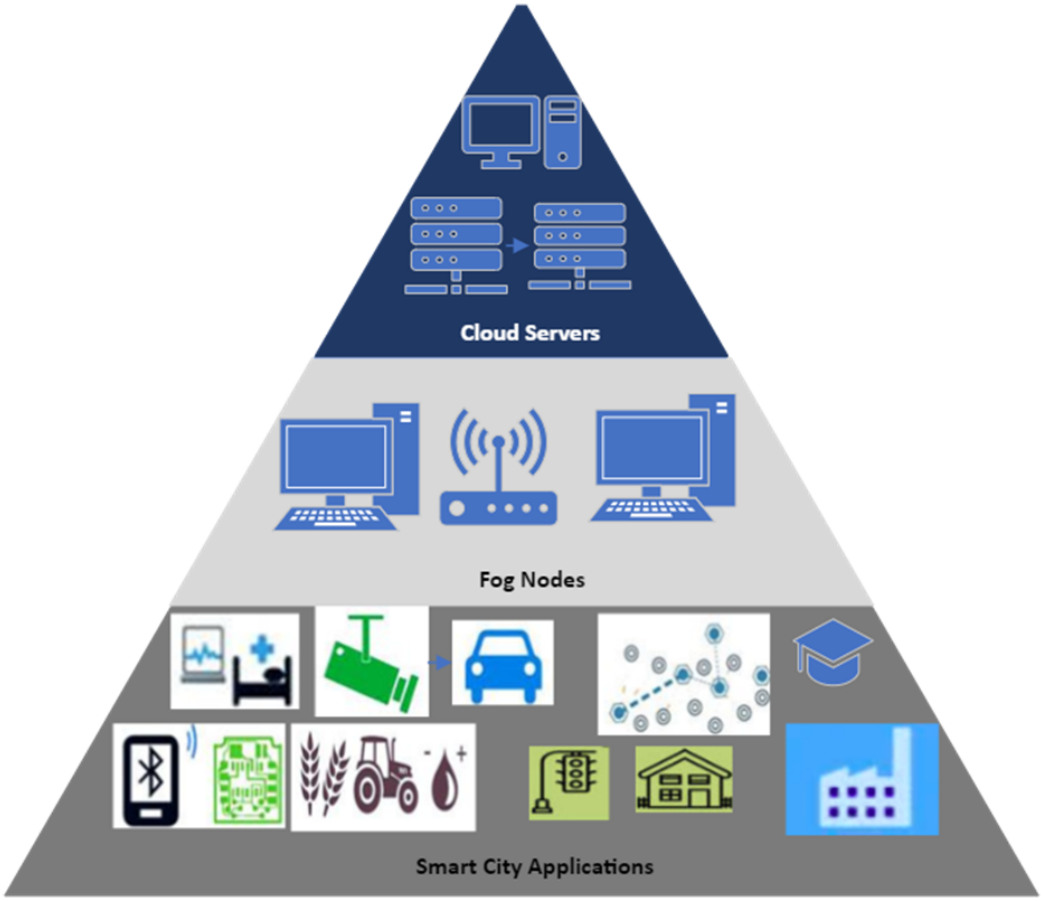

Figure 1. Smart city applications with fog nodes.

Smart cities rely on IoT applications, which use numerous sensors to transmit data to other nodes by sharing the medium. Different medium access control (MAC) protocols have been designed for wireless sensor networks. MAC protocols are differentiated into contention-based and contention-free access. In contention-based MAC protocols, data-sending nodes have to contend with other nodes to transmit their data within the medium [7,8]. However, in contention-free MAC protocols, data-sending nodes are assigned dedicated slots to transmit their data without contending with other nodes [9–12]. The increased number of nodes increases the probability of collisions in contention-based MAC protocols; hence, contention-free MAC protocols are preferred in such environments.

Wireless mediums in IoT networks are vulnerable to malicious nodes causing network collision, denial of service, and unfair allocation [13]. The presence of malicious nodes raises a question as to the trust level of the nodes' requests, and a trust level of the networking nodes is required, resulting in compromised QoS in the networks [14–17]. Attacks during the contention period are hard to identify because there are already chances of collision, medium busyness, and unfairness due to system occupation by other nodes. Thus, contention-free MAC protocols are preferred over contention-based MAC protocols because there are dedicated slots for each node and because interference of any other node can easily be determined. However, in contention-free MAC protocols, malicious nodes acting as member nodes are allocated dedicated slots in each communication session and disturb the medium, leading to reduced medium utilization.

In smart cities, different applications require sensor nodes to collect and transmit data in various time intervals. These data requirements may change based on the situation and need to be adaptable. A classic TDMA-based MAC protocol with fixed-duration time slots for all nodes is not recommended because each node is assigned a fixed-duration time slot in each communication session. If a node wants to transmit more data than the

assigned allocation, it may not be possible during that communication session. To meet these adaptive data requirements, an adaptive TDMA-based MAC protocol with enhanced medium utilization with secure and reliable data communication is required. This can be achieved by implementing a trust management algorithm that verifies the authenticity of nodes [18,19].

This research work proposes a time-slot-based medium access control protocol to meet adaptive data requirements (*TMPAD*) of trusted nodes for fog-enabled smart city applications. *TMPAD* distinguishes between malicious and trusted nodes, and guaranteed time slots (GTSs) are only assigned to the trusted nodes. The salient features of *TMPAD* are mentioned below.

1. A trust mechanism for fog nodes to differentiate the data forwarded from the legitimate and malicious nodes.
2. An efficient TDMA-based MAC protocol for IoT nodes to transfer their data in a contention-free manner to meet adaptive data requirements.
3. TMPAD allows varying lengths of communication sessions. If the number of data-requesting slots is less, the session is reduced. However, if the slot requests increase, the session size increases till a certain threshold.
4. TDMA slots are assigned to the nodes by applying TOPSIS and 0/1 knapsack algorithms.

The organization of the manuscript is as follows:

The previous studies and trust-based data management in different areas of communications applications are discussed in Section 2. Our proposed TDMA-based MAC protocol with different frame structures, a trust management system, and the slots allocation mechanism are described in detail in Section 3. Comparative performance analysis of the proposed scheme with other well-known algorithms, along with extensive simulation results, is discussed in Section 4, and Section 5 provides the conclusion of our manuscript.

2. Related Work

IoT-based smart city applications rely on secure wireless communications and require efficient techniques to detect malicious attacks and evaluate the trust of the IoT nodes. Trustful communication between different wireless nodes is an important area of research these days.

In [20], the authors examined the trade-off between benefits and risks associated with trustworthy communication. They introduced an adaptive trust-based situation-aware access control framework, known as MATS, which incorporated semantic web technologies and game theory. The proposed framework used the dynamic and adaptive nature of the network, and its performance was evaluated through experimental results.

In [21], the authors proposed a trust framework for IoTs and UAVs in the scenario of vehicular networks. The proposed framework implemented security and privacy measures to enhance the trust level of high-priority vehicle safety data. Additionally, the work proposed a trajectory optimization algorithm to establish a trustworthy agreement between nodes for data collection. The proposed framework improved the trust level of vehicle safety data by 33.34%.

In [22], the authors proposed a framework named MapReduce to manage trust in big data. The proposed framework reduced data complexity and applied a trust mechanism using the bipartite matching algorithm. Furthermore, the work allocated trust-based time slots to the network users. Results showed a 94.7% improvement in searching the big data as compared to the exhaustive search technique.

The authors in reference [23] addressed the necessity for reliable data in online networks, highlighting potential threats such as the creation of fraudulent accounts. They introduced a trust-aware framework designed to identify malicious users and utilized a data-balancing technique to pinpoint nodes causing disruptions in terms of frequent message sharing. Through extensive experiments, it was shown that the proposed scheme enhances data precision by 59.16% compared to baseline algorithms.

In reference [24], the authors presented a trust-based attestation model to mitigate challenges arising from virtual networks in online real estate scenarios. The model uses rule-based decision making to recognize legitimate users of property data, thus establishing trust to foster reliable applications for authorized data users. The work also implemented the prototype of the proposed trust management system on an open-source MANO platform.

Exploiting evidence theory, the authors in reference [25] proposed a decentralized service-oriented trust management model, concentrating on an IoT-based healthcare system. The work introduced a reward and punishment system where legitimate healthcare service providers are rewarded and punitive measures are applied to malicious users. The proposed system enabled trust in healthcare services and recommendations by providing security against on–off attacks, bad-mouthing, and good-mouthing simultaneously.

3. Proposed Scheme

Smart-city-based IoT applications use a Wireless Personal Area Network (WPAN) comprising IoT devices, sensors, and computing servers. IoT devices face security attacks by different malicious nodes that corrupt the data and increase the chances of collisions while accessing the medium for data transmission. It is hard to identify attacks by malicious nodes when they become a member of the network. In addition, malicious attacks in accessing medium during the contention access period are hard to identify and may cause a collision. That is why contention-free medium access control protocol is preferred over contention-based MAC protocol in wireless sensor-based IoT networks. A proposed time-slot-based medium access control protocol to meet adaptive data requirements (TMPAD) is designed for IoT-based wireless networks in fog-enabled smart cities. The protocol aims to provide a collision-free environment for nodes to transfer varying amounts of data while identifying and preventing malicious attacks. TMPAD's main features include the following:

- Offers scalability by allowing new nodes to become members of the WPAN in each session.
- Modifies data slots to align with the adaptive data requirements determined by GTS-requesting IoT nodes.
- Computes the trusted data by calculating the trust level of each node.
- Efficiently allocates GTS to accommodate more GTS-requesting nodes with better data transmission and higher mean trust value by applying the TOPSIS algorithm.

A list of notations used in Sections 3 and 4 of this manuscript, along with their descriptions, are mentioned in Table 1.

Table 1. List of Notations used in this Paper.

Notation	Description
CD	CAP duration
SD	Time slot
SD_b	Number of bits in a time slot
GTS_{Req}	Number of GTS required
DTP	Data transmission period
T_Y^X	Trust value of node X for node Y
A^{tf}	Trust value calculated for node A
T_N^{tf}	Trust value calculated by all neighboring nodes
σS	Sigmoid function to find trust probability of a node
C_i	TOPSIS value calculation criteria
V_i	TOPSIS value of node i
TS_a	Time slots received in session a

Table 1. *Cont.*

Notation	Description
σ_{TD}	Data transmitted by all nodes
σ_{TDL}	Transmitted data by legitimate nodes only
ζ_{ND}	Total network delay
ζ_{NDL}	Total network delay by legitimate nodes only
ζ_{NDA}	Average network delay
ζ_{NDLA}	Average transmission delay by legitimate nodes only

3.1. Communication Round of the Proposed Scheme

The communication round in the proposed MAC protocol comprises multiple sessions. Each session starts with a beacon frame, followed by a contention access period, an announcement period, and a data transmission period, as shown in Figure 2.

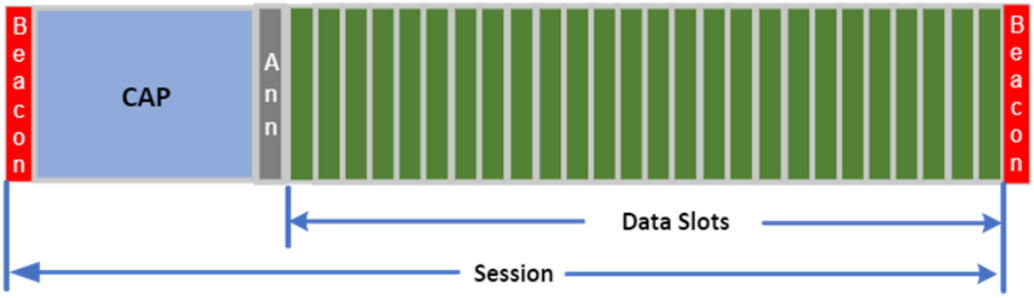

Figure 2. Communication session of the proposed MAC.

3.1.1. Beacon Frame

The beacon frame of each session in the proposed scheme informs other nodes about its presence and informs nodes about the CAP duration (CD) with the help of TDMA slot duration (SD) as follows:

$$CD = 16 \times SD \tag{1}$$

However, SD is calculated by following the modulation scheme and symbol rates used in IEEE 802.15.4 standard [26] by allowing 4 bits/symbol with a symbol rate of 62,500 symbols/s and a data rate of 250 kbps as follows:

$$SD(msec) = \frac{250 \times 1000}{62,500} \tag{2}$$

and the number of bits in an SD (SD_b) is calculated as

$$SD_b = \frac{62,500 \times 4}{250} \tag{3}$$

3.1.2. Contention Access Period

During the contention access period, all nodes contend with each other in accessing the medium by applying the CSMA/CA algorithm. In this period, two types of requesting frames are received by the PAN coordinator, such as

1. Joining request from nonmember nodes.
2. GTS requests from member nodes.

Nonmember nodes after listening to the beacon frame message send a join request (JR) message to the PAN coordinator to become a member of the WPAN. The JR frame includes the information of the requesting node ID. In response, the WPAN coordinator allocates

the requested node a membership during the next beacon frame message and allocates a unique 8-bit short address. All the member nodes in a WPAN are allocated a unique 8-bit short address for their identification. This limits the size of WPAN to 256 nodes.

All member nodes that are assigned a short 8-bit address are allowed to send their data during the contention-free period by allocating GTS. Nodes without allocating a short address are not allowed to send their GTS request to the WPAN coordinator. Nodes having data transmit a GTS request to the WPAN coordinator with the number of slots required to send their data. Nodes calculate the number of GTS required (GTS_{Req}) in transmitting their data with the help of data slot capacity and the amount of data (D) required to be transmitted as follows:

$$\left\lceil GTS_{Req} = \frac{D}{SD_b} \right\rceil \tag{4}$$

3.1.3. GTS Announcement Period

The WPAN coordinator, after successfully receiving all the GTS requests, informs nodes about their slot allocation in the GTS announcement period (GAP). In addition, the PAN coordinator also informs about the total number of GTS assigned to the nodes, which helps nodes to know about the start of the next beacon frame. Each node is informed about its allocated slot by providing information about the starting slot and the number of requested slots assigned to the node. The duration of the GAP varies in each session and depends upon the number of GTS assigned to GTS-requesting nodes in each session. A complete GTS announcement frame is shown in Figure 3.

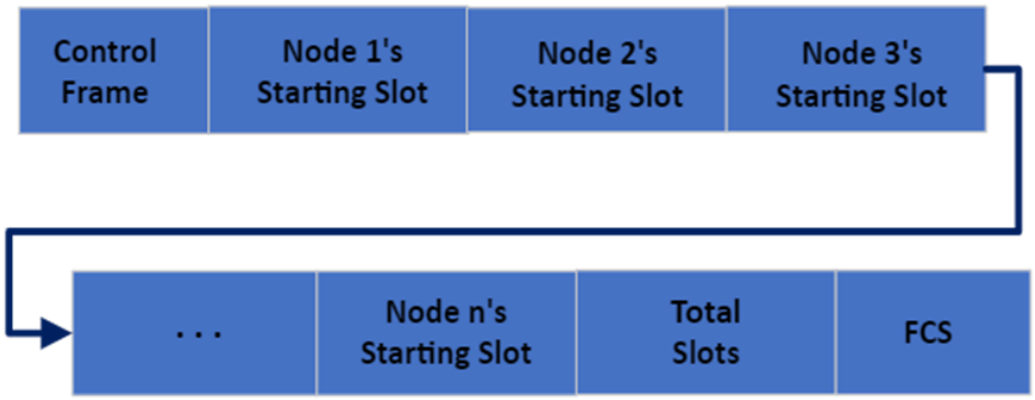

Figure 3. Frame structure of announcement period.

3.1.4. Data Transmission Period

At the end of the announcement period, nodes know their allocated GTS. The data transmission period (DTP) comprises a number of TDMA slots and allows nodes to transfer their data without contending with other nodes. All those nodes that are allocated GTS are allowed to send their data to the WPAN coordinator in their assigned data slots. DTP varies according to the number of slots allocated to the GTS-requesting nodes. However, the maximum duration of a DTP (DTP_{max}) in a session is calculated as follows:

$$DTP_{max} = [SD \times (250 - 16)] - (BF + GAP) \tag{5}$$

If the number of GTS requests is less than the available slot limit, all the nodes are assigned GTS according to their requests on a first come first serve (FCFS) basis. However, if the number of GTS requests is more than the available slots limit, GTS will be allocated to nodes by applying the TOPSIS algorithm, as described in Section 3.3.1. A communication session concludes once all nodes transmit their data during the DTP.

3.2. Trusted Data Identifications

Infiltration of malicious nodes in IoT networks degrades the performance of the network. Malicious nodes being network members seriously degrade the network's performance as they are not easily identified. This section outlines a method to identify the trustworthy data. *TMPAD* assists fog nodes in differentiating the malicious and legitimate nodes by evaluating the probability of trusted and malicious nodes within a WPAN.

The trust level of a node is established through the assessment of its interactions with neighboring nodes during packet exchanges. More precisely, it is determined by the accurate reception of requested data packets in response to the generated requests. For example, if a node X transmits A requests to its neighboring node Y and receives B response files from node Y, with A_{corr} denoting the corrupted files received, the trust probability of node X for node Y (T_Y^X) is calculated as follows:

$$T_Y^X = \frac{(B - A_{corr})^2 \times SNR_{max}}{SNR_{XY} \times A \times B} \tag{6}$$

where the channel capacity (CC) between two links is calculated by using Shannon capacity [27] as follows:

$$CC = B \times log_2(1 + SNR) \tag{7}$$

where SNR is the signal-to-noise ratio, and B represents the bandwidth. The higher the trust value, the higher the trust level of the neighboring node.

Each node computes the trust value for every directly connected neighboring node. In the case where there are m nodes situated close to a trust-finding node A^{tf}, and the trust value for each neighboring node is established using Equation (6), the collective trust value of A^{tf} is then calculated as follows:

$$A^{tf} = \frac{\sum_{i=1}^{m} T_i^{tf}}{m} \tag{8}$$

The potential threat arises from a malicious node capable of transmitting a deceitful trust value, whereas a genuine node would accurately relay the true trust value of its neighboring nodes. Consequently, depending solely on this parameter is insufficient for determining the trustworthiness of a node. To authenticate the trust value provided by a node, it becomes essential to collect input from other nodes within the clustering network. All nodes directly linked to the trust-finding nodes are required to transmit their trust values concerning the trust-finding node to the PAN coordinator, as shown in Figure 4. This procedural step aids in the assessment of the trust value associated with each neighboring node.

The trust value of a node by all its available N neighbors T_N^{tf} is calculated as

$$T_N^{tf} = \frac{\sum_{i=1}^{N} T_i^{tf}}{N} \tag{9}$$

To calculate the trust probability of all nodes in a cluster, a weighted metric is assigned to the trust values calculated in Equations (8) and (9) as below.

1. A high weight is assigned to the neighboring trust value, as calculated in Equation (9).
2. A low weight is assigned to the self-calculated trust value, as determined in Equation (8).

A sigmoid function is defined to determine the trust probability for each network node in a WPAN and is calculated as follows:

$$\sigma S_i = \frac{1}{1 + e^{-[a(A_i^{tf}) + T_n^{tf}(T_i^{tf})]}} \tag{10}$$

A higher probability of trust value means that nodes are legitimate and their GTS requests needs to be preferred over the nodes with reduced probability. However, if the

trust probability is less than a critical value then its GTS request will not be entertained. Algorithm 1 represents the complete procedure.

Algorithm 1: Trust-evaluating algorithm.

Input:
Number of nodes K
Total number of channels M
Threshold Value V_{th}
SNR between all possible links $SNR_1, SNR_2, SNR_3, \ldots SNR_M$,
Trust value calculated by node itself $T_1^{tf}, T_2^{tf}, T_3^{tf}, \ldots, T_K^{tf}$
Trust value computed by neighboring nodes T_{nx}^{tf}
for $i = 1$ *to* K **do**
    Calculate σS_i
    if $\sigma S_i \leq V_{th}$ **then**
        Malicious node
    end
    else
        Legitimate node
    end
    $i++$
end

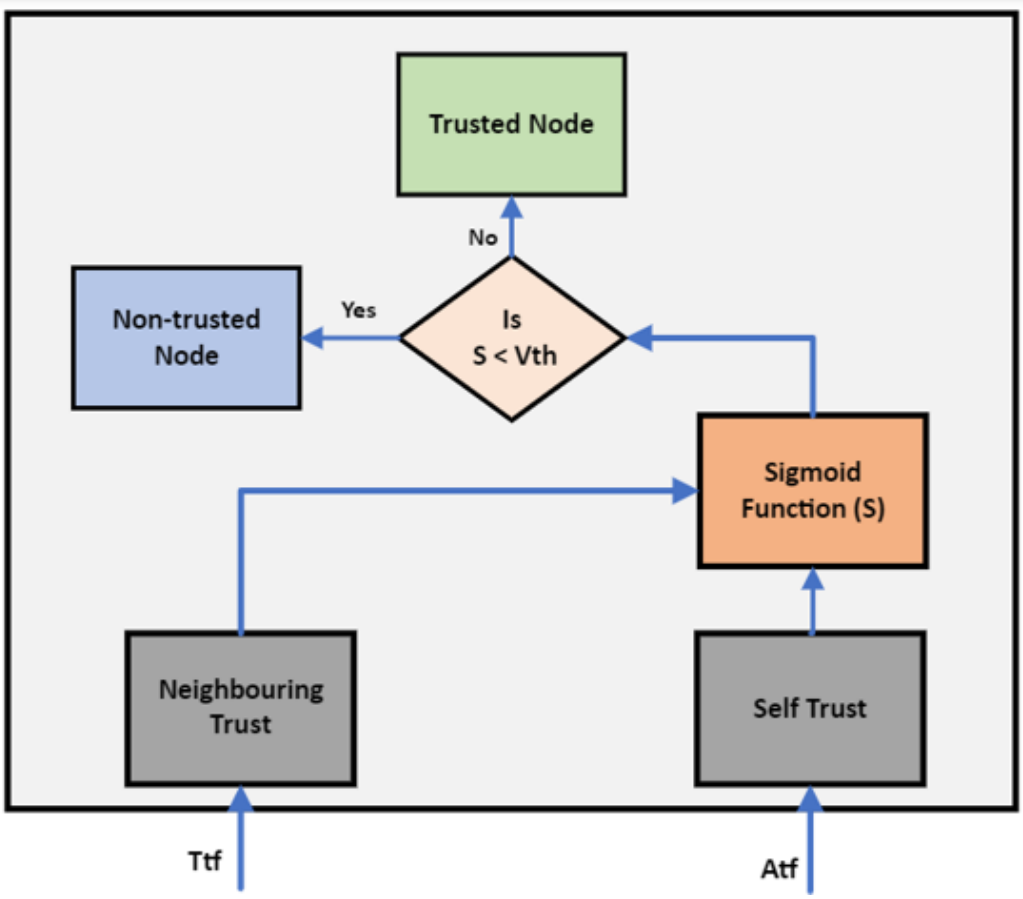

Figure 4. Trust management procedure to determine legitimate nodes in the proposed scheme.

3.3. GTS Allocation Procedure

All the member nodes who have data calculate the number of GTSs required to send their data with the help of Equation (4). The WPAN coordinator, after receiving all these requests, discards the GTS requests that are originated by the malicious nodes and calculates the total number of GTS requests originated by the nonmalicious nodes. If the total number of GTSs requested by the nodes is within the available GTS limit of the session, as mentioned in Equation (5), then all the nodes are assigned GTSs according to their demand by applying the TOPSIS algorithm, as mentioned in Section 3.3.1. Suppose the number of GTSs is more than the available limit of the session. In that case, the PAN coordinator scrutinizes the nodes by applying the 0/1 knapsack algorithm, as mentioned in Section 3.3.2. If the number of GTSs exceeds the session limit, the PAN coordinator evaluates nodes using the 0/1 knapsack algorithm outlined in Section 3.3.2.

3.3.1. TOPSIS

If the number of GTS requests received by the PAN coordinator is less than its available slot capacity, then all the GTS requests will be entertained by informing all the successful nodes about their time slot information. Due to adaptive data slot requirements with varying channel conditions and the waiting time to transmit data, a technique for order performance by similarity to ideal solution (TOPSIS) is applied to determine the priority of the GTS-requesting nodes. The node with a higher TOPSIS value will be assigned GTSs before the other nodes.

In this work, the TOPSIS value is calculated by considering the following parameters.

- **Trust value:** The trust value of the requesting nodes is accorded the highest weight in the TOPSIS calculation, as detailed in Section 3.2. This ensures that nodes with a higher level of trustworthiness are given precedence in GTS allocation.
- **Emergency data:** Priority is assigned to nodes presenting emergency data, such as critical patient health parameters. This proactive approach ensures that nodes with urgent data transmission requirements are prioritized for GTS allocation.
- **Channel capacity:** Considering the communication channel's pivotal role in data transfer from IoT nodes to the PAN coordinator, the TOPSIS algorithm factors in the channel capacity. A superior communication channel is favored, as it enables the PAN coordinator to achieve better data rates during the data transfer process.
- **Data request failure:** Nodes that were not allocated GTS in the last session receive preferential treatment in the TOPSIS calculation. This ensures a fair and adaptive allocation strategy, taking into account the historical performance and needs of the nodes.

By integrating the TOPSIS algorithm into our GTS allocation framework, we aim to optimize the prioritization process, considering dynamic data slot requirements and varying channel conditions. This adaptive approach enhances the efficiency and responsiveness of our PAN coordinator in managing GTS requests, ultimately contributing to the overall robustness of our wireless communication system.

The TOPSIS algorithm is applied to guaranteed time slot (GTS) allocation, where the goal is to rank alternatives (nodes) based on their performance across multiple criteria, as shown in Algorithm 2. In the first step, each criterion value for each alternative is normalized to bring them to a comparable scale. This is achieved by dividing each criterion value by the square root of the sum of the squared values for that criterion across all alternatives. Next, the normalized values are then multiplied by their respective weights and summed to obtain a weighted normalized value for each alternative. This step incorporates the importance of each criterion in the decision-making process. Further, for each criterion, the maximum (ideal solution) and minimum (negative-ideal solution) values across all alternatives are determined. These values represent the best and worst possible performances for each criterion. Then, Euclidean distances are calculated for each alternative based on the differences between the weighted normalized values and the ideal and negative-ideal solutions. This step quantifies how close or far each alternative is from the ideal and negative-ideal solutions for each criterion.

Algorithm 2: TOPSIS algorithm for GTS allocation.

Data: Decision matrix X representing GTS allocation performance on each criterion, Weights w_j for each criterion (w_t, w_e, w_c, w_f)

Result: Ranked list of alternatives based on relative closeness values for GTS allocation

Normalization:
for *each alternative i and criterion j* **do**
  $\text{Normalized Value}_{ij} = \frac{\text{Value}_{ij}}{\sqrt{\sum_{i=1}^{n}(\text{Value}_{ij})^2}}$
end
Weighted Normalized Decision Matrix:
for *each alternative i* **do**
  $\text{Weighted Normalized Value}_i = w_t \cdot \text{Normalized Value}_{it} + w_e \cdot \text{Normalized Value}_{ie} + w_c \cdot \text{Normalized Value}_{ic} + w_f \cdot \text{Normalized Value}_{if}$
end
Ideal and Negative-Ideal Solutions:
for *each criterion j* **do**
  $A_j^+ = \max(\text{Weighted Normalized Value}_i)$ `// for each` i
  $A_j^- = \min(\text{Weighted Normalized Value}_i)$ `// for each` i
end
Calculate Euclidean Distances:
for *each alternative i* **do**
  Calculate d_i^+ and d_i^- for Trust Value, Emergency Data, Channel Capacity and Data Request Failure
  $d_i^+ = \sqrt{\sum_{j=1}^{m}(\text{Weighted Normalized Value}_{ij} - A_j^+)^2}$
  $d_i^- = \sqrt{\sum_{j=1}^{m}(\text{Weighted Normalized Value}_{ij} - A_j^-)^2}$
end
Calculate Relative Closeness:
for *each node i* **do**
  Calculate $C_i(\text{TV}) = \frac{d_i^-}{d_i^+ + d_i^-}$
  Calculate $C_i(\text{ED}) = \frac{d_i^-}{d_i^+ + d_i^-}$
  Calculate $C_i(\text{CC}) = \frac{d_i^-}{d_i^+ + d_i^-}$
  Calculate $C_i(\text{DRF}) = \frac{d_i^-}{d_i^+ + d_i^-}$
end
Overall TOPSIS Value:
for *each node i* **do**
  Calculate overall TOPSIS value
  $V_i = \frac{1}{4}(C_i(\text{TV}) + C_i(\text{ED}) + C_i(\text{CC}) + C_i(\text{DRF}))$
end
Ranking:
Rank nodes based on their overall TOPSIS values in descending order.

After that, the relative closeness values are calculated for each alternative by following the criteria described in TOPSIS algorithm [28], as shown in Equation (11).

$$C_i(Criterion) = \frac{d_i^-}{d_i^+ + d_i^-} \tag{11}$$

Here, d_i^+ and d_i^- are the distances calculated from the positive and negative ideals, respectively.

After that, the relative closeness values for all criteria are combined into an overall TOPSIS value for each alternative by following Equation (12).

$$V_i = \frac{1}{4}(C_i(TV) + C_i(ED) + C_i(CC) + C_i(DRF)) \quad (12)$$

This step ensures a balanced consideration of all criteria. Finally, alternatives are ranked based on their overall TOPSIS values in descending order. The higher the TOPSIS value, the better the alternative is considered in terms of GTS allocation.

The algorithm provides a systematic and quantitative way to evaluate and rank alternatives considering multiple criteria. It is particularly useful in decision-making scenarios where a trade-off between different criteria needs to be made.

3.3.2. 0/1 Knapsack Algorithm

When the number of GTS requests is more than the available GTSs in that session, then it is not possible for the PAN coordinator to allocate GTSs to all the legitimate requested nodes, and it needs to scrutinize them.In this work, an optimal GTS allocation mechanism based on the 0/1 knapsack algorithm is proposed. The 0/1 knapsack problem optimally scrutinizes the most valuable items within the available sack capacity.

The nodes have different amounts of data with different priority levels that are required to be sent to the fog node acting as a PAN coordinator. In addition, all the nodes do not have the same channel capacity. If we represent the number of available GTSs in a session as GTS_{Avail}, and the GTS requests received by the PAN coordinator from node i as GTS_i, along with its calculated TOPSIS value $TOPSIS_i$, we can map the GTS scrutiny problem to the 0/1 knapsack problem. This can be seen in Table 2.

Table 2. Knapsack mapping table.

GTS Allocation Problem	Knapsack Problem
Total GTS capacity in a session	Carrying capacity of the knapsack
GTS request of an tiem	Weight of an item
TOPSIS value of an item	Value of an item

Suppose T legitimate nodes require N number of GTS slots to transmit their data to the PAN coordinator. The proposed 0/1 knapsack-based algorithm scrutinizes k number of GTS-requesting nodes to send their data in the allocated TDMA-based slots by meeting the following two constraints:

- The requested slots should be allocated within the available limit and are represented as

$$\sum_{i=1}^{T} \leq GTS_{Avail} \quad (13)$$

- The GTS-requesting nodes must be selected to maximize the TOPSIS values of the selected nodes.

$$Max \sum_{a=1}^{k} TOPSIS_a \quad (14)$$

The proposed knapsack algorithm to scrutinize the requested nodes is shown in Algorithm 3.

Algorithm 3: Optimal GTS allocation criteria.

Input:
$T \leftarrow$ *Total Number of TDMA requesting legitimate nodes*
$N \leftarrow$ *Total number of TDMA slots requested by nodes*
$GTS_{Avail} \leftarrow$ *Available slots in a session*
$i \leftarrow$ *GTS-requesting node ID*
$X[i,j] \leftarrow$ *Fill the cell value*i^{th}*node and* j^{th}*slot*
$S_i \leftarrow$ *Slots requested by* i^{th}*node*
If $T \leq GTS_{Avail}$
Allocates slots to all requesting nodes by prioritizing high TOPSIS value nodes
Else
Apply 0/1 knapsack algorithm
0/1 knapsack algorithm
Initialize the first row of knapsack table with all 0
Initialize the first column of knapsack table with all 0
for *all GTS-requesting nodes* **do**
  for *all TDMA slots* **do**
    If $S_i \leq j$ **If** $S_i + X[i-1, j-j_i] > X[i-1, j]$
    $X[i,j] = j_i + X[i-1, j-j_i]$
    Else
    $X[i,j] = X[i-1, j]$
    EndIf
    Else
    $X[i,j] = X[i-1, j]$
    EndIf
  end
  $T \leftarrow i$
  $N \leftarrow j$
end
optimal slot requesting nodes:
while $i > 1$ *and* $j > 1$ **do**
  If $B[i,j] > B[i-1, j]$
  i^{th} node will be selected
  $i = i - 1$
  $j = j - j_i$
  Else
  $i = i - 1$
  EndIf
end

4. Results and Analysis

To analyze the performance of the proposed MAC protocol, a system model based on multiple IoT-based smart sensors with adaptive data requirements creates a WPAN in a smart city. These IoT nodes may comprise some malicious nodes that disturb the TDMA-based medium by requesting the number of TDMA-based time slots. All the nodes in the WPAN transfer their data to the fog node acting as a PAN coordinator by using TDMA-based time slots in a contention-free manner.

WPAN comprises a total of J nodes comprising K legitimate nodes and M malicious nodes. If one of the WPAN nodes i requests S_i slots in transferring its data in a session to

the PAN coordinator, then the number of slots requests received by a fog node in a session a (TS_a) is calculated as

$$TS_a = \sum_{i=1}^{J} S_i \tag{15}$$

If node i transmits D_i amount of data in its allocated time slots and N nodes are successfully assigned GTSs in transmitting their data in a session, then the data that are transmitted in X number of sessions (σ_{TD}) are calculated as

$$\sigma_{TD} = \sum_{a=1}^{X} \sum_{i=1}^{N} \times D_{ia} \tag{16}$$

If there are B malicious nodes that are allocated TDMA slots during these sessions, then the total data transmitted by only legitimate nodes (σ_{TDL}) during X number of sessions are calculated as

$$\sigma_{TDL} = \sigma_{TD} - \sum_{a=1}^{X} \sum_{b=1}^{B} D_{ba} \tag{17}$$

The data transmission time of a node is calculated as the time when a node has data to send until it successfully transmits all its data to the PAN coordinator. The data transmitting time of slots allocating time depends upon the data size as well as the channel capacity between the data transmitting node and the PAN coordinator. If the transmission time of node i in transmitting its time during a session is t_i, and P nodes successfully transmitted their data during that session, then the network delay ND of all the GTS-allocating nodes in a session is calculated as

$$ND = \sum_{i=1}^{P} t_i \tag{18}$$

For X number of sessions, the total network delay (ζ_{ND}) is calculated as

$$\zeta_{ND} = \sum_{a=1}^{X} \sum_{i=1}^{P} \times t_{ia} \tag{19}$$

If there are B malicious nodes among the successfully allocated nodes, then the time calculated in transmitting the data of only legitimate nodes during X number of sessions ζ_{NDL} is calculated as

$$\zeta_{NDL} = \zeta_{ND} - \sum_{a=1}^{X} \sum_{b=1}^{B} \times t_{ba} \tag{20}$$

Nodes have different times for transmitting their data to the PAN coordinator. If there are T_N nodes that transmitted data during X sessions and there are T_M malicious nodes during these sessions, then the average transmission delay of all slots allocating nodes and of legitimate nodes for X are represented as ζ_{NDA} and ζ_{NDLA}, respectively, and are calculated as shown in Equations (21) and (22).

$$\zeta_{NDA} = \frac{\sum_{a=1}^{X} \sum_{i=1}^{P} \times t_{ia}}{T_N} \tag{21}$$

$$\zeta_{NDLA} = \frac{zeta_{ND} - \sum_{a=1}^{X} \sum_{b=1}^{B} \times t_{ba}}{T_N - T_M} \tag{22}$$

To assess the effectiveness of $TMPAD$ from various perspectives, we set up a simulation environment that covers a wide range of possibilities that can be experienced by the network, ranging from 50 to 150 nodes. The network mostly comprises legitimate nodes and there are two to four malicious nodes in the network as, generally, the malicious nodes will be fewer than the legitimate nodes. These IoT nodes are a member of a WPAN and the fog node is their PAN coordinator. The data rate is 250 kbps. The simulation environment

also considers the varying bandwidth as well as varying power of the transmission of the node's data, as these are the resources that can be adjusted for data transmission. The simulation parameters used in this simulation environment are shown in Table 3.

The results of the proposed scheme in this simulation environment are compared with well-known standards such as first come first serve (FCFS) [29], longest job first (LJF) [30], and shortest job first (SJF) [31] algorithms. The comparative analysis includes transmitted data, number of GTS-requesting nodes entertained, transmission delay, and the mean trust value of GTS-allocating nodes for varying numbers of data-requesting nodes, for varying channel bandwidth, and for varying signal power. Furthermore, the performance of the proposed scheme is compared in different prospects with 0/1 knapsack, FCFS, SJF, and LJF for varying trust threshold values.

Table 3. Simulation parameters.

Parameter	Value
Coverage area	50 × 50 m
Maximum duration of a session (s)	1
Number of nodes	50–150
Maximum data rate (kbps)	250
Maximum number of data slots	232
Maximum slot capacity (kB)	4
Slot duration (s)	0.004
Contention access period (s)	0.064
Channel bandwidth	2000–10,000
Data transmission power	10–30
Trust threshold probability	0.3
Number of legitimate nodes	48–146
Number of malicious nodes	2–4

4.1. Data Transmission

Data transmitted is calculated for a node that is allocated its requested GTS to transmit its data. In this section, data transmission by the legitimate node is calculated for a complete communication session. The results are evaluated for varying numbers of GTS-requesting nodes, for varying transmitting power, and for varying bandwidth capacity.

Results shown in Figure 5 are observed for varying numbers of GTS-requesting nodes received by the fog node. It is evident from the results that when the number of GTS-requesting nodes is 50 then GTSs allocated to the nodes in TMPAD select nodes in such a way that maximum data are transmitted within a session. This difference with other competitors increases with the increase in data-requesting nodes with the same number of data slots. It is because the proposed scheme by applying TOPSIS and the knapsack algorithm allows more data to be transmitted within a session; however, the other three standards do not scrutinize data-requesting nodes intelligently and almost the same amount of data is transmitted with the increase in data-requesting nodes.

Results in Figures 6 and 7 show the amount of data transmitted by nodes for varying trust probability of the networks calculated and for varying transmission power of 100 data-requesting nodes. It is evident from the results that the data transmitted in a session by the proposed TMPAD is significantly more than the other three standards. Figure 6 shows that when GTS-requesting nodes have varying channel bandwidth, the higher the channel bandwidth, the more the data can be transmitted by a node. These results show that with the increase in channel bandwidth, the data transmitted by all session nodes increases. However, TMPAD allows significantly more data as compared to the other three algorithms.

The results in Figure 7 are observed when the transmitting power of the GTS-requesting node is increased. The results show that increased transmission power increases the data transmission capability of a node. It is evident from the results that TMPAD allows more data-transmitting nodes to transmit their data within a session as compared to the other three standards for all different values of transmission power.

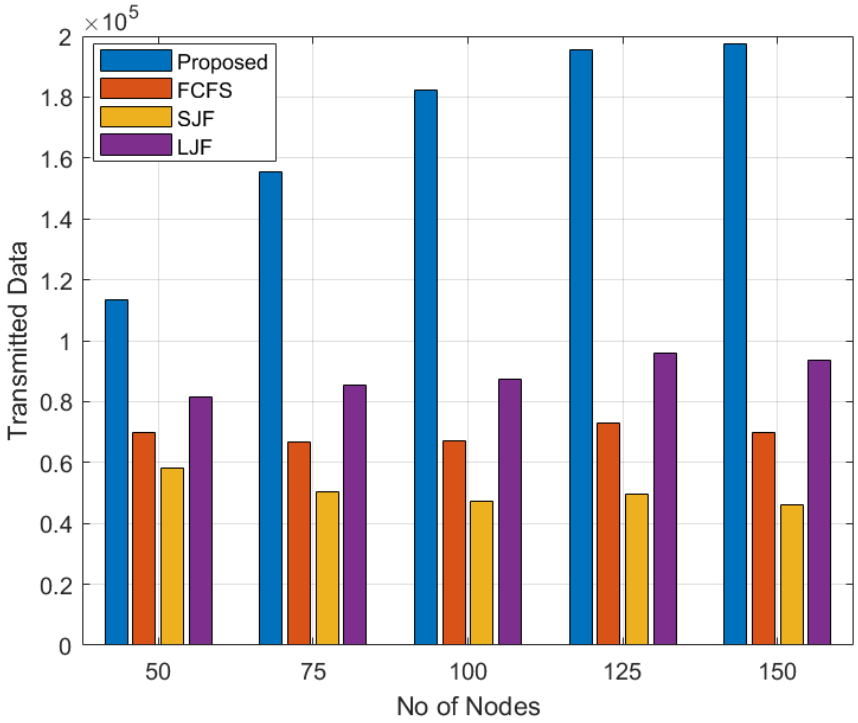

Figure 5. Data transmitted by legitimate nodes in a session for varying number of nodes.

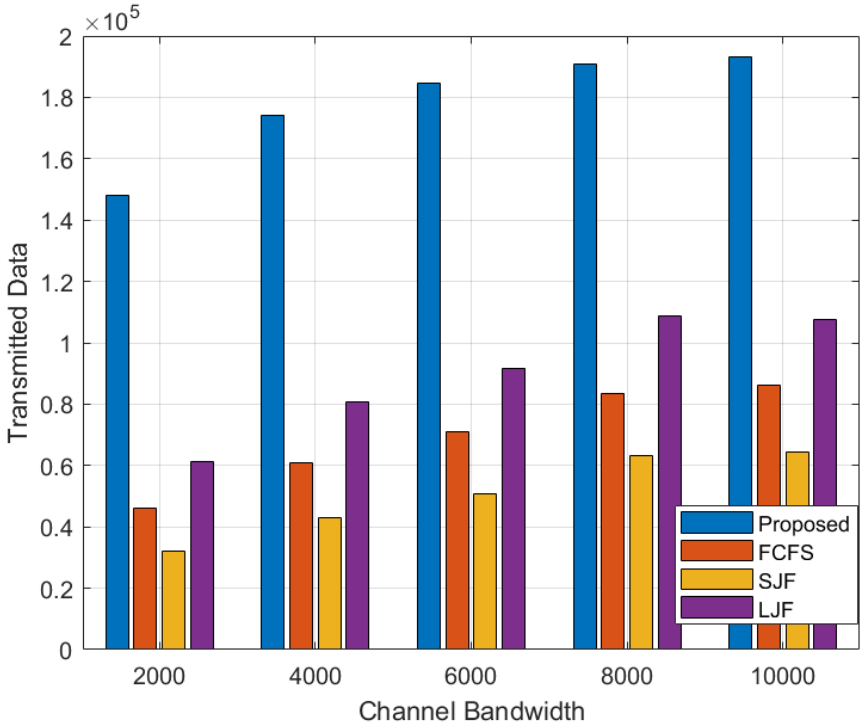

Figure 6. Data transmitted by legitimate nodes in session for varying channel bandwidth.

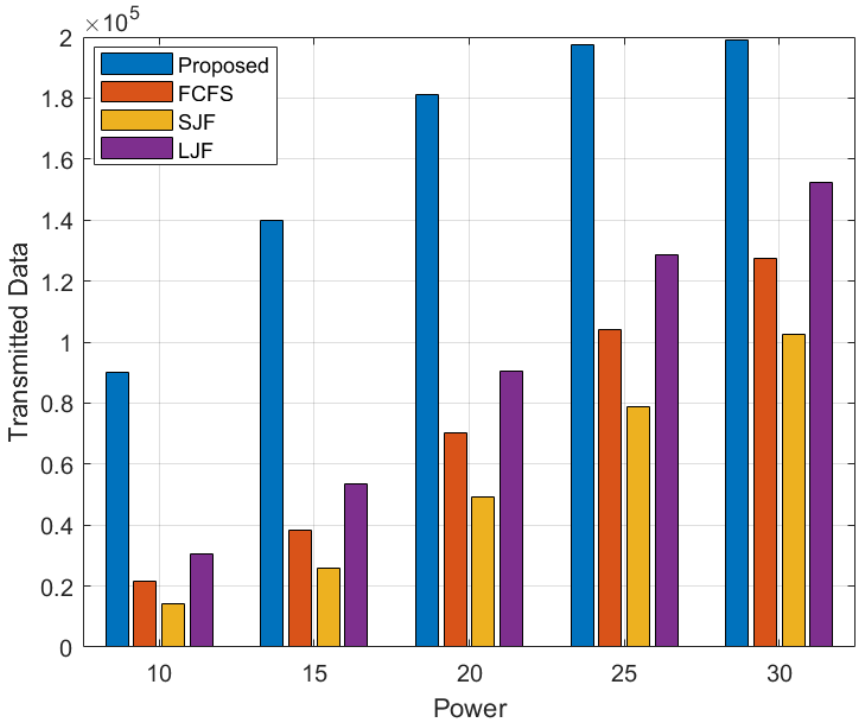

Figure 7. Data transmitted by legitimate nodes in session for varying transmission power.

4.2. GTS-Allocated Nodes

Fog nodes, after receiving several GTS requests during CAP of a session, scrutinize the nodes that are allowed to send their data in their allocated GTS during that session. The proposed *TMPAD* assists fog computing nodes in scrutinizing nodes by applying the 0/1 knapsack algorithm on TOPSIS values. However, FCFS assigns GTSs to nodes by applying a first come first serve algorithm, SJF allocates GTSs to those nodes first which have initiated the least number of time slot requests, and in LJF, the nodes with the highest slot requests will be allocated GTSs first. The results are compared for varying numbers of nodes, varying channel bandwidth, and varying transmission power of the GTS-requesting nodes and are shown in Figures 8, 9 and 10, respectively.

Results shown in Figure 8 verify that the *TMPAD* entertains more GTS-requesting nodes in a session as compared to the other three algorithms. When the number of GTS-requesting nodes is 50, then *TMPAD* scrutinizes 32 legitimate nodes to transmit their data as compared to 20 legitimate nodes in other algorithms within the same time slot capacity in a session, because *TMPAD* scrutinizes the nodes with higher trust values. However, the other three algorithms do not differentiate between legitimate and malicious nodes. With the increase in the number of GTS-requesting nodes, more legitimate nodes are entertained in *TMPAD* as it allocates GTSs to nodes efficiently to transmit more data. However, the other standards have the same number of GTS-requesting nodes because they have already fulfilled all of the available GTSs in the session.

Results in Figures 9 and 10 verify that *TMPAD* allocates the same number of available GTSs to more GTS-requesting nodes as compared to the other three algorithms for varying channel bandwidth and varying transmission power of the nodes. Results in Figure 9 show that the proposed *TMPAD* entertains more legitimate nodes in allocating GTSs as compared to all three standards. The increase in channel bandwidth allows nodes to send fewer GTSs for the same amount of data. This results in allowing more GTS-requesting nodes to assign their requested GTSs in a communication session. However, for all varying channel bandwidths, *TMPAD* allocates time slots to more GTS-requesting nodes as compared to the other three standards. The same trend follows for varying transmission power of nodes, as shown in Figure 10. It is evident from the results that the *TMPAD* assigns the same number of GTSs to more GTS-requesting nodes for all different transmission power levels.

The increase in power allows more data transmission capacity within a time slot, which allows nodes to transmit the same amount of data in fewer time slots, resulting in more GTS-requesting nodes to be assigned the same number of GTSs within a current session.

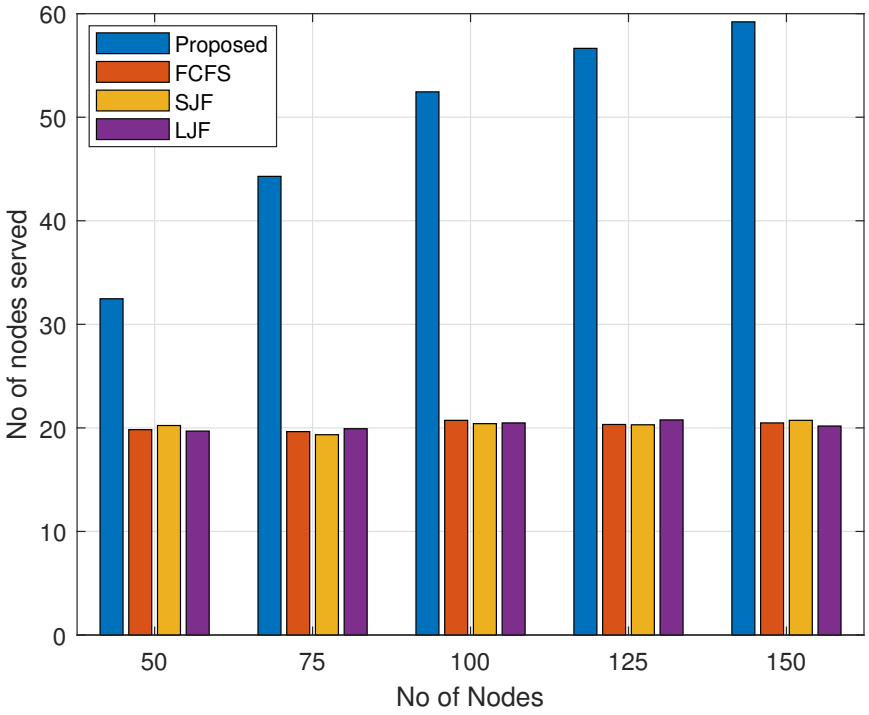

Figure 8. Number of GTS-allocating nodes for varying number of GTS-requesting nodes.

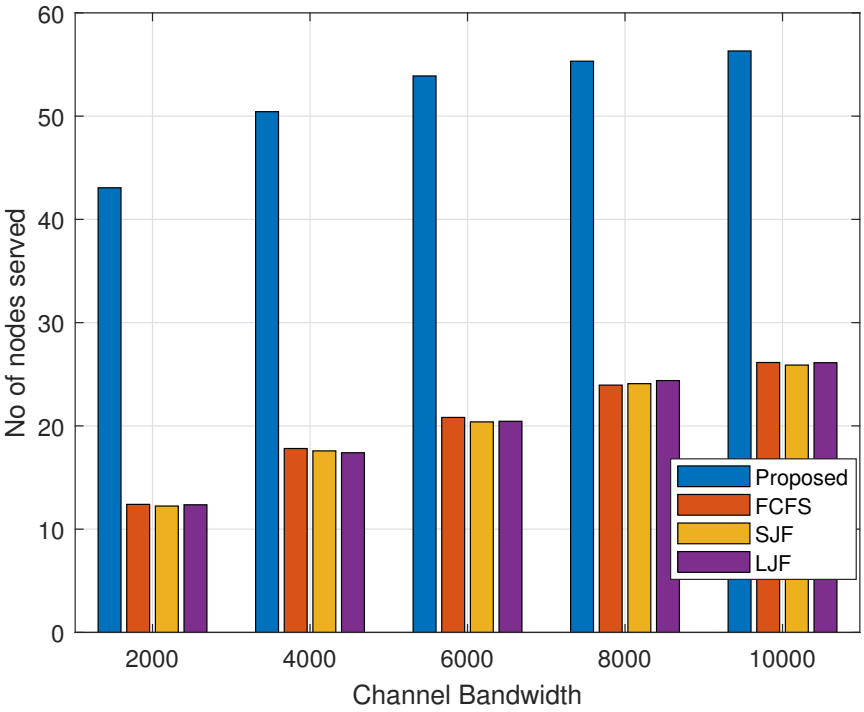

Figure 9. Number of GTS-allocating nodes for varying channel bandwidth of GTS-requesting nodes.

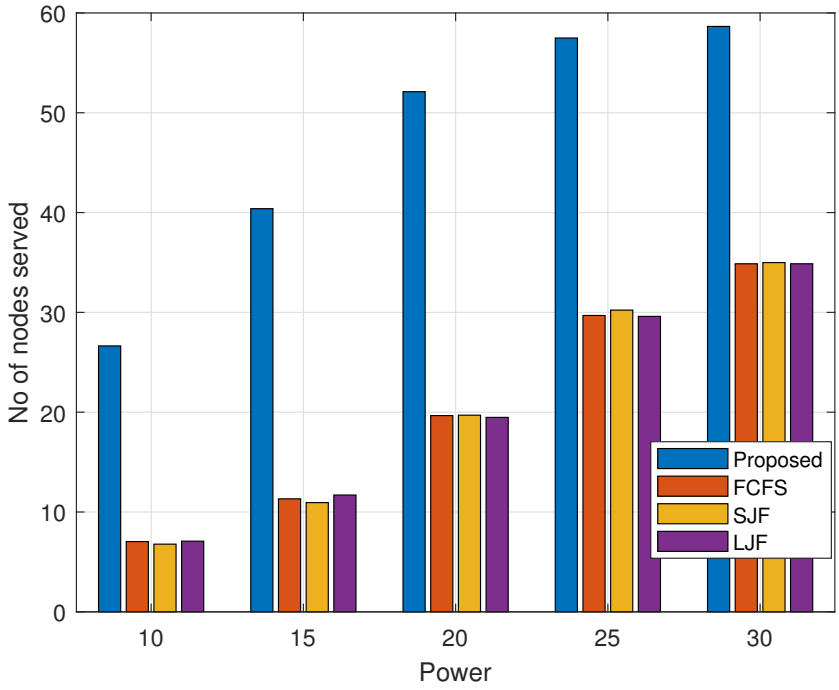

Figure 10. Number of GTS-allocating nodes for varying transmission power of GTS-requesting nodes.

4.3. Transmission Delay

The data transmission delay of a node is calculated when a node has data to transmit until it successfully transmits its data in its allocated time slot. If a data-requesting node is not entertained in the current session then it will resend the GTS request again in the next session. However, its data transmission time will be calculated from the time when it originates its first GTS request to the fog node. The data transmission time in a session is calculated for only those nodes that are successfully allocated GTSs in the current communication session. The results are observed for varying numbers of GTS-requesting nodes, for varying channel bandwidth, and varying transmission power, as shown in Figures 11, 12 and 13, respectively.

Results in Figure 11 show the accumulated transmission time of all GTS-allocating nodes in a session for varying numbers of nodes. The results show that the nodes' transmission time in TMPAD is longer than the other three schemes. This is because the TMPAD prioritizes GTS-requesting nodes that could not transmit their data in the previous session/s, and the longer time a node has to wait, the greater will be its priority towards the GTS allocation. This results in increased data transmission time of the nodes during that session. However, all other algorithms decide to scrutinize the GTS-requesting nodes by considering their current session request. The results show that in *TMPAD*, data transmission time is more than the other three standards. The same trend is observed in Figure 12 when 100 nodes are sending GTS requests to a fog computing node in a session with varying channel bandwidth. With an increase in channel bandwidth, nodes can transmit more data in a slot, resulting in less requirement of GTSs in transmitting the same amount of data. This allows more nodes to be allowed to transmit their data in the allocated time slots in a session, prioritizing the waiting nodes and, consequently, increasing the accumulated transmission time of nodes.

Results in Figure 13 are observed for 100 GTS-requesting nodes with varying transmission power. The results show that with the increase in transmission power, more nodes can transmit their data to the fog node in fewer time slots. This allows more GTS-requesting nodes to transmit their data in a session, resulting in more accumulating transmission time for all successfully allocated GTS nodes.

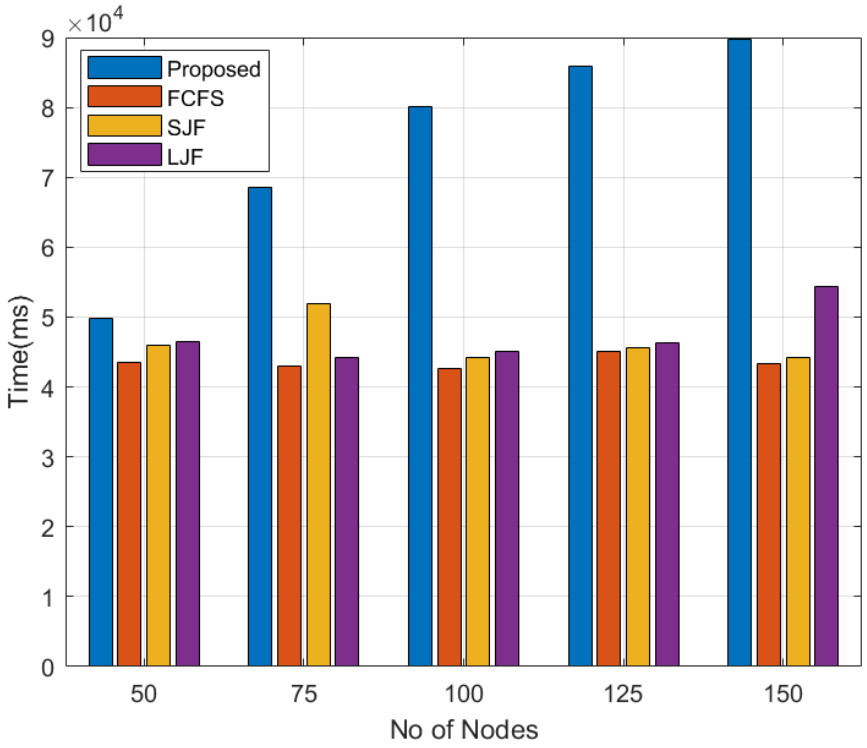

Figure 11. Transmission time of legitimate nodes for varying GTS-requesting nodes.

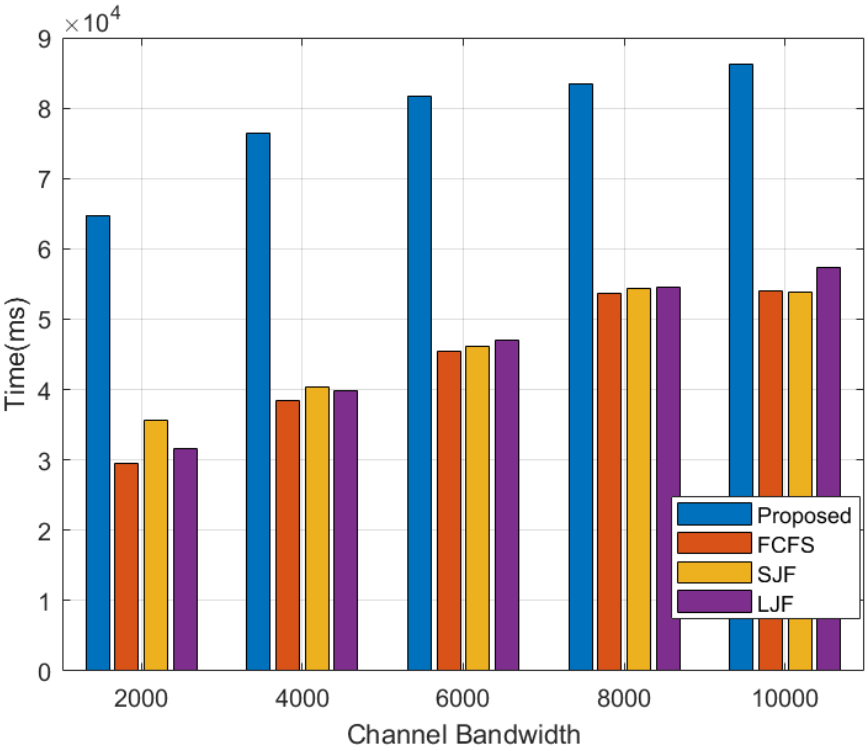

Figure 12. Transmission time of legitimate nodes for varying channel bandwidth.

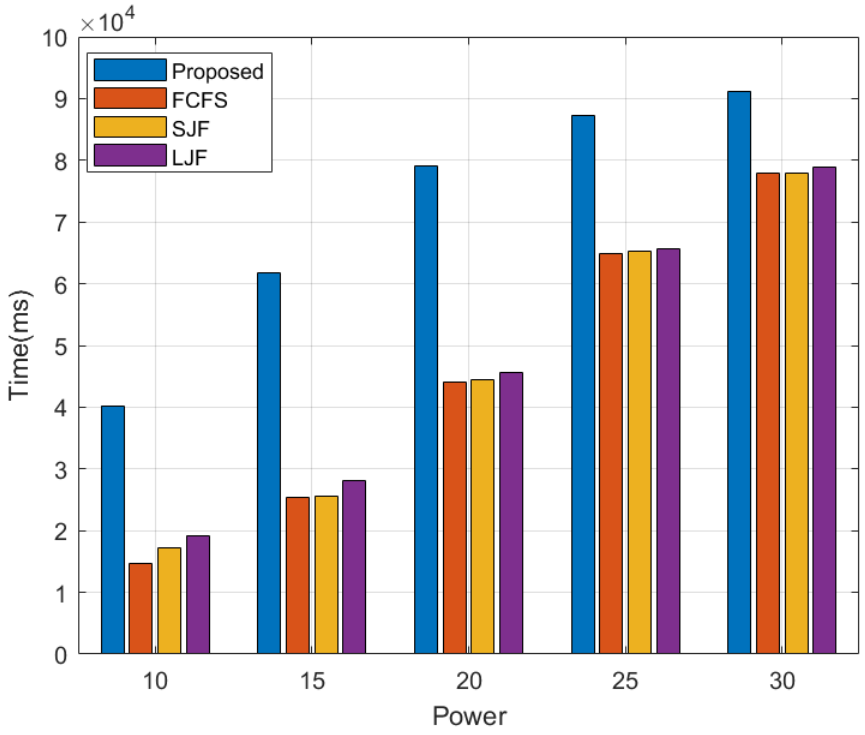

Figure 13. Transmission time of legitimate nodes for varying transmitting power.

4.4. Mean Trust Value

The mean trust value in a session is calculated for all GTS-requesting nodes that have been successfully allocated GTSs. The results are calculated for varying numbers of nodes, for varying bandwidth, and for varying transmission power, as shown in Figures 14, 15 and 16, respectively.

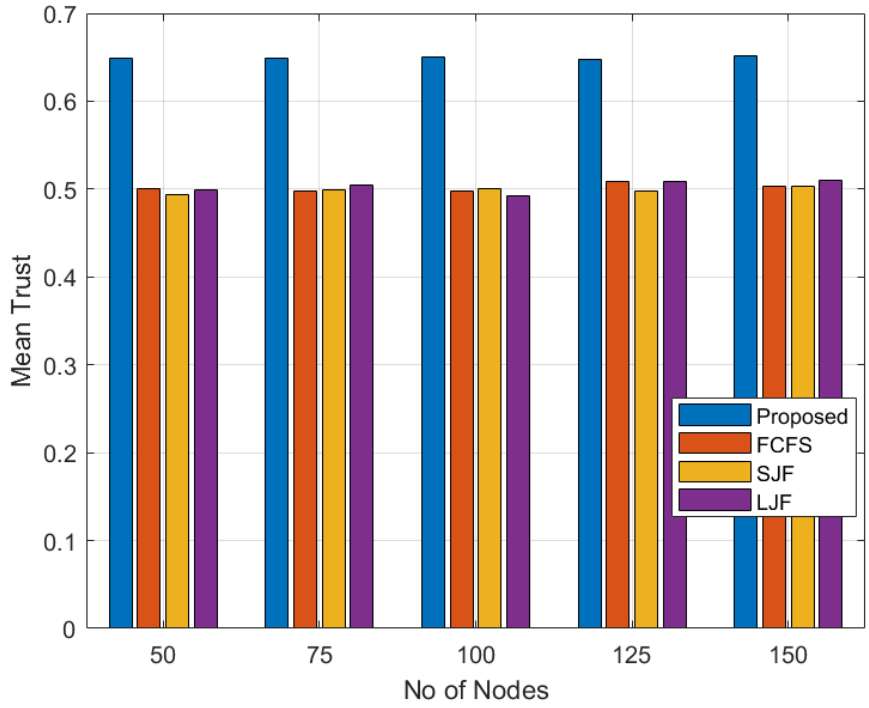

Figure 14. Mean trust value calculated for varying GTS-requesting nodes.

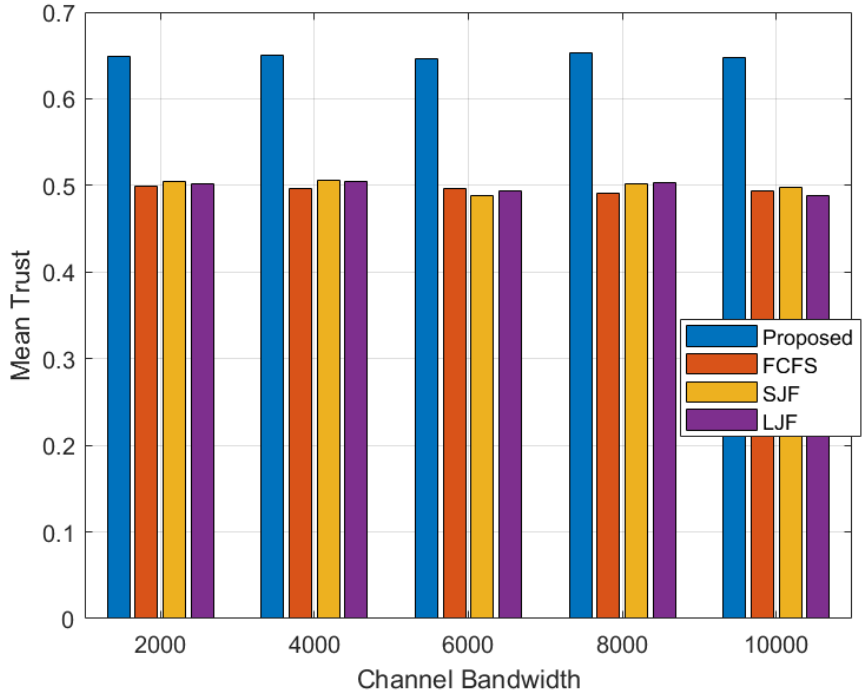

Figure 15. Mean trust value of GTS-requesting nodes for varying bandwidth capacity.

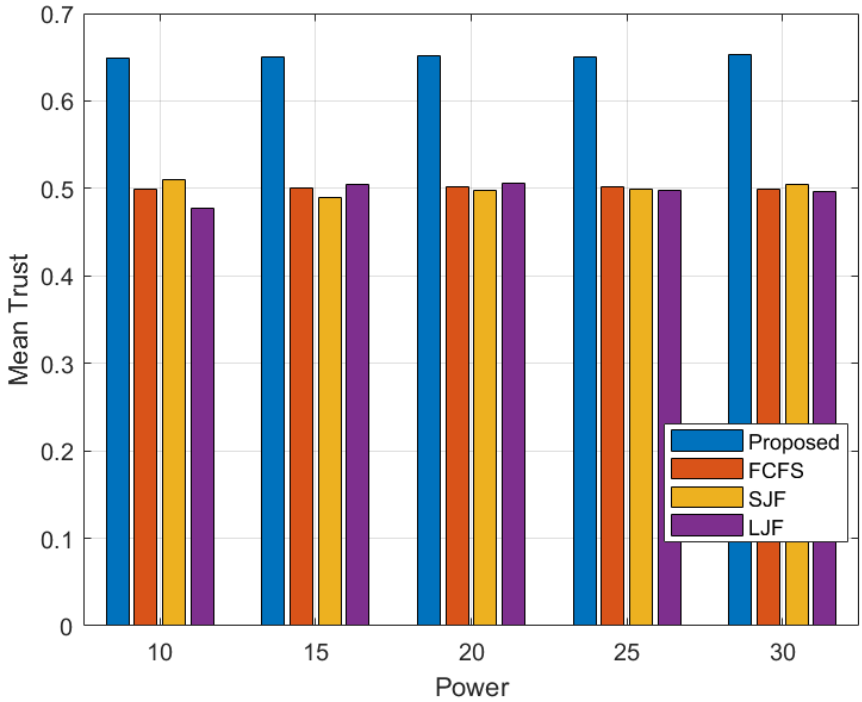

Figure 16. Mean trust value of GTS-requesting nodes for varying transmitting power.

Figure 14 shows that the means trust value of all the successfully allocated nodes in a session in the proposed *TMPAD* is higher than the other three standards for varying numbers of nodes. It is due to the fact that *TMPAD* scrutinizes GTS-requesting nodes in transmitting their data by applying TOPSIS, and trust calculation is one of the important parameters in calculating the TOPSIS value. The results show that with the increase in the number of GTS-requesting nodes, the mean trust value of all GTS-allocating nodes remains similar; however, it is more than the other three standards.

Results in Figures 15 and 16 further show that the mean trust value of all successfully allocated GTS-requesting nodes in *TMPAD* is higher than the other three algorithms for varying channel bandwidth and for varying transmission power, respectively. Figure 15 shows that the mean trust value calculated for increasing channel bandwidth is 0.64, whereas mean trust values in all GTS-allocated nodes of all three algorithms are about 0.5. It means that *TMPAD* assists the fog node in scrutinizing GTS-requesting nodes with higher calculated trust values. However, all other algorithms do not scrutinize GTS-requesting nodes based on their trust values and, consequently, their mean trust value is less than the proposed scheme. The same trend is observed for varying transmission power, as shown in Figure 16. It is evident from the results that the mean trust value calculated in *TMPAD* is significantly greater than the other three schemes, and up to 33% more mean trust value is achieved because *TMPAD* allows fog nodes to scrutinize nodes with higher trust values. However, the other three standards allocate GTS without considering their trust values, and some malicious nodes may also be entertained.

In the end, we evaluated the performance of our proposed scheme in different prospects for varying trust threshold values, as discussed in Algorithm 1. Figure 17 comprises four subfigures, showing transmitted data of legitimate nodes, the number of GTS-requesting nodes entertained, the data transmission time of all the legitimate nodes, and the mean trust values. The results are compared with the 0/1 knapsack algorithm [32] in addition to the other three standards. The results show that the proposed scheme allows more legitimate nodes to transmit their data as compared to the four other schemes. In addition, it allows more data to be transmitted by all legitimate nodes to the fog computing node. Another subplot shows that the time to transmit data by all the selected legitimate nodes is more than the other four schemes. This is due to the reason that the transmission time of only the successfully allocated nodes is calculated, and TMPAD allows more nodes to transmit their data as compared to the other four competing algorithms. The fourth subplot evaluates the mean trust value of the selected nodes that are allowed to transmit their data. The results show that the proposed scheme allocates GTS to those GTS-requesting nodes which have higher trust values as compared to the other four schemes.

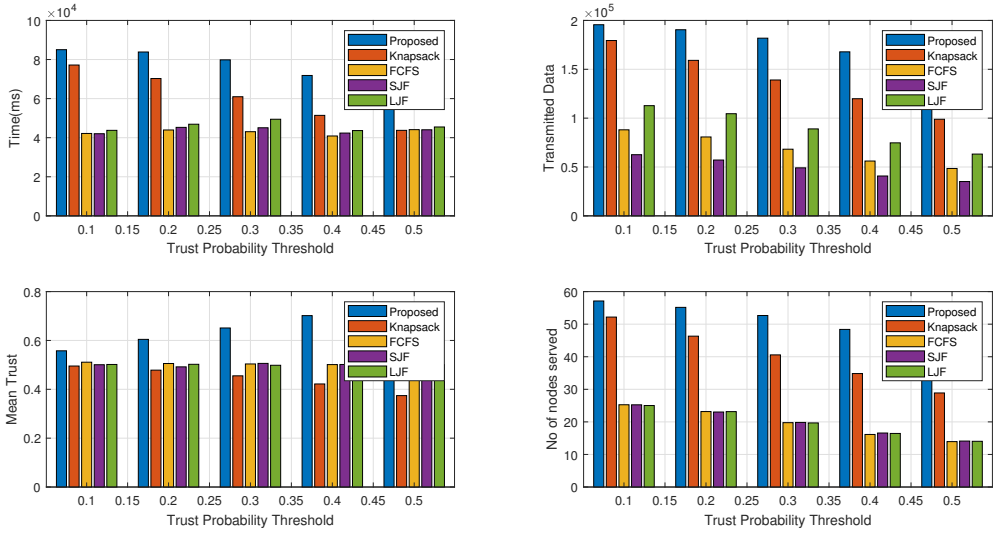

Figure 17. Comparative results of GTS-requesting nodes for trust probability threshold.

5. Conclusions

In this work, a TDMA-based MAC protocol to meet adaptive data requirements (TMPAD) of trusted nodes for fog-enabled smart cities by applying TOPSIS and 0/1 knapsack algorithms is proposed. TMPAD efficiently scrutinizes the GTS-requesting nodes to increase their data transmission in a session, by allowing more number of GTS-requesting nodes to transmit their data with increased mean trust value. The results show

that *TMPAD* allows 110% up to 335% more data transmission in a session than the well-known standards of FCFS, SJF, and LJF for varying numbers of nodes, varying channel bandwidth, and varying transmission power. Furthermore, the proposed scheme entertains up to 188% more GTS-requesting nodes to transmit their data in a session, as compared to the other algorithms. The results further verify that the mean trust value calculated for GTS-allocating nodes in *TMPAD* is up to 33% more than the other three algorithms for varying communication environments. However, the accumulated transmission time of all GTS-allocating nodes in *TMPAD* is more than the other three algorithms because it allocates GTSs to more nodes and, consequently, more nodes require more time to transmit their data. The results show that data transmission time calculated in *TMPAD* is 18% to 167% more, as compared to the other three algorithms with 110% to 335% more data transmission.

Author Contributions: Conceptualization, A.N.A., M.A. (Mumtaz Ali), M.S.S., M.A. (Mohammed Alkhathami), D.A., B.A. (Bushra Alghamdi) and B.A. (Badriya Alenzi); writing—original draft, A.N.A., M.A. (Mumtaz Ali), M.S.S. and M.A. (Mohammed Alkhathami); writing—review and editing, D.A., B.A. (Bushra Alghamdi) and B.A. (Badriya Alenzi). All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported and funded by the Deanship of Scientific Research at Imam Mohammad Ibn Saud Islamic University (IMSIU) (grant number IMSIU-RG23112).

Data Availability Statement: The data presented in this study are available in article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Haseeb, K.; Saba, T.; Rehman, A.; Ahmed, Z.; Song, H.H.; Wang, H.H. Trust Management with Fault-Tolerant Supervised Routing for Smart Cities Using Internet of Things. *IEEE Internet Things J.* **2022**, *9*, 22608–22617. [CrossRef]
2. Bornholdt, H.; Röbert, K.; Kisters, P. Accessing Smart City Services in Untrustworthy Environments via Decentralized Privacy-Preserving Overlay Networks. In Proceedings of the 2021 IEEE International Conference on Service-Oriented System Engineering (SOSE), Oxford, UK, 23–26 August 2021; pp. 144–149. [CrossRef]
3. Yang, J.; Kwon, Y.; Kim, D. Regional Smart City Development Focus: The South Korean National Strategic Smart City Program. *IEEE Access* **2021**, *9*, 7193–7210. [CrossRef]
4. Kwak, Y.H.; Lee, J. Toward Sustainable Smart City: Lessons from 20 Years of Korean Programs. *IEEE Trans. Eng. Manag.* **2023**, *70*, 740–754. [CrossRef]
5. Lytras, M.D.; Şerban, A.C. E-Government Insights to Smart Cities Research: European Union (EU) Study and the Role of Regulations. *IEEE Access* **2020**, *8*, 65313–65326. [CrossRef]
6. Mora-Sánchez, O.B.; López-Neri, E.; Cedillo-Elias, E.J.; Aceves-Martínez, E.; Larios, V.M. Validation of IoT Infrastructure for the Construction of Smart Cities Solutions on Living Lab Platform. *IEEE Trans. Eng. Manag.* **2021**, *68*, 899–908. [CrossRef]
7. Haw, C.Y.; Awang, A.; Hussin, F.A. A Contention-Based MAC and Routing Protocol for Wireless Sensor Network. *Wirel. Sens. Netw.* **2023**, *15*, 1–32. [CrossRef]
8. Masud, F.; Abdul-Salaam, G.; Anwar, M.; Abdelmaboud, A.; Malik, M.S.A.; Ab Ghani, H.B. Contention-based traffic priority MAC protocols in wireless body area networks: A thematic review. *Egypt. Inform. J.* **2023**, *24*, 100410. [CrossRef]
9. Alvi, A.N.; Bouk, S.H.; Ahmed, S.H.; Yaqub, M.A.; Sarkar, M.; Song, H. BEST-MAC: Bitmap-Assisted Efficient and Scalable TDMA-Based WSN MAC Protocol for Smart Cities. *IEEE Access* **2016**, *4*, 312–322. [CrossRef]
10. Aman, W.; Snekkenes, E. Managing security trade-offs in the Internet of Things using adaptive security. In Proceedings of the International Conference for Internet Technology and Secured Transactions, London, UK, 14–16 December 2015; pp. 362–368.
11. Khan, S.; Alvi, A.N.; Javed, M.A.; Bouk, S.H. An enhanced superframe structure of IEEE 802.15.4 standard for adaptive data requirement. *Comput. Commun.* **2021**, *169*, 59–70. [CrossRef]
12. Alvi, A.N.; Khan, S.; Javed, M.A.; Konstantin, K.; Almagrabi, A.O.; Bashir, A.K.; Nawaz, R. OGMAD: Optimal GTS-Allocation Mechanism for Adaptive Data Requirements in IEEE 802.15.4 Based Internet of Things. *IEEE Access* **2019**, *7*, 170629–170639. [CrossRef]
13. Swaminathan, K.; Ravindran, V.; Ponraj, R.P.; Venkatasubramanian, S.; Chandrasekaran, K.S.; Ragunathan, S. A Novel Composite Intrusion Detection System (CIDS) for Wireless Sensor Network. In Proceedings of the 2023 International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT), Bengaluru, India, 5–7 January 2023; pp. 112–117. [CrossRef]
14. Dhelim, S.; Aung, N.; Kechadi, M.T.; Ning, H.; Chen, L.; Lakas, A. Trust2Vec: Large-Scale IoT Trust Management System Based on Signed Network Embeddings. *IEEE Internet Things J.* **2023**, *10*, 553–562. [CrossRef]

15. Lewis, C.; Li, N.; Varadharajan, V. Targeted Context-Based Attacks on Trust Management Systems in IoT. *IEEE Internet Things J.* **2023**, *10*, 12186–12203. [CrossRef]
16. Khandelwal, N.; Gupta, S. A Review: Trust based Secure IoT Architecture in Mobile Ad-hoc Network. In Proceedings of the 2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC), Salem, India, 9–11 May 2022; pp. 1464–1472. [CrossRef]
17. Bai, J.; Dong, H. Federated Learning-driven Trust Prediction for Mobile Edge Computing-based IoT Systems. In Proceedings of the 2023 IEEE International Conference on Web Services (ICWS), Chicago, IL, USA, 2–8 July 2023; pp. 131–137. [CrossRef]
18. Iftikhar, A.; Qureshi, K.N.; Altalbe, A.A.; Javeed, K. Security Provision by Using Detection and Prevention Methods to Ensure Trust in Edge-Based Smart City Networks. *IEEE Access* **2023**, *11*, 137529–137547. [CrossRef]
19. Fan, J.; Yang, W.; Liu, Z.; Kang, J.; Niyato, D.; Lam, K.Y.; Du, H. Understanding Security in Smart City Domains from the ANT-Centric Perspective. *IEEE Internet Things J.* **2023**, *10*, 11199–11223. [CrossRef]
20. Kim, D.Y.; Alodadi, N.; Chen, Z.; Joshi, K.P.; Crainiceanu, A.; Needham, D. MATS: A Multi-aspect and Adaptive Trust-based Situation-aware Access Control Framework for Federated Data-as-a-Service Systems. In Proceedings of the 2022 IEEE International Conference on Services Computing (SCC), Barcelona, Spain, 10–16 July 2022; pp. 54–64. [CrossRef]
21. Guo, J.; Liu, A.; Ota, K.; Dong, M.; Deng, X.; Xiong, N.N. ITCN: An Intelligent Trust Collaboration Network System in IoT. *IEEE Trans. Netw. Sci. Eng.* **2022**, *9*, 203–218. [CrossRef]
22. Dang, T.D.; Hoang, D.; Nguyen, D.N. Trust-Based Scheduling Framework for Big Data Processing with MapReduce. *IEEE Trans. Serv. Comput.* **2022**, *15*, 279–293. [CrossRef]
23. Shen, X.; Lv, W.; Qiu, J.; Kaur, A.; Xiao, F.; Xia, F. Trust-Aware Detection of Malicious Users in Dating Social Networks. *IEEE Trans. Comput. Soc. Syst.* **2022**, *10*, 2587–2598. [CrossRef]
24. Varadharajan, V.; Karmakar, K.K.; Tupakula, U.; Hitchens, M. Toward a Trust Aware Network Slice-Based Service Provision in Virtualized Infrastructures. *IEEE Trans. Netw. Serv. Manag.* **2022**, *19*, 1065–1082. [CrossRef]
25. Ebrahimi, M.; Haghighi, M.S.; Jolfaei, A.; Shamaeian, N.; Tadayon, M.H. A Secure and Decentralized Trust Management Scheme for Smart Health Systems. *IEEE J. Biomed. Health Inform.* **2022**, *26*, 1961–1968. [CrossRef] [PubMed]
26. Khan, S.; Alvi, A.N.; Khan, M.Z.; Javed, M.A.; Alhazmi, O.H.; Bouk, S.H. A novel superframe structure and optimal time slot allocation algorithm for IEEE 802.15.4–based Internet of things. *Int. J. Distrib. Sens. Netw.* **2020**, *16*, 1–13. [CrossRef]
27. Alvi, A.N.; Javed, M.A.; Hasanat, M.H.A.; Khan, M.B.; Saudagar, A.K.J.; Alkhathami, M.; Farooq, U. Intelligent Task Offloading in Fog Computing Based Vehicular Networks. *Appl. Sci.* **2022**, *12*, 4521. [CrossRef]
28. Alvi, A.N.; Ali, B.; Saleh, M.S.; Alkhathami, M.; Alsadie, D.; Alghamdi, B. TETES: Trust Based Efficient Task Execution Scheme for Fog Enabled Smart Cities. *Appl. Sci.* **2023**, *13*, 12799. [CrossRef]
29. Soni, G.; Selvaradjou, K. Optimal GTS distribution to heterogeneous sensors in IEEE 802.15.4 network for healthcare monitoring applications. *Pers. Ubiquitous Comput.* **2022**, *26*, 131–153. [CrossRef]
30. Tareen, F.N.; Alvi, A.N.; Malik, A.A.; Javed, M.A.; Khan, M.B.; Saudagar, A.K.J.; Alkhathami, M.; Abul Hasanat, M.H. Efficient Load Balancing for Blockchain-Based Healthcare System in Smart Cities. *Appl. Sci.* **2023**, *13*, 2411. [CrossRef]
31. Harms, O.; Landsiedel, O. MASTER: Long-Term Stable Routing and Scheduling in Low-Power Wireless Networks. In Proceedings of the 2020 16th International Conference on Distributed Computing in Sensor Systems (DCOSS), Marina del Rey, CA, USA, 25–27 May 2020; pp. 86–94. [CrossRef]
32. Wan, Z.; Dong, X. Computation power maximization for mobile edge computing enabled dense network. *Comput. Netw.* **2023**, *220*, 109458. [CrossRef]