Journal Homepage: -www.journalijar.com

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH (IJAR)

Article DOI:10.21474/IJAR01/12402
DOI URL: http://dx.doi.org/10.21474/IJAR01/12402

RESEARCH ARTICLE

DEFENSE STRATEGY AGAINST SELFISH NODES TAKING INTO ACCOUNT TRAFFIC DIFFERENTIATION IN VEHICULAR DELAY-TOLERANT NETWORKS

Gballou Yao Théophile[1,2], Touré Kidjegbo Augustin[1] and Tiecoura Yves[1,2]

1. Ecole Doctorale Polytechnique de l'INP-HB Yamoussoukro, Côte d'Ivoire.
2. Laboratoire De Recherche En Informatique et Télécommunication (LARIT).

Manuscript Info

Manuscript History
Received: 27 November 2020
Final Accepted: 30 December 2020
Published: January 2021

Key words:-
Vehicular Delay-Tolerant Networks, Selfish Node, Selfish Degree, Transmission Rate, Priority Class Of Service

Abstract

Vehicular Delay-Tolerant Networks (VDTNs) are vehicle networks where there is no end-to-end connectivity between source and destination. As a result, VDTNs rely on cooperation between the different nodes to improve its performance. However, the presence of selfish nodes that refuse to participate in the routing protocol causes a deterioration of the overall performance of these networks. In order to reduce the impact of these selfish nodes, proposed strategies, on the one hand, use the nodes' transmission rate that does not take into account the message priority class of service, and on the other hand, are based on traditional buffer management systems (FIFO, Random). As a result, quality of service is not guaranteed in this type of network where different applications are derived from messages with different priorities. In this paper, we propose a strategy for detecting selfish nodes and taking action against them in relation to priority classes in order to reduce their impacts. The operation of this strategy is based, on a partitioned memory management system taking into account the priority and the lifetime of messages, on the calculation of the transmission rate of the node with respect to the priority class of the node with the highest delivery predictability, on a mechanism for calculating the node's degree of selfishness with respect to the priority class, and on the monitoring mechanism. . The simulations carried out show that the proposed model can detect selfish nodes and improve network performance in terms of increasing the delivery rate of high-priority messages, reducing the delivery delay of high-priority messages, and reducing network overload.

Introduction:-

VDTN (Vehicular Delay-Tolerant Networks) are presented as a new alternative to vehicular communication based on the SCF (Store-Carry-and-Forward) paradigm of DTN (Delay-Tolerant Networks) to solve the problems of intermittent connectivity, high error rate, high latency, high asymmetric data rate and lack of end-to-end connectivity [1][2]. VDTNs are used in several areas such as remote rural connectivity, business information dissemination, road traffic management such as disaster area communication media, and road safety. All these applications have different requirements in terms of delivery time and message delivery rates. However, VDTNs support the different applications simultaneously and asynchronously by classifying the traffic into three priority

Corresponding Author:- Gballou Yao Théophile
Address:- Ecole Doctorale Polytechnique de l'INP-HB Yamoussoukro, Côte d'Ivoire.

classes of service: low priority for bulk messages, medium priority for normal messages, and high priority for expedited messages [3].

VDTNs are characterized by a layered architecture that places a bundle layer below the network layer of the Open Systems Interconnection (OSI) model to facilitate the routing of large messages as opposed to small IP packets. In addition, this VDTN architecture considers out-of-band signaling with a separation of the bundle layer into two planes: the control plane and the data plane. For this purpose, before the transfer of messages between two intermediate nodes or from an intermediate node to the final destination, a prior cooperation is worked out between the nodes. Thus, at the level of the control plane, the nodes exchange signaling information (node storage capacity, speed, geographical location, contact date and duration) in order to make settings, to plan the period of availability of the contact and to reserve resources for the proper transmission of messages between the nodes. In addition, at the data plane level, functions such as memory management, traffic prioritization and message transmission are performed [2] [4].

However, despite all these precautions to ensure better message transmission, the presence of selfish nodes affects network performance in terms of reduced delivery rate, increased latency and increased network overload [5][6][7][8]. Indeed, these selfish nodes, in order to save their resources (buffer space, bandwidth, energy), are not willing to relay messages from other nodes, yet they exploit and consume network resources [9][10][11][12].

Recently, in order to detect selfish nodes and then reduce their impact on overall network performance, several works have been based on the node's transmission rate by comparing for each node the total number of messages transferred to the total number of messages received. Thus, if the value of the node's transmission rate is less than one, then the node is selfish. Otherwise, the node is cooperative. This calculation of the transmission rate is only applicable when all messages have the same priority. Since, when traffic differentiation occurs, then a node may appear cooperative without relaying messages of a certain priority class. In addition, a node may appear selfish when it has the ability to relay messages of a certain priority class.

Moreover, in order to improve cooperation in buffer management, recent strategies require nodes to reserve a percentage of their buffer for cooperation [6]. However, these strategies are based on traditional buffer management systems such as Random or First-In-First-Out (FIFO) methods, which do not guarantee a short message delivery time [13].

In this paper, we propose a new strategy, taking into account traffic differentiation, to detect the selfishness of a node with respect to priority classes and to act on these selfish nodes. To do this, we use a partitioned memory management system based on message weight, which takes into account the message time-to-live and priority class in order to guarantee both the buffer cooperation threshold and the quality of service. In addition, taking into account the most predictable delivery of the nodes, we calculate the transmission rate of the node relative to the priority class in order to know whether the node is cooperative or selfish relative to the priority class.

When the node is selfish with respect to a priority class, then the calculation of the degree of selfishness of that node with respect to that priority class is performed to check whether that node is partially or fully selfish with respect to the priority class. An entirely selfish node is isolated relative to the priority class. On the other hand, a partially selfish node is monitored for a certain period of time to ensure that its behavior changes. If this node becomes cooperative, then it can continue communication, otherwise it cannot receive messages from the priority class.

The main contributions of this paper are as follows:
A state of the art of the most important contributions of selfish node detection systems based on the node's residual energy and buffer management system, systems based on the degree of selfishness, and systems based on the transmission rate.

The proposal of a defense strategy against selfish nodes based on a partitioned memory management system and taking into account the class of priority service, calculating the transmission rate relative to the priority class, determining the degree of selfishness of the node relative to the priority class, and monitoring the node.

Performance evaluations to assess the impact of the proposed strategy compared to a strategy without detection of selfish nodes. These evaluations are done using the Opportunistic Network Simulator (ONE) considering as

performance metrics: the high priority message delivery rate, the average high priority message delivery time and the network overload.

The rest of the paper is organized as follows: Section 2 presents a state of the art of node detection and action strategies on selfish nodes. The proposed node defense strategy approach is presented in section 3. Section 4 presents evaluations of the impacts of the number of selfish nodes on network performance. Finally, the conclusion and perspectives are elaborated in Section 5.

Related Works:
As mentioned above, the presence of selfish nodes in the network seriously affects network performance in terms of increasing delivery delay and reducing the rate of messages delivered to the destination. As a result, routing protocols must determine the best nodes to transfer messages despite the presence of selfish nodes. Several strategies have been developed to detect selfish nodes and then act on these nodes to improve the overall performance of the network. In this section, we provide an overview of related work to this field.

Detection based on the residual energy of the node:
The authors in [15] proposed a routing algorithm called MPSSR (Message-Priority Socially-Selfishly Routing). This algorithm combines the degree of centrality of the node, the residual energy of the node, and a buffer management based on the priority class to detect the best relay nodes. In this algorithm, during an opportunistic contact, a node transfers its messages to the node with the highest degree of centrality when the number of message hops from the receiver is less than three (3). On the other hand, if the number of hops in the receiver's messages is greater than three (3), then three situations are defined. If the residual energy of the receiver is very insufficient, then it is chosen as the destination of the transmitter messages. However, if the power of the receiver is limited, then the receiver and sender are in the same community and the receiver with greater predictability of delivery receives messages in their order of priority. In addition, if the sender has sufficient energy and buffer space, then the receiver with the highest predictability of delivery receives messages in the order of priority.

This strategy shows that the detection of selfish nodes can be combined with buffer management that takes into account the priority class of service to guarantee the quality of service. However, since the vehicles have sufficient energy supplied by the batteries [29], then node selfishness based on the node's residual energy is not directly applicable to vehicles in a vehicle-to-vehicle communication architecture.

Strategies based on the selfish degree:
In [16] [17] [18], the authors proposed strategies for detecting selfish nodes based on their selfish degree. Thus in [16], the authors proposed in their algorithm, a decision method named *wantToCollaborate()* which allows to check the node behavior. This node behavior is indicated by comparing the selfish degree of the node which is represented by a random number taken between 0 and 100, where 0 indicates that the node is cooperative and 100 indicates that the node is entirely selfish. This strategy detects selfish nodes and reduces message loss. However, it is ineffective as the number of nodes increases. In [17], the authors proposed an algorithm that classifies nodes into three categories according to their selfish degree. Thus, cooperative nodes have a selfish degree less than or equal to zero, partially cooperative nodes have a selfish degree between 0 and 100, and fully selfish nodes have a selfish degree equal to 100. In addition, cooperative nodes are induced with an energy input. This strategy improves network performance, but the delivery rate decreases as the number of selfish nodes increases. Similarly, in [18], Sunil K. proposed an algorithm based on a function to ensure the functioning of the node and taking into account the selfish degree of the node. Thus, when the node cooperates in normal operation, then its selfish degree is constant. Otherwise, it increases. In addition, by taking into account the selfish degree, cooperative nodes, partially selfish nodes and fully selfish nodes are detected. Entirely selfish nodes are isolated by the function described above. This strategy improves the performance of the protocol. However, the strategy does not take any action on partially selfish nodes. These strategies show the existence of three categories of nodes (cooperative, partially selfish, entirely selfish) in the network. However, these strategies use traditional buffer management systems (Radom; FIFO) that are less reducing the message delivery delay.

Strategies based on node transmission rate:
In [19] [20] [21], the authors use the transmission rate of the node to determine its behavior in the network, on the one hand, and to guarantee the reliability of the transmitted messages, on the other hand. Thus, in [19], the authors propose a strategy called CFV (Combined-Faith Value) in order to detect the reliability of the next relay. For this

purpose, before transferring a message to a node, the sending node calculates the CFV of the node from the information of its past performance (the number of messages transferred and the number of messages received) obtained from its neighbors and the number of messages created by this node since the beginning of the communication. Thus, if a node's CFV is below a threshold, then the node is isolated and consequently another node is chosen. This process is repeated until the message reaches the destination. However, CFV uses a threshold value that is specific only to the Spray-and-Wait routing protocol [26]. Similarly in [20], the authors proposed a routing algorithm called TBER (trust-based Epidemic routing). TBER determines the next relay by calculating its global confidence. The global confidence is composed of the direct confidence and the recommended confidence. The direct confidence is based on the energy, altruism and connectivity of the next relay. Altruism is based on the transmission rate of the node. As for the recommended trust, it is determined by the set of nodes in the neighborhood of the next relay. This mechanism allows path construction by eliminating non-cooperative nodes. In addition, it promotes an improvement in the delivery rate. However, it is based on the epidemic routing protocol, which leads to an increase in network overload [24]. Bhoiwala J. P. et al. proposed in [21] a strategy called CBDM (Cooperation-Based Defense Mechanism). This strategy, on the one hand obliges nodes to share a percentage of their buffer capacity [6] [22] [23], and on the other hand uses the combination of the transmission rate of the node and the performance of the routing protocol PRoPHET [14], to determine which cooperative and non-cooperative nodes have the highest predictability of delivery. In addition, a dummy message is used to check the honesty of the nodes. If the node is honest, then it is monitored for a period of time to ensure that its behavior changes. This strategy leads to network improvement in terms of reducing the number of selfish nodes, reducing the number of message losses, and reducing network overload. However, the monitoring mechanism is applied to all non-cooperative nodes regardless of their degree of selfishness. As a result, we can observe an unnecessary waste of time especially when the node is completely selfish.

Based on the above research, by combining the benefits and making improvements, we propose a new strategy of defense against selfish nodes taking into account the traffic differentiation required in these networks. Thus, the new strategy of defense against selfish nodes taking into account the priority class of service, is also based on the combination of the performance of the PRoPHET routing protocol and the transmission rate of the node, on the one hand, and on the other hand, it is based on the degree of selfishness of the node and the monitoring mechanism.
In the following section, we detail the proposed defense strategy against selfish nodes.

Proposed Defense Strategy Approch:

Basic Assumptions:

In this model, we assume that each node holds a table with each line indicating the ID of the node encountered, the number of messages sent to this node and the number of messages received from this node by priority class. Thus, at each contact opportunity between two nodes, each node updates its table. In addition, in this model we do not admit any trusted authority or fixed relay nodes. Message exchanges take place directly between the mobile nodes when they are within communication range of each other. Furthermore, we assume that all nodes in the network have the same buffer capacity.

The Partitionned Memory Management System:

In this model, each node is obliged to share a percentage of its buffer [6]. Thus, the buffer is divided into two parts. One part will be used to store messages intended for the node (NSS: Node Storage Space), while the other part will be used to store messages to be transferred to other nodes (NCS: Node Cooperation Space) [27] in order to ensure a buffer cooperation threshold. However, the available buffer space in general, and the two spaces (NSS and NCS) in particular, can decrease considerably due to excessive replication of multi-copy routing protocols and storage of messages for varying lengths of time [24]. Therefore, we develop a buffer management system to guarantee the buffer cooperation threshold on the one hand, and to guarantee the quality of service on the other hand.

The message weight:

In [13], it is shown that buffer management based on message time-to-live (TTL) reduces the delivery delay better than traditional buffer management strategies. Therefore, taking into account the priority of messages due to the existence of several services with different requirements, we propose a metric called message weight. This metric consists of message TTL and message priority to ensure quality of service for the different applications supported.

The weight of the message is given by the equation (1) :

$$\varpi_j = \log(1+100\times\frac{TTL_r}{TTL_0}) + P_i \quad j\in\{1,2,...,n\},\ i\in\{1,2,3\} \quad (1)$$

Where TTL_r is the remaining time-to-live of the message, TTL_0 is the initial time-to-live of the message and P_i ($i\in\{1,2,3\}$) is the priority class of the message.

The Drop Policy:

The drop policy will apply to the storage space for messages destined for the node (NSS: Node Storage Space), and to the cooperation space for storage and transfer of the node's messages (NCS: Node Cooperation Space) to guarantee the node's cooperation threshold.

This drop policy removes messages from the NSS according to the increasing remaining time-to-live, if the current node is the destination of the message. However, if two messages have the same time-to-live, then the lower priority message is deleted first [28]. On the other hand, if the current node is an intermediate node, then the NCS messages are deleted according to their increasing weight. However, if two messages have the same weight value, then the lower priority message is dropped first.

Figure 1:- Below illustrates the activity diagram of the buffer policy when the buffer is partitioned into two spaces: ESN and ECN.

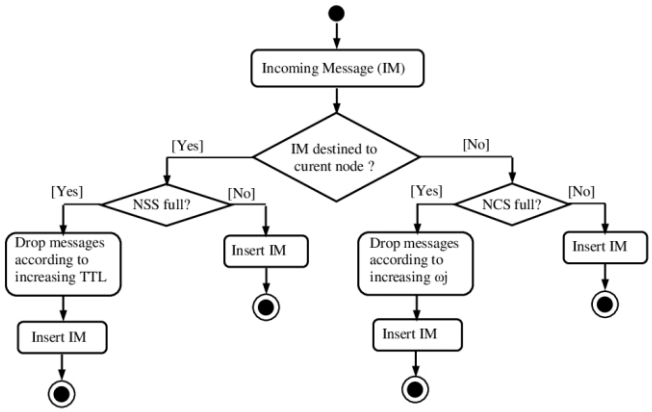


Figure 1. Flow chart of Drop Policy

Scheduling policy:

In our model, the scheduling policy orders the NCS messages according to descending weight. Thus, in opportunistic contacts, the message with the highest weight is transferred first. However, if two messages have the same weight value, then the high priority message is transferred first.

Classification based on Transmission Ration by Priority Class:

In VDTNs, selfish nodes drop messages from other nodes in the network. As a result, their presence in the network reduces the routing protocol performance. The main idea of our model is to allow a sender node to check the reliability of a receiver node before forwarding the message of a given priority class to it. Thus, we determine the transmission ratio of the message according to the priority class of the message that the node must receive.

This message transmission ratio per priority class is the ratio of the number of messages transferred from a priority class to the number of messages received from that priority class. It is given by the expression:

$$\theta_i = \frac{\sum_{m=0}^{n} F_m^i}{\sum_{m=0}^{n} R_m^i} \qquad i\in\{1;2;3\} \quad (2)$$

Where $\sum_{m=0}^{n} F_m^i$ is the sum of messages of a priority class sent by the node to other nodes in the network, but of which it is not the source. $\sum_{m=0}^{n} R_m^i$ is the sum of messages of a priority class received by the node, but not destined

for this node. And P_i ($i \in \{1, 2, 3\}$) determines the priority class for low priority P_1, medium priority P_2 and high priority P_3 messages.

Thus, if the message transmission ratio per priority class P_i ($i \in \{1; 2; 3\}$) is greater than or equal to 1, then the node is cooperative with respect to this priority class (CN P_i). Otherwise, the node is selfish node relative to this priority class (SN P_i).
The following diagram (Figure 2.) illustrates the classification of nodes with respect to a priority class P_i ($i \in \{1; 2; 3\}$).

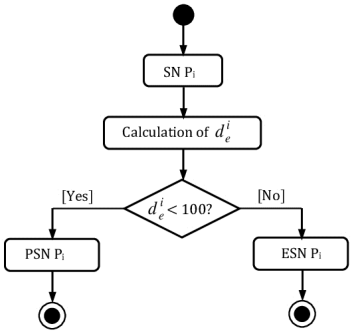


Figure 2. :-The nodes classification according to the transmission ratio by priority class.

Furthermore, since the traffic consists of emergency messages of high priority P_3, normal messages of medium priority P_2 and bulk messages of low priority P_1 [3], then we can determine the behavior of nodes in the network with respect to messages of these priority classes.

Nodes Behavior in VDTN:
In our model, we develop the classification of the nodes taking into account simultaneously the values of θ_1, θ_2 and θ_3. Thus, we have cooperative nodes, both cooperative and selfish nodes, and selfish nodes relative to the three priority classes.

Cooperative node according to all Priority Classes
The node is cooperative when for all priority classes P_i ($i \in \{1, 2, 3\}$), the number of sent messages of a priority class is greater than or equal to the number of received messages of that priority class. The cooperative node is characterized by the expression given by:

$$\theta_1 \geq 1; \theta_2 \geq 1; \theta_3 \geq 1 \qquad (3)$$

Thus, this cooperative node transfers messages of any priority class.

Hybrid Nodes:
Hybrid nodes are nodes that are both cooperative and selfish. These nodes fall into two categories:

The category of nodes that are both cooperative with respect to two priority classes and selfish with respect to one priority class is given by:

- The node is cooperative for low and medium priority messages, but selfish for high priority messages.

$$\theta_1 \geq 1; \theta_2 \geq 1; \theta_3 < 1 \qquad (4)$$

- The node is cooperative for low and high priority messages, but selfish for medium priority messages.

$$\theta_1 \geq 1; \theta_2 < 1; \theta_3 \geq 1 \qquad (5)$$

- the node is cooperative with medium and high priority messages, but selfish with low priority messages

$$\theta_1 < 1; \theta_2 \geq 1; \theta_3 \geq 1 \qquad (6)$$

The category of nodes that are both cooperative relative to one priority class and selfish relative to two priority classes is given by :

- The node is cooperative for high-priority messages, but selfish for low and medium-priority messages

$$\theta_1 < 1; \theta_2 < 1; \theta_3 \geq 1 \qquad (7)$$

- The node is cooperative for medium priority messages, but selfish for low and high priority messages.

$$\theta_1 < 1; \theta_2 \geq 1; \theta_3 < 1 \qquad (8)$$

- The node is cooperative for low priority messages, but selfish for medium and high priority messages.

$$\theta_1 \geq 1; \theta_2 < 1; \theta_3 < 1 \qquad (9)$$

Selfish node according to all priority classes:
The node is selfish node when for all priority classes Pi (i ∈ {1, 2, 3}) the number of sent messages of a priority class is less than the number of received messages of that priority class. Thus, the selfish node is characterized by the expression (10) given by:

$$\theta_1 < 1; \theta_2 < 1; \theta_3 < 1 \qquad (10)$$

This node refuses to forward messages of any priority class.

Diagram of Node Classification according to all Priority Classes:
Thus, taking into account simultaneously the transmission ratios θ_1, θ_2 and θ_3 for low priority P_1, medium priority P_2 and high priority P_3 messages respectively, we propose the classification of all nodes in the network represented by the flow chart in Figure 3.

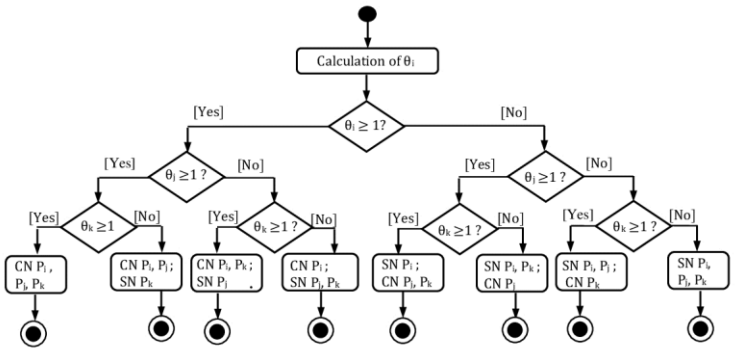


Figure 3. :-Nodes classification in VDTN according to all priority classes.

In the diagram above, {CN (P_i, P_j); SN P_k /i, j, k **∈{1, 2, 3} with i≠j≠k**} denote both cooperative node with respect to messages of priority classes P_i and P_j and selfish node with respect to messages of priority class P_k.

Classification and Monitoring of Selfish Nodes:
In our model, a hybrid node cannot be directly isolated if it is not cooperative with respect to a priority class. We give it another chance to become cooperative relative to the priority class for which it is selfish. Before giving it a chance, we determine its degree of selfishness relative to that priority class. In [17] [18], a selfish node can be partially selfish or completely selfish. Thus, in our model, we develop a classification of selfish nodes according to their degree of selfishness in order to determine the partially selfish nodes and the fully selfish nodes with respect to the priority classes.

Selfish nodes classification according to the degree of selfishness:
In the network, not all nodes that behave selfishly with respect to a priority class have the same willingness to refuse to forward messages of a priority class from other nodes. Some nodes tend to refuse the transfer of a larger number of messages of a certain priority class than other nodes. Thus, to distinguish these nodes, we calculate the degree of selfishness of the node with respect to messages of a priority class Pi (i ∈{ 1, 2, 3}).
The expression of the degree of selfishness relative to the priority class is given by equation (11):

$$d_e^i = 100 \times \frac{\sum_{m=0}^{n} R_m^i - \sum_{m=0}^{n} F_m^i}{\sum_{m=0}^{n} R_m^i} \qquad i \in \{1;2;3\} \qquad (11)$$

Where $\sum_{m=0}^{n} F_m^i$ is the sum of messages of a priority class sent by the node to other nodes in the network, but of which it is not the source. $\sum_{m=0}^{n} R_m^i$ is the sum of messages of a priority class received by the node, but not destined for this node. And P_i ($i \in \{1, 2, 3\}$) determines the priority class for low priority P_1, medium priority P_2 and high priority P_3 messages.

Thus, from expression (11), if the degree of selfish node with respect to the priority class Pi is $\boldsymbol{d_e^i}$ = 100; then this node is entirely selfish with respect to messages of this priority class Pi (ESN Pi). On the other hand, if the degree of selfish node with respect to messages of priority class Pi is between $\boldsymbol{d_e^i}$ = 0 (cooperative node with respect to messages of priority class Pi) and $\boldsymbol{d_e^i}$ =100 (ESN Pi), then this node is partially selfish with respect to messages of this priority class Pi (PSN Pi).

The following diagram illustrates the classification of selfish nodes relative to the priority class (SN P) according to their degree of selfishness relative to this priority class Pi ($\boldsymbol{d_e^i}$)

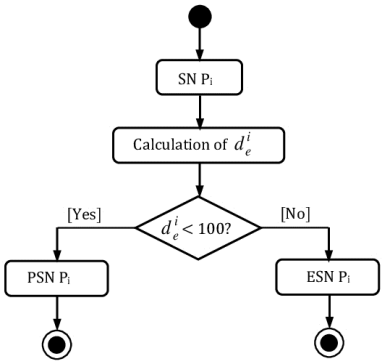


Figure 4. :-Selfish nodes classification accordind to the degree of selfishness relative to the priority class Pi.

Monitoring Mechanism:

In our model, when a node is completely selfish with respect to a priority class, then the other nodes will avoid forwarding messages from that priority class to it. On the other hand, if the node is partially selfish with respect to the messages of a priority class, then this node is monitored for a certain period of time [23]. However, if during the monitoring period, the node becomes cooperative with respect to messages of this priority class, then it can receive the message of this priority class. However, if after the monitoring time, the node does not change its behavior, then it cannot receive messages of this priority class.

The following diagram illustrates the detection of PSN P_i and ESN P_i, and then the monitoring mechanism for PSN P_i.

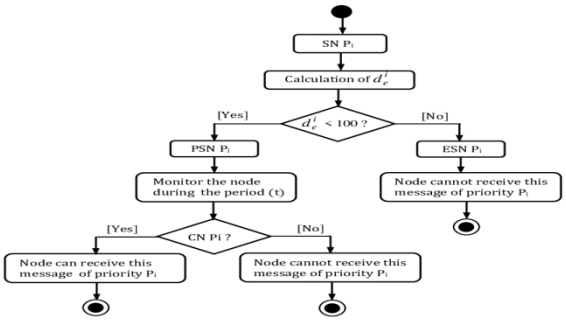


Figure 5. :-Selfish nodes detection and monitoring mechanism in relation to the priority class Pi.

The proposed Defense Strategy:

The proposed defense strategy against selfish nodes is based on the performance of the PRoPHET routing protocol [14]. Thus, when a node wishes to forward a message of priority P_i to an intermediate node, the following points are elaborated:

We check the predictability of delivery of the intermediate node. If this intermediate node has the highest predictability of delivery, then it is selected as the next relay. Otherwise, another node is selected;

We check the behavior of the node with the highest predictability of delivery relative to the priority class of the incoming message. For this purpose, the transmission rate of the receiver relative to the priority class of the message is calculated.

If this transmission rate is greater than 1, then this receiver is cooperative with respect to the priority class of the incoming message. Then, the availability of the NCS is checked. If this space is available, then the message of priority P_i is inserted. On the contrary, if this space is unavailable (saturated), then the receiver drops the NCS messages according to the increasing weight. Then the message of given priority is inserted.

On the contrary, if this transmission rate is less than 1, then the degree of selfishness of the node with respect to this priority class is calculated.

If the calculation of the degree of selfishness shows that the receiver is entirely selfish with respect to the priority class of the incoming message, then the receiver cannot receive the incoming message. Therefore, another receiver with a high probability of delivery is determined.

If the calculation of the degree of selfishness shows that the receiver is partially selfish in relation to the priority class of the incoming message, then the receiver is monitored for a certain period of time.

If during the monitoring period the node becomes cooperative, then the availability of the NCS is checked. If this space is available, then the priority message Pi is inserted. On the other hand, if this space is unavailable (saturated), then the receiver drops the NCS messages according to the increasing weight. Then the message of given priority is inserted. If after the monitoring time the node becomes selfish, then it cannot receive any message of this priority class.

The figure below illustrates the flow chart of proposed selfish node defense strategy.

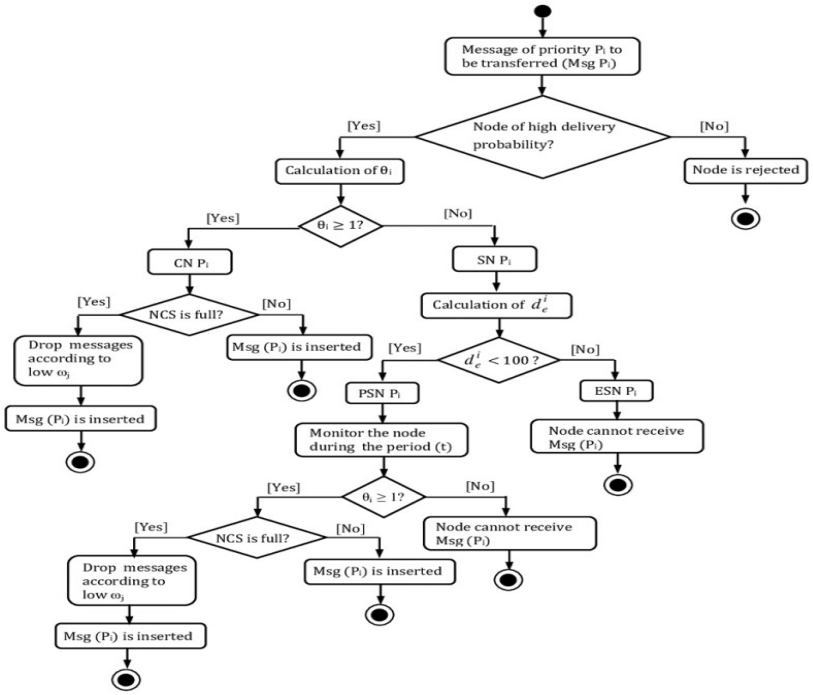


Figure 6. :-Flow chart of the proposed defense strategy.

Performance evaluation:-

In this section, we evaluate our strategy by comparing its performance to a model without detection of selfish nodes. This model combines the performance of the PRoPHET routing protocol [14] and the buffer management system based on message weight. This weight-based buffer management system, contrary to the proposed strategy, allows only one buffer partition. However, in case of buffer congestion, if the current node is the destination of the message, then the messages in the buffer are deleted with increasing TTL. On the other hand, if the current node is an intermediate node, then the messages in the buffer are deleted as their weight increases.

The simulation tool used to perform the evaluations is the ONE Simulator [30]. It is a discrete event simulator that is written with the JAVA language. This simulator can be run under Windows and Linux.

This section consists of two parts. The first part is dedicated to simulation settings and performance metrics. The second part presents performance analyses of the high priority message delivery rate, high priority message delivery time and network overload respectively.

Simulation settings:-

In all scenarios, we consider the map of a 4000m×4000m section of downtown Bouaké in Côte d'Ivoire (Figure 7). This simulation area admits neither fixed relay nodes nor trusted authority. We simulate scenarios for 24 hours with a number of vehicles varying from 20, 40, 60, 80 to 100 vehicles with constant buffer capacities fixed at 50MB. The vehicles move with a Shortest-Path Map-Based Movement model. Once their destination is reached, the vehicles pause for 600 to 900 seconds before choosing a new destination with speeds between 30 and 50 km/h.

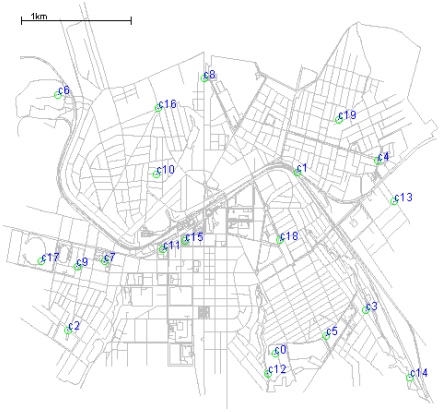


Figure 7. :-Initial positions of 20 mobile nodes on the map of the city center of Bouaké in Côte d'Ivoire.

In each scenario, in order to study the impact of selfish nodes on the performance of the proposed model, we vary the percentage of selfish nodes by 10%, 30%, and 50%.

In addition, for each of the selfish node percentages, in order to study the impact of selfish nodes on network performance with respect to high priority messages, we vary the degree of selfishness with respect to high priority messages by 10%, 30%, and 50%. In these scenarios, we set the degree of selfishness for low and medium priority messages at 10%.

The nodes use wireless communication technology (WiFi) with a transmission rate of 6 Mbps and a transmission range of 30m. In addition, the traffic volumes generated by high priority messages are large. For example, a high-priority message has a priority value of P=3 and a size in the range [750KB, 1.5MB]. A medium priority message has a priority value of P=2 and a size in the range [250KB, 750KB]. A low priority message has a priority value of P=1 and a size within the range [100MB, 250MB] [31]. In addition, all messages of each priority class are generated by three event generators, with a creation time in the range [15, 30] seconds and a TTL of 120 minutes. In the proposed model, each node is obliged to share 50% of its buffer memory. The monitoring time for partially selfish nodes is 60 minutes. All the parameters of the settings are given in table I below.

Table 1:- Simulation parameters and their values.

Parameter	***Values***
Simulation time	24 hours (86400 s)
Simulation area	4000m × 4000m
Number of nodes	20-40-60-80-100
Buffer size	50 MB
Transmission rate	6 Mbps
Transmission range	30 m
Random speed	30-50 km/h
Messages TTL	120 minutes
Waiting time	600-900 seconds
Creation interval	15-30 seconds
Message size	100KB-1.5MB
Percentage of selfish nodes	10%- 30%- 50%
Degree of selfishness in relation to low and medium priority message message	10%
Degree of selfishness in relation to the high-priority message	10%- 30%- 50%
Mobility model	Shortest-path map-based movement

In addition, we used three performance measures to evaluate the performance of our model: the delivery rate of high-priority messages, the average delivery time of high-priority messages and the network overload rate.

Performance analyses:-

This part focuses on the analysis of the performance of the proposed model. To this end, we analyze the impact of node selfishness on the delivery rate of high-priority messages, the delivery time of high-priority messages and network overload.

Impact of the number of selfish nodes on the delivery rate of high-priority messages:

Figures 8 (a and b) show the delivery rate of high priority messages when the network contains 10% selfish nodes. The degree of selfishness of the nodes in relation to the high priority message varies from 10%, 30% and 50%.

Based on the percentage of selfish nodes of 10% and the degree of selfishness of the node with respect to high priority messages of 10%, 30% and 50% in Figures 8(a) and 8(b), we determine the average delivery rate when the number of nodes varies between 20, 40, 60, 80 and 100. We find that the proposed model is able to improve the delivery rate of high priority messages. This improvement of the proposed model varies according to the degree of selfishness (10%, 30% and 50%).

When the degree of selfishness is 10%, the average delivery rate of high priority messages with the proposed model is 86.52%, while the rate without detection is 78.98%. This trend is verified with the degree of selfishness of 30% where the proposed model offers 86.60% average message delivery rate for HP messages compared to 78.82% for the model without detection. Finally, for the degree of selfishness of 50%, the average proportion of deliveries of high priority messages is, as before, higher with the proposed model (86.60%) compared to the model without detection (78.96%). Therefore, when the percentage of selfish nodes is 10% and the degree of selfishness of the node relative to high priority messages is 10%, 30% and 50%, our model increases the delivery rate relative to the model without detection by 7.55%, 7.78% and 7.63% respectively.

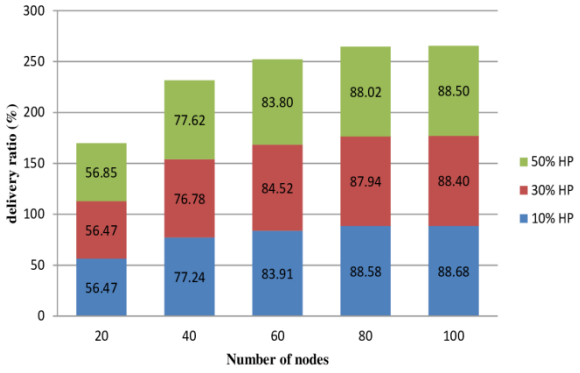


Figure 8:-

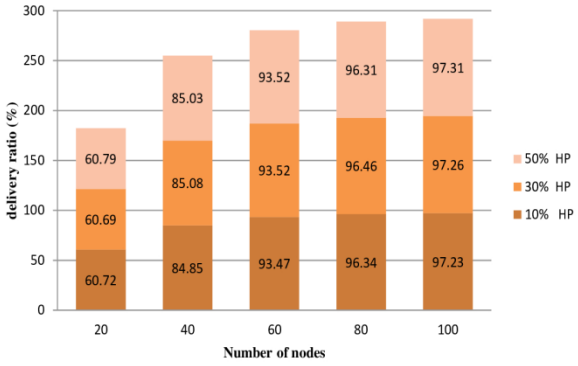


Figure 9:-

The delivery rate of high priority messages as a function of the number of nodes for 10% of selfish nodes for (a) no detection model (b) proposed model.

Figures 9 (a and b) show the delivery rate of high priority messages when the network contains 30% selfish nodes. The degree of selfishness of the nodes in relation to the high priority message varies from 10%, 30% and 50%. Similarly, based on the percentage of selfish nodes of 30% and the degree of selfishness of the node with respect to high priority messages of 10%, 30% and 50% in Figures 9(a) and 9(b), we determine the average delivery rate when the number of nodes varies between 20, 40, 60, 80 and 100. For our model, the average delivery rate is 86.59%, 86.55% and 86.58%. For the model without detection of selfish knots, the average delivery rate is 79.03%, 79.04% and 79.05%. Our model supports an increase in the delivery rate of high priority messages compared to the model without selfish node detection of 7.56%, 7.51% and 7.53%.

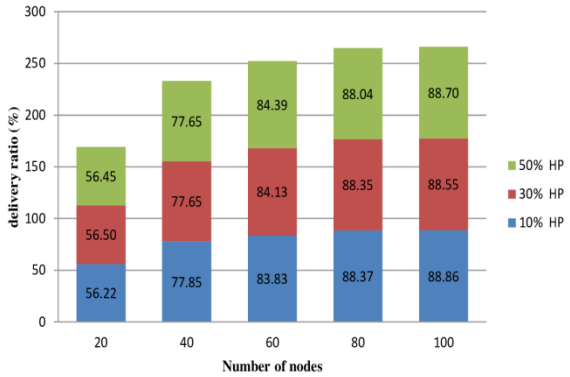


Figure 10 :-

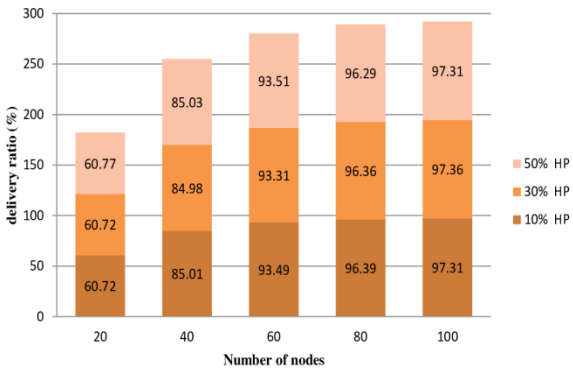


Figure 11:-

The delivery rate of high priority messages as a function of the number of nodes for 30% of selfish nodes for (a) no detection model (b) proposed model.

Figures 10 (a and b) show the delivery rate of high priority messages when the network contains 50% selfish nodes. The degree of selfishness of the nodes in relation to the high priority message varies from 10%, 30% and 50%. Similarly, based on the percentage of selfish nodes of 50% and the degree of node selfishness relative to the high priority messages of 10%, 30%, and 50% in Figures 10(a) and 10(b), we determine the average delivery rate when the number of nodes varies from 20, 40, 60, 80, and 100.

The average delivery rate for our model is 86.54%, 86.61% and 86.57%, compared to 79.22%, 79.14% and 79.08% for the model without detection, when the degree of selfishness relative to the high-priority message varies from 10%, 30% and 50%. Therefore, our model favors an increase in the delivery rate of high priority messages compared to the model without detection of 7.32%, 7.47% and 7.49%.

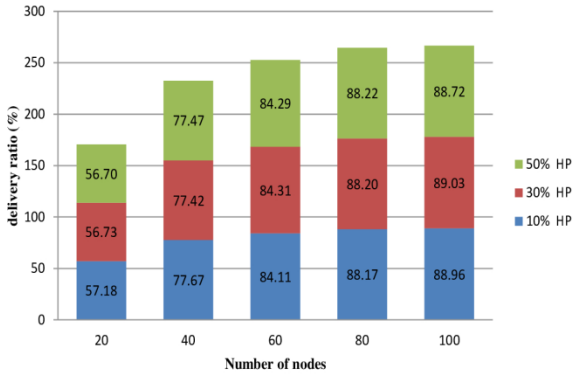


Figure 12 :-

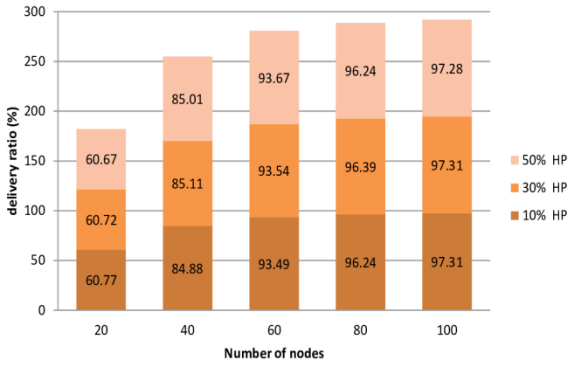


Figure 13:-

The delivery rate of high priority messages as a function of the number of nodes for 50% of selfish nodes for (a) no detection model (b) proposed model.

The observed results of high priority message delivery rates show that the proposed model allows for an improvement in the rate of high priority messages. The reason for these observed results is due to the fact that, in the proposed model, on the one hand nodes are obliged to share their buffer memory, and on the other hand nodes that are entirely selfish and partially selfish with respect to high priority messages are detected. As a result, either another cooperative node relative to the priority class of the high-priority message is selected in case the node is entirely selfish relative to high-priority messages, or the number of cooperative nodes relative to high-priority messages is increased by the monitoring mechanism.

Impact of the number of selfish nodes on the delivery delay of high-priority messages:
Figures 11 (a and b) give the average delivery delay for high priority messages when the network contains 10% selfish nodes. The degree of selfishness of the nodes in relation to the high priority message varies from 10%, 30% and 50%.

Regarding the delivery delay of high-priority messages, when the percentage of selfish nodes is 10%, the proposed model is more efficient than the model without detection for at least 40 nodes. Below 20 nodes, the proposed model is less efficient than the no-detection model because it offers a higher average delivery delay than the no-detection model. But globally, the proposed model must be prioritized because even if initially (for the 20 node size), it has a slightly longer delivery delay; thereafter, the delivery delay only decreases for an increasing number of nodes. However, the relevance of the proposed model differs according to the degree of selfishness of the node with respect to the high-priority messages of 10%, 30% and 50%.

With a 10% degree of selfishness, the proposed model offers a shorter average delay (48.09 min) compared to the model without detection where we obtain an average delay of 55.41 min. This tendency is confirmed with the 30% degree of selfishness where the proposed model is even better than the model without detection because (TimeModel = 48.12 min < Timewithout-detection = 55.34 min).

Considering the past results, it is not surprising to note that with the 50% degree of selfishness, the proposed model is still better than the model without detection. Here, too, the average delivery delay is shorter with the proposed model (48.11 min) compared to 55.46 min with the model without detection.

The overall average delivery delay for high priority messages calculated in the case of the proposed model (proven to be better than the model without detection), gives results between 48.09 min and 48.11 min. These are reasonable times which suggest that the strategy of the proposed model should be implemented to improve the quality of the network.

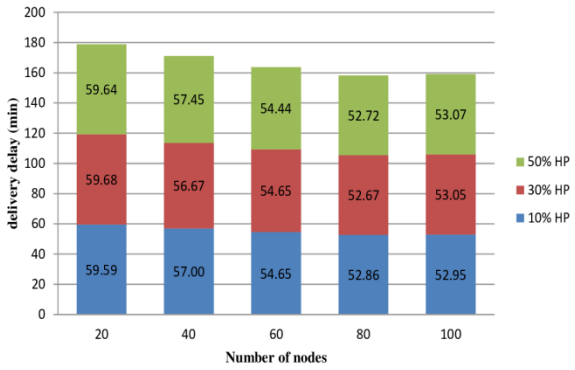


Figure 14:-

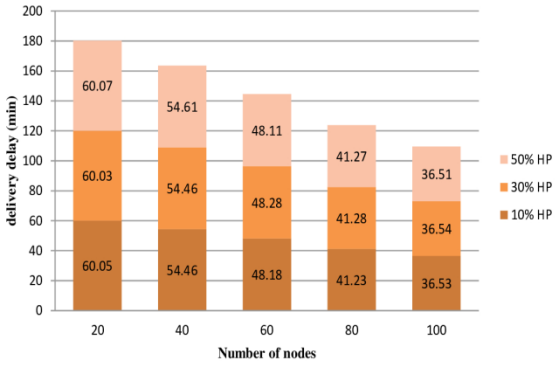


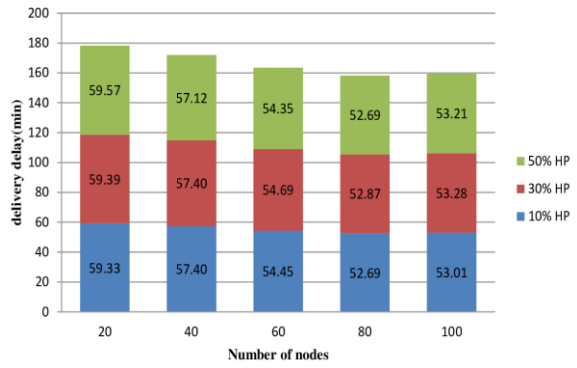
Figure 15:-

The delivery delay of high priority messages as a function of the number of nodes for 10% of selfish nodes for (a) no detection model (b) proposed model.

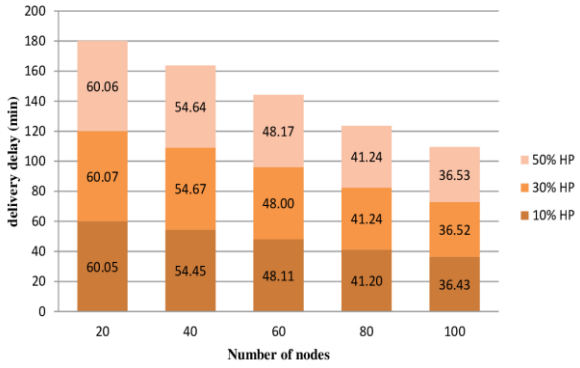
Figures 12 (a and b) show the average delivery delay for high priority messages when the network contains 30% selfish nodes. The degree of selfishness of the nodes in relation to the high priority message varies from 10%, 30% and 50%.

Similarly, based on the percentage of selfish nodes of 30% and the degree of node selfishness relative to high priority messages of 10%, 30% and 50% in Figures 12(a) and 12(b), for 20 nodes, HP's average message delivery delay for the proposed model (60.05 min) is slightly higher than the model without detection (59.43 min). Overall, the proposed model should be prioritized because although it initially (for the 20 node size) has a slightly longer delay, the delivery delay only decreases as the number of nodes increases. However, the relevance of the proposed model differs according to the degree of selfishness of the node with respect to the high priority messages of 10%, 30% and 50%.

When the degree of selfishness is 10%, the proposed model offers a shorter average delay (48.05 min) compared to the model without detection where the average delay is 55.37 min. The same is true for a 30% degree of selfishness. That is to say 48.1 min for the proposed model, compared to 55.52 min for the model without detection. Moreover, for 50% of the degree of selfishness, the proposed model has an average delivery delay of 48.12, against 55.4 min for the model without detection.



(a)



(b)

Figure 16:-

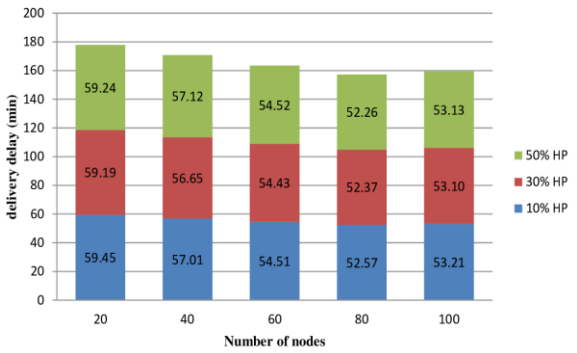
Figure 17:-

The delivery delay of high priority messages as a function of the number of nodes for 30% of selfish nodes for (a) no detection model (b) proposed model.

Figures 13 (a and b) give the average delivery delay for high priority messages when the network contains 50% selfish nodes. The degree of selfishness of the nodes in relation to the high priority message varies from 10%, 30% and 50%.

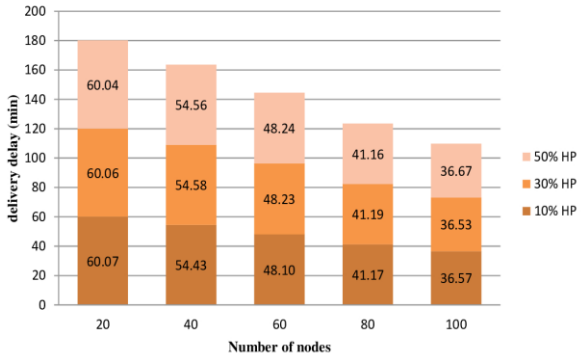
Similarly, based on the percentage of selfish nodes of 30% and the degree of node selfishness relative to high priority messages of 10%, 30%, and 50% in Figures 11(a) and 11(b), for 20 nodes, HP's average message delivery delay for the proposed model (60.06 min) is slightly higher than the model without detection (59.429 min). Overall, the proposed model has a slightly longer delivery delay when the network consists of 20 nodes; thereafter, the delivery delay only decreases as the number of nodes increases. However, the appropriateness of the proposed model differs according to the degree of selfishness of the node with respect to high priority messages of 10%, 30% and 50%.

With a 10% degree of selfishness, the proposed model offers a shorter average delay (48.07 min) compared to the model without detection where the average delay is 55.35 min. The same is true for a 30% degree of selfishness, where the proposed model offers 48.11 min, compared to 55.14 min for the model without detection. Furthermore, for 50% of the degree of selfishness, the proposed model offers an average delivery delay of 48.13 min, compared to 55.25 min for the model without detection.



(a)

Figure 18:-



(b)

Figure 19:-

The delivery delay of high priority messages as a function of the number of nodes for 50% of selfish nodes for (a) no detection model (b) proposed model.

The observed results of the average delivery delay for high priority messages show that the proposed model offers a shorter average delay than the model without detection of selfish nodes. The reason for these observed results for average delivery delay of high priority messages is due to the fact that in the proposed model, nodes that are entirely selfish and partially selfish with respect to high priority messages are quickly detected. As a result, either another cooperative node relative to the priority class of the high-priority message is selected to transfer high priority in case the node is entirely selfish, or the monitoring mechanism is used in case the node is partially selfish relative to the high-priority messages.

Impact of the number of selfish nodes on network overload:

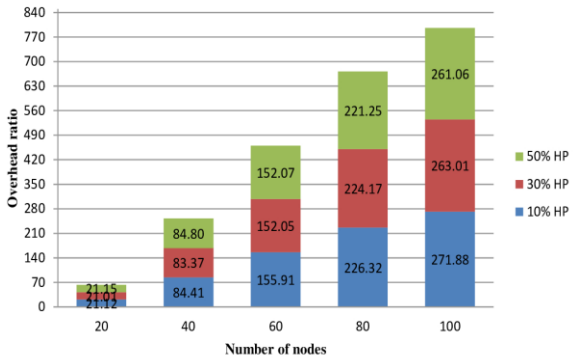
Figures 14 (a and b) show the network overload when the network contains 10% selfish nodes. The degree of selfishness of the nodes in relation to the high-priority message varies from 10%, 30% and 50%.

With regard to network overload, the proposed model greatly reduces this phenomenon compared to the model without detection. However, taking into account the different degrees of node selfishness in relation to the high priority messages of 10%, 30% and 50%, the variants of this performance can be seen.

When the degree of selfishness is 10%, the proposed model reduces the network overload rate by an average of about 36 compared to 152 in the model without detection of selfish nodes. This is a good result that complements the already detected performance bottlenecks (i.e. HP's message delivery rate and time) of our proposed model.

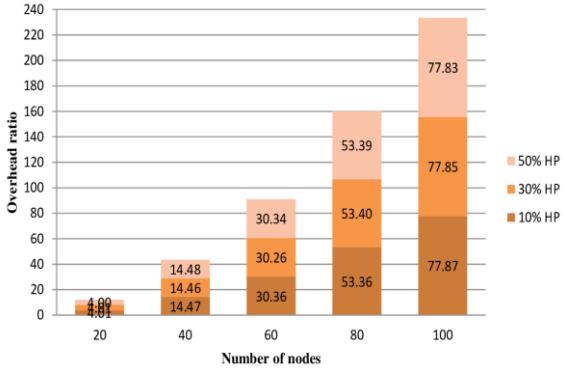
Moreover, this trend holds true for the 30% degree of selfishness. Indeed, based on the 30% degree of selfishness, the proposed model limits network overload to an average of 36 HP messages, while the model without detection of selfish nodes only manages to limit the network overload rate to an average of 148.72. Finally, for the 50% selfishness level, the proposed model still outperforms the model without detection of selfish nodes with a reduction of network overload leading to an average of only 36, compared to an average of 148.07 for the model without detection of selfish nodes.

Whatever the degree of selfishness, the results give an average network overload rate of 36 in the case of our proposed model. This is also a result that militates in favor of implementing this proposed model in order to once again improve the quality of the network for users



(a)

Figure 20:-



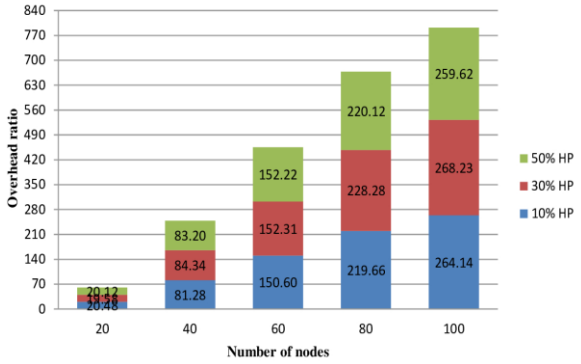
(b)

Figure 21:-

Network overload ratio as a function of the number of nodes for 10% of selfish nodes for (a) no detection model (b) proposed model.

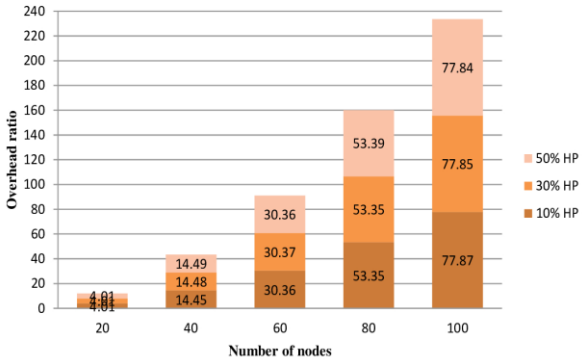
Figures 15 (a and b) give the network overload when the network contains 30% selfish nodes. The degree of selfishness of the nodes in relation to the high-priority message varies from 10%, 30% and 50%.

When the number of selfish nodes is 30%, the proposed model reduces the network overload more than the model without detection of selfish nodes. This performance is evidenced by the different degrees of node selfishness relative to high priority messages of 10%, 30% and 50%. When the degree of selfishness is 10%, the proposed model reduces the network overload on average by 36.01 compared to 147.23 for the model without selfish node detection. On the other hand, based on 30% selfish nodes, the proposed model reduces the network overload on average by 36.01 compared to 150.55 for the model without selfish node detection. Indeed, based on 50% degree of selfishness, the proposed model reduces the network overload on average by 36.02 compared to 147.06 for the model without detection of selfish nodes.



(a)

Figure 22:-



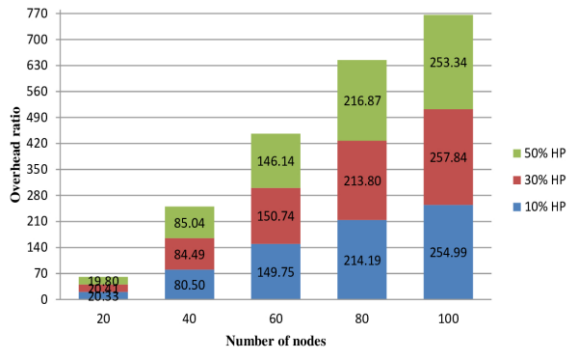
(b)

Figure 23:-

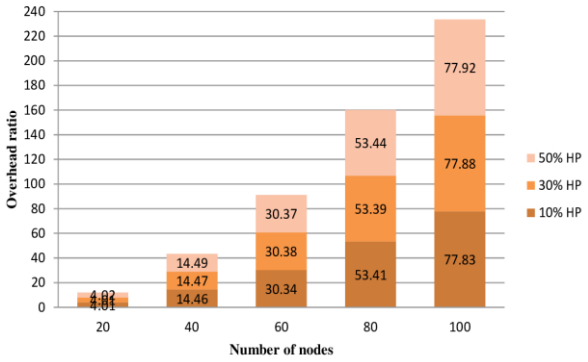
Network overload ratio as a function of the number of nodes for 30% of selfish nodes for (a) no detection model (b) proposed model.

Figures 16 (a and b) show the network overload when the network contains 50% selfish nodes. The degree of selfishness of the nodes in relation to the high-priority message varies from 10%, 30% and 50%.

Analogous to the previous case, we find that regardless of the degree of selfishness, the proposed model significantly reduces the network overload compared to the model without detection of selfish nodes. When the number of selfish nodes is 50%, the proposed model reduces the network overload by 36.01, 36.03 and 36.05 when the degree of selfishness of the high priority message varies by 10%, 30% and 50% respectively compared to 143.95, 145.46 and 144.24 for the model without selfish node detection.



(a)



(b)

Figure 24:- **Figure 25:-**

Network overload ratio as a function of the number of nodes for 30% of selfish nodes for (a) no detection model (b) proposed model.

The observed results of network overload rates show that the proposed model reduces network overload more than the model without detection of selfish nodes. The reason for these observed results for network overload rates is due to the fact that in the proposed model, selfish nodes (fully and partially) are detected, either by calculations of the node's transmission rate relative to the priority class, or by calculations of the degree of selfishness of the node. As a result, the transfer of messages to selfish nodes is preferentially avoided. As a result, the number of copies is reduced.

Conclusion and Perspectives:-

The presence of selfish nodes in the network affects the performance of routing protocols in terms of delivery rate, delivery delay and network overload. In order to reduce their impact in networks where services are differentiated by messages with different priorities, we propose a new strategy for detecting selfish nodes taking into account the priority class of messages. This strategy is based on the calculation of the transmission rate of the node with the highest predictability of delivery, the calculation of the degree of selfishness of the node with respect to the priority class, the monitoring mechanism and a system for managing a partitioned memory taking into account the time-to-live and the priority class of messages. The results of the simulations performed with the ONE simulator show that the proposed strategy reduces the delivery delay of high priority messages, increases the delivery rate of high priority messages and reduces network overload.

In this paper, we used the PRoPHET routing protocol under the assumption that nodes could not change their delivery probabilities. However, since the PRoPHET routing protocol uses the encounter history to determine the delivery probability of nodes, some nodes can modify this property to be chosen as the next valid delay node while launching attacks. This will then lead to a degradation of the overall network performance. Therefore, our next work is part of the fight against malicious and selfish node attacks.

Refernces:-

1. Cao, Yue and Sun, Zhili. Routing in delay/disruption tolerant networks: A taxonomy, survey and challenges. *IEEE Communications surveys & tutorials*, 2013, vol. 15, no 2, p. 654-677.
2. Isento, J. N., Rodrigues, J. J., Dias, J. A., Paula, M. C., & Vinel, A. Vehicular delay-tolerant networks - A novel solution for vehicular communications. *IEEE Intelligent Transportation Systems Magazine*, 2013, vol. 5, no 4, p. 10-19.
3. Soares Vasco NGJ, Farid Farahmand, and Joel JPC Rodrigues. "Traffic differentiation support in vehicular delay-tolerant networks." *Telecommunication Systems* 48.1-2 (2011) : 151-162.
4. V. N. G. J. Soares, F. Farahmand, and J. J. P. C. Rodrigues, "A Layered Architecture for Vehicular Delay-Tolerant Networks," in *IEEE Symp. onComp. and Comm. (ISCC 2009)*, Sousse, Tunisia, 2009, pp. 122 - 127.
5. Dias, João AFF, Joel JPC Rodrigues, and Liang Zhou. "Cooperation advances on vehicular communications: A survey." *Vehicular communications* 1.1 (2014): 22-32.
6. Dias J. A. F. F., J. J. P. C. Rodrigues et L. Shu. Performance evaluation of cooperative strategies for Vehicular Delay-Tolerant Networks. *Transactions on Emerging Telecommunications Technologies*, 2014, vol. 25, no 8, p. 815-822.
7. Dias J. A. F. F., J. J. P. C. Rodrigues, N. Kumar, and K. Saleem, "Cooperation Strategies for Vehicular Delay-Tolerant Networks," no. December, pp. 88–94, 2015
8. Dias J. A. F. F, Rodrigues Joel JPC, Isento João NG, *et al.* "The impact of cooperative nodes on the performance of vehicular delay-tolerant networks". *Mobile Networks and Applications*, 2013, vol. 18, no 6, p. 867-878.
9. Magala Naércio, Pereira Paulo Rogério, et Correia Miguel P. Nodes' misbehavior in Vehicular Delay-Tolerant Networks. In : *Future Internet Communications (CFIC), 2013 Conference on*. IEEE, 2013. p. 1-9.
10. Benamar Maria, Nabil Benamar, and Driss El Ouadghiri. "The effect of cooperation of nodes on VDTN routing protocols." *Wireless Networks and Mobile Communications (WINCOM), 2015 International Conference on*. IEEE, 2015 p. 1-7.
11. Khalid Waqar, Ullah Zahid, Ahmed Naveed, *et al.* A taxonomy on misbehaving nodes in delay tolerant networks. *computers & security*, 2018, vol. 77, p. 442-471.
12. Abdelhamid Abouhassane et Benamar Nabil. "A new analysis of strategies against non-cooperative nodes in DTN environnement". Conference Paper May 2016 .DOI: 10.13140/RG.2.2.26280.78085
13. De Jesus, Vasco Nuno da Gama, Rodrigues Joel José Puga Coelho, Ferreira Paulo Salvador, *et al.* Improvement of messages delivery time on vehicular delay-tolerant networks. In : *Parallel Processing Workshops, 2009. ICPPW'09. International Conference on*. IEEE, 2009. p. 344-349.
14. A. Lindgren, A. Doria, and O. Schelen, "Probabilistic routing in intermittently connected networks. SIGMOBILE Mob," Comput. Commun. Rev. vol. 7, no. 3, 2003
15. Xu Xiaoqiong, et al. "A routing algorithm based on node selfishness and buffer management in delay tolerant networks." *2015 International Conference on Wireless Communications & Signal Processing (WCSP)*. IEEE, 2015.
16. Sharma, A., Singh, D., Sharma, P., & Dhawan, S. (2015, February). Selfish nodes detection in delay tolerant networks. In *2015 International Conference on Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE)* (pp. 407-410). IEEE.
17. Girdhar, V., & Banga, G. (2015). A incentive based scheme to detect selfish nodes in MANET. *International Journal of Advanced Research in Computer Science and Software Engineering*, *5*(8), 561-565.
18. Sunil Kumar "Detecting and Avoiding of Selfish Nodes in Delay Tolerant Networks (DTNs) " International Journal of Recent Research Aspects ISSN: 2349-7688, Vol. 5, Issue 1, March 2018, pp. 325-329
19. A. K. Gupta, I. Bhattacharya, P. S. Banerjee, and J. K. Mandal, "A Cooperative Approach to Thwart Selfish and Black-Hole Attacks in DTN for Post Disaster Scenario," 2014 Fourth International Conference of Emerging Applications of Information Technology, 2014.
20. Liang Xuan, Qin Junxiang, Wang Meimei, *et al.* An effective and secure epidemic routing for disruption-tolerant networks. In : *Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2014 Sixth International Conference on*. IEEE, 2014. p. 329-333.
21. Bhoiwala Jaina P. et Jhaveri, Rutvij H. Cooperation based defense mechanism against selfish nodes in DTNs. In : *Proceedings of the 10th International Conference on Security of Information and Networks*. ACM, 2017. p. 268-273.
22. Dias João A.F.F, Rodrigues J. J. P. C, Kumar Neeraj, *et al.*A hybrid system to stimulate selfish nodes to cooperate in vehicular delay-tolerant networks. In : *Communications (ICC), 2015 IEEE International Conference on*. IEEE, 2015b. p. 5910-5915.

23. Dias João A.F.F, Rodrigues J. J. P. C, F. Xia, et C. X. Mavromoustakis, "IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS A Cooperative Watchdog System to Detect Misbehavior Nodes in Vehicular Delay-Tolerant Networks," 2015.
24. Spaho Evjola, et al. "Evaluation of Single-Copy and Multiple-Copy Routing Protocols in a Realistic VDTN Scenario." *2016 10th International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS)*. IEEE, 2016.
25. Bhoi, Sourav Kumar, and Pabitra Mohan Khilar. "Vehicular communication: a survey." *IET networks* 3.3 (2013): 204-217.
26. T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait : an efficient routing scheme for intermittently connected mobile networks," in *Proc. of the 2005 ACM SIGCOMM workshop on Delay tolerant networking*, ser. WDTN '05. New York, NY, USA: ACM, pp. 252–259, 2005.
27. V.N.G.J. Soares, J.J.P.C.Rodrigues, Cooperation in DTN-based network architectures, in : S. Misra, M .Obaidat (Eds.), Cooperative Networking , Wiley, ISBN 978-0-470-74915-9, 2011, pp. 101–115 (Chapter7), http://dx.doi/org/10.1002/9781119973584.ch7.
28. Ruchira more and Milind Penurkar . "Scheduling and Dropping Policies in Delay Tolerant Network". In International Advanced Research Journal in Science, Engineering and Technology (IARJSET), 2015 Vol. 2, Issue 10, DOI 10.17148/IARJSET.2015.21014
29. Bhoi, Sourav Kumar, and Pabitra Mohan Khilar. "Vehicular communication: a survey." *IET networks* 3.3 (2013): 204-217.
30. Keränen A, Ott J, Kärkkäinen T. (2009) The ONE simulator for DTN protocol evaluation[C]//proceedings of the 2nd international conference on simulation tools and techniques. ICST (Institute for Computer Sciences, social-informatics and telecommunications engineering), 55-64
31. Sadreddini, Z., & Afshord, M. M. (2013). Impact of using several criteria for buffer management in vehicular delay tolerant networks. *World Applied Sciences Journal*, *22*(9), 1204-1209.