

Article

Efficient Data Transfer by Evaluating Closeness Centrality for Dynamic Social Complex Network-Inspired Routing

Manuel A. López-Rourich ^{1,*}  and Francisco J. Rodríguez-Pérez ²¹ Department of Crowdsourcing Mobile Data Intelligence, Trecone Solutions, 10003 Cáceres, Spain² Department of Computing and Telematics Systems Engineering, University of Extremadura, 10003 Cáceres, Spain; fjrodri@unex.es

* Correspondence: alopez@trecone.com; Tel.: +34-927251657

Abstract: Social Complex Networks in communication networks are pivotal for comprehending the impact of human-like interactions on information flow and communication efficiency. These networks replicate social behavior patterns in the digital realm by modeling device interactions, considering friendship, influence, and information-sharing frequency. A key challenge in communication networks is their dynamic topologies, driven by dynamic user behaviors, fluctuating traffic patterns, and scalability needs. Analyzing these changes is essential for optimizing routing and enhancing the user experience. This paper introduces a network model tailored for Opportunistic Networks, characterized by intermittent device connections and disconnections, resulting in sporadic connectivity. The model analyzes node behavior, extracts vital properties, and ranks nodes by influence. Furthermore, it explores the evolution of node connections over time, gaining insights into changing roles and their impact on data exchange. Real-world datasets validate the model's effectiveness. Applying it enables the development of refined routing protocols based on dynamic influence rankings. This approach fosters more efficient, adaptive communication systems that dynamically respond to evolving network conditions and user behaviors.



Citation: López-Rourich, M.A.; Rodríguez-Pérez, F.J. Efficient Data Transfer by Evaluating Closeness Centrality for Dynamic Social Complex Network-Inspired Routing. *Appl. Sci.* **2023**, *13*, 10766. <https://doi.org/10.3390/app131910766>

Academic Editor: Christos Bouras

Received: 12 August 2023

Revised: 21 September 2023

Accepted: 25 September 2023

Published: 27 September 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The vast increase in autonomous and heterogeneous wireless devices poses challenges for the future of communication systems due to the complexity of their interconnection, which involves multiple networking technologies and a wide range of device capabilities. According to statistics provided by Statista [1], the global number of smartphones reached nearly 6.6 billion in 2022 and is projected to surpass 7.8 billion by 2028. In other words, a world of pervasive mobile devices is being built that has vast processing capabilities and allows for smooth communication among them, enabling greater connectedness. Each node or gadget in a mobile communication network is a connecting point that is innately connected to a person who is moving and takes part in the network's data exchange. Moreover, Mobile Social Networks (MSNs) leverage wireless devices and function as a communication infrastructure designed for point-to-point and short-range communications, seeking increased data exchange efficiency by adapting to the typical movements and behaviors of individuals using mobile devices [2–5]. In this regard, opportunistic networks emerge as a classification of wireless networks characterized by sporadic, unreliable, or constrained user-to-user ad hoc connections [6]. In such networks, conventional routing algorithms often depend on the “storage-carry-and-forward” approach, whereby a node forwards messages to a varying number of neighboring nodes it encounters based on the specific routing algorithm employed. Nevertheless, this flooding strategy can result in a proliferation of message duplicates, potentially leading to network and device congestion [7].

Within such a context, a Dynamic Social Complex Network (DSCN) can be defined as a network that leverages human social behavior, such as daily routines, mobility patterns, and interests, to facilitate message routing and data sharing over time. In these networks, nodes (users with mobile devices) can form on-the-fly social networks to communicate with each other. Considering users' social routines when determining whether a node should retransmit a message to another node can reduce transmission delay and routing overhead [8]. Consequently, minimizing routing overhead will decrease the average number of hops that message routes traverse before reaching their destination.

The search for the most influential or important nodes is a critical component of analyzing and comprehending the network topology dynamics due to its intrinsic role in determining the network's overall structure or efficiency [9–11]. Influential devices often act as key hubs for data exchange and efficient communication pathways and, depending on the objective, the significance of a node can vary. There are several metrics of node importance, such as degree centrality, closeness centrality, or social centrality. The first refers to the number of its direct connections to other nodes and analyzes its level of activity in the network's topology compared to others. Closeness centrality measures a node's proximity to other nodes and indicates how efficiently it can access or distribute information within the network. Finally, social centrality aims to capture the extent to which certain nodes in the network hold influential positions in terms of data forwarding or control over the network's social dynamics. Social centrality metrics take into account factors such as connectivity, interactions, and relationships among nodes to assess their relative significance in the social behavior of the network.

However, conventional centrality metrics used in traditional networks are not useful in DSCNs, as they rely on a static network model where there are multiple connections and disconnections over time that are usually aggregated into a single binary network. As a result, traditional metrics have been extended to work with weighted or dynamic topologies [12–15]. The authors of [16] considered the number of connections as link weights and redefined the centrality metrics to consider both the number of links and their weights in the graphs. On the other hand, the authors of [17] proposed time-based measures that leverage the temporal patterns of changing topologies. Furthermore, individuals have inherent social tendencies, and their behavioral patterns, which are substantially influenced by the patterns of interaction among individuals, are not random [18,19]. Thus, when identifying hub nodes in the network, mobility patterns, spatiotemporal connections, and social behavior must also be considered.

In examining the current landscape of detecting influential nodes, several notable works have delved into incorporating both non-social and social attributes associated with network nodes. The research conducted by the authors of [20] focuses on centrality metrics, which help identify important nodes in a network, crucial for understanding network structures and behaviors. Static and dynamic centrality metrics are discussed, including their relevance in weighted networks. The study highlights challenges, proposes new centrality metrics, and emphasizes the importance of considering temporal aspects in network analysis. It also explores network resilience and the impact of centrality on fault tolerance. However, the authors do not explore a broader range of centrality measures or compare this measure with other existing centrality measures comprehensively. The research conducted by the authors of [21] delves into online information propagation within complex networks, emphasizing the critical role of influential nodes in network structure and operation. The paper classifies centrality measures into global, local, and semi-local types, exploring their effectiveness in identifying influential nodes. It introduces a novel centrality measure, 'centripetal centrality,' and presents an algorithm, 'seeds exclusion,' to enhance information propagation. The work demonstrates the effectiveness of 'centripetal centrality' in identifying key nodes and improving propagation effects. The authors assume that identifying influential spreaders is essential for maximizing information coverage. However, this assumption might not always hold true, especially in scenarios where the objective differs from maximizing information spread. For instance, in communication networks, it is crucial

to minimize overhead rather than increase it. The authors of [22] address the prediction of social network dynamics and evolution, distinguishing between short-term dynamics and long-term changes. The proposed methodology, MONDE, utilizes hidden Markov models and a genetic algorithm to predict individual, group, and network dynamics. The approach aims to provide a comprehensive view of network evolution and dynamics, benefiting fields such as marketing and public security by aiding decision-making and strategy planning. However, the accuracy and effectiveness of MONDE heavily depend on the quality and availability of data, especially the posting activities and comments used for feature extraction. Incomplete or inaccurate data could lead to less reliable predictions, particularly in the case of low-density networks where empty or discontinuous samples may exist in the data. The study carried out by the author of [23] explores critical node detection and introduces a novel centrality measure, known as isolating centrality, to identify nodes that significantly impact network connectedness. The paper emphasizes the importance of accurately identifying critical nodes for ensuring network reliability and provides a comparative analysis of centrality measures' performance. It also investigates the correlation between leverage centrality and critical nodes, showcasing the effectiveness of the proposed centrality measure. However, it is worth noting that the effectiveness of this proposed measure is influenced by the structure of the nodes' neighborhood, especially in detecting critical nodes that segregate the network into connected components. This dependency might limit its effectiveness in certain low-density topologies. The authors of [24] focus on seed node selection in online social networks (OSNs) for information propagation and influence maximization. The study explores various centrality measures, such as clustering coefficients and node degree, to identify influential seed nodes. It considers Twitter as a platform for opinion generation and discusses the relevance of centrality measures as seed nodes in large-scale networks. The study also conducts a comparative analysis using benchmark similarity measures to assess the effectiveness of different centrality measures in seed node selection. The study acknowledges that the effectiveness of seed node selection is influenced by the network's structure. Certain propagation approaches, like Random Walk, are affected by local clustering. This sensitivity to network structure implies that the effectiveness of the proposed approach could vary significantly in network topologies with insufficient connections.

In summary, these works propose or utilize specific centrality measures to assess the importance of nodes in a network, hence their focus on identifying influential or critical nodes within the network as shown in Table 1. These nodes are deemed essential for information propagation and can play a crucial role in maximizing information coverage within the network. However, the objective is not always to maximize the pathways through which information circulates. In communication networks, it is preferable to maintain low overhead values to avoid unnecessary consumption of memory resources in intermediary devices forwarding data to their destinations. Furthermore, these works exhibit a certain dependence on network topology, implying that effectiveness could vary in cases of low connection density, as observed in opportunistic networks.

Based on the aforementioned concerns, this research paper aims to identify and rank hub nodes using a dynamic network model to analyze how device connections evolve over time. For that, the behavior of a DSCN is gathered by a progression of graphs as the devices connect and disconnect throughout the network operation. First, we introduce a novel local centrality metric, Dynamic Degree centrality, as we believe that both the number of neighbors and the frequency of connections with them serve as valuable cues of a node's importance in the network.

This metric seamlessly integrates both factors, effectively gauging the node's centrality based on the progression of its connections and contact frequency with neighboring nodes. Furthermore, we have developed a closeness centrality measure to address the potential impact of longer forwarding delays on storage capacity utilization (network overhead) and its subsequent influence on data forwarding likelihood. To quantify a device's global centrality, we propose the Dynamic Closeness centrality based on the temporal evolution network

model, which considers forwarding overhead. We also propose the Social-based Closeness Centrality Metric, which considers social relations to provide an effective centrality metric to ensure that data are carried and forwarded by relay devices with a high likelihood of reaching the destination host. This is because social relations and behaviors among wireless users are typically long-term characteristics and less fluctuating than device mobility. Thus, to assess the usefulness of our suggested centrality metrics and to examine the properties of the centrality distribution applied to various Quality of Service (QoS) measurements, we evaluate the results of experiments run on real-world datasets.

Table 1. Comparison of related works on the detection of the most influential nodes in a network.

Study	Objective	Methodology	Key Findings
[20]	Detection of influential nodes in dynamic weighted networks.	Time-ordered weighted graph models with Opshal's algorithms, considering temporal aspects.	New hybrid centrality measure: Temporal Closeness-Closeness measure.
[21]	Identification of influential spreaders.	Integrate degree, constraint coefficient, and k-shell for a comprehensive assessment of node importance.	Centripetal centrality as an effective measure to identify influential nodes.
[22]	Prediction of the dynamics and evolution of a social network.	Two-layer HMM to model individual and group dynamics.	MONDE, demonstrating prediction accuracy rates for dynamics and evolution in social networks.
[23]	Detection of critical nodes of networks.	Compare centrality measures' effectiveness.	Isolating centrality as an effective measure for identifying critical nodes.
[24]	Correlation between seed node detection and information flow.	Investigate different centrality measures for seed node detection.	Emphasize the impact of network structure on seed node selection.

The remainder of this paper is organized as follows: Section 2 provides an overview of the dynamic network model used. The proposed local and global influence metrics are described in detail in Section 3. In Section 4, we present the experimental results of all the proposed centrality metrics, including a comparative analysis of different QoS metrics. Finally, we offer conclusions in Section 5.

2. Model and Method

In this part, we first use the dynamic network model to show how the topological structures of DSCNs are constantly evolving. Using this model as a guide, we look at people's social connections and movement patterns to create new interpretations of traditional influence measurements that are based on the network's dynamic.

A graph G is made up of a limited number of nodes (V) and edges (E). Since there cannot be an empty set of nodes, $V \neq \emptyset$, and thus $V = \{v_1, v_2, \dots, v_n\}$. Pairs of nodes (v_i, v_j) that represent some sort of connection pattern between nodes make up the collection of edges E . The terms nearby and neighboring are used to describe two nodes connected by an edge. The network is referred to as undirected if the edges are unordered, where $(v_i, v_j) = (v_j, v_i)$.

If there is a relationship between the nodes v_i, v_j , then an adjacency matrix M with elements $m_{ij} = 1$ and 0 otherwise can be used to fully describe network G . Unweighted or binary networks are examples of this. In general, G is characterized using an adjacency matrix, where $m_{ij} \geq 1$ if there is an edge between nodes v_i, v_j , and 0 otherwise, where the edges contain a numerical value measuring a feature of the edge. A network G is also considered to be connected if, for every pair of distinct nodes v_i, v_j , there is a route from v_i to v_j ; otherwise, it is said to be unconnected [25]. However, complex networks are a special kind of graph in which the nodes and edges have complicated organizational structures and non-trivial topological properties. Since these networks contain complicated patterns and features, the interactions between the pieces in this situation are not clear-cut or simple.

Dynamic Model of a Complex Network

A Dynamic Complex Network (DCN) is made up of several nodes, which stand for individual devices, and edges, which represent the connections or temporal interactions between them. Traditional static complex network models cannot adequately capture such dynamic evolution since the topology and device placements coevolve over time. The time-ordered network was suggested by the authors of [17] to transform a dynamic network into a static network with directed flows. The authors of [26] analyzed the uniformity of device behavior over time. We build upon their work and propose a dynamic network model that captures the evolving nature of a DCN. Our objective is to predict the behavioral trends of devices in the network and measure their QoS parameters. The authors of [26] rely on Shannon entropy to verify the uniformity of device behavior, whereas we go further by conducting regression studies to analyze not only uniformity but also the trend of node behavior and directly apply it to QoS metrics in opportunistic networks with low connection density.

By considering the temporal sequence, length, and correlations between connections or devices happening at various moments in time, our model seeks to give a clear and thorough framework for understanding the evolutionary patterns of the network. Our model illustrates the evolution of interactions between devices in a DCN over a certain time period by using a series of snapshots. We shall outline the basic concepts of the dynamic network model in the parts below:

Define a finite set of devices V (nodes) and a set of connections E (edges) between these devices. The connections between devices are assumed to take place over a time span T . We use L to denote the duration of each spatial snapshot (or time window size), and $F_T = \frac{T}{L}$ represents the number of spatial snapshots during the time span T . The dynamics of the network can be subsequently described by $G_T = (V_T, E_T, M_T)$,

where

$V_T \subseteq V$ is the collection of all networked devices throughout the duration of T .

$E_T \subseteq E$ is the set of edges that stands for connections between devices throughout the course of time T .

M_T is a sequence of connectivity matrices that record contact events of devices during the time span T .

A discretized collection of static complex networks, $G_T = \{G_1, G_2, \dots, G_{F_T}\}$, can be used to simulate a DCN. In this model, the edges in each connection matrix are not binary as they are in the adjacency matrix of an unweighted graph. The connectivity matrix's edge weights, which range from 0 to F_T , indicate how frequently points of contact occur.

For the sake of clarity, we provide the following example in Figure 1, where the network's aggregated view is represented by G_T . In this scenario, two devices are said to be connected if they have made contact within a time interval t_i . The network snapshots are denoted as $\{G_1, G_2, \dots, G_{10}\}$. All information from both geographical and temporal data is included in this network's representation. Figure 2 displays the connection matrixes in order.

The connectivity matrix is symmetric since each snapshot is an undirected graph with a connection between devices denoting the presence of a contact link in both directions. For instance, the weight of M_{AC} in M_1 is 0, indicating that device A and device C did not make contact during time unit t_1 . One link between device A and device C was represented by the weight of M_{AC} in M_8 , which is 1, during the time unit t_8 .

The costs of the routes between a source and a destination in the domain G_T can be represented as a function of delay, $\{\delta_{ij}x_{ij}\}_{t_{ij}}$, if the objective is to minimize the message delivery delay, or as a function of load, $\{\lambda x_{ij}\}_{t_{ij}}$, if the objective is to minimize the overhead.

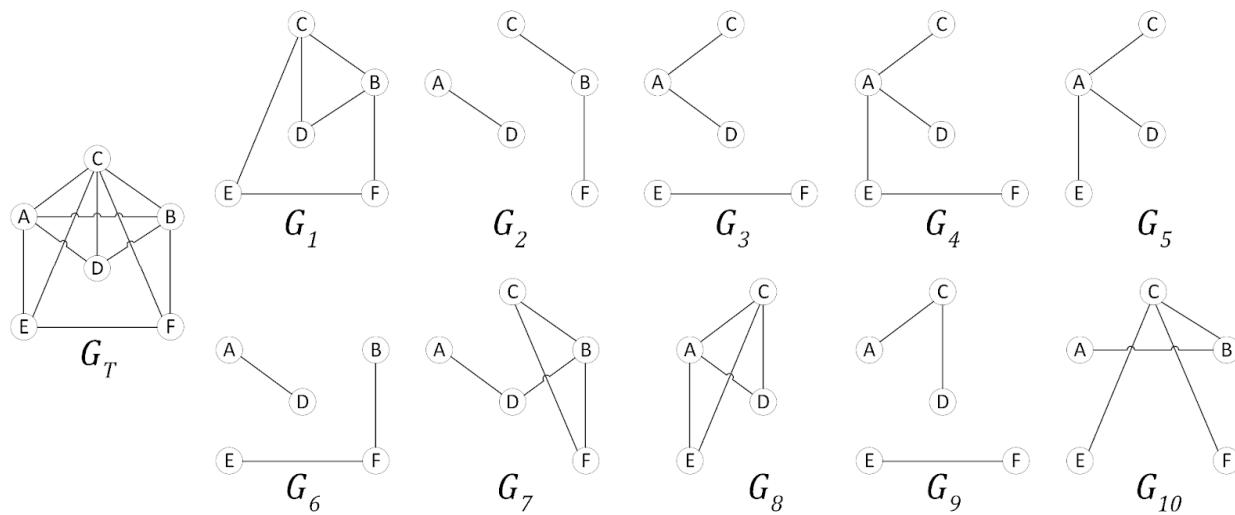


Figure 1. Example of a dynamic network model.

	A B C D E F	A B C D E F	A B C D E F	A B C D E F	A B C D E F
A	- 0 0 0 0 0	A - 0 0 1 0 0	A - 0 1 1 0 0	A - 0 1 1 1 0	A - 0 1 1 1 0
B	0 - 1 1 0 1	B 0 - 1 0 0 1	B 0 - 0 0 0 0	B 0 - 0 0 0 0	B 0 - 0 0 0 0
C	0 1 - 1 1 0	C 0 1 - 0 0 0	C 1 0 - 0 0 0	C 1 0 - 0 0 0	C 1 0 - 0 0 0
D	0 1 1 - 0 0	D 1 0 0 - 0 0	D 1 0 0 - 0 0	D 1 0 0 - 0 0	D 1 0 0 - 0 0
E	0 0 1 0 - 1	E 0 0 0 0 - 0	E 0 0 0 0 - 1	E 1 0 0 0 - 1	E 1 0 0 0 - 0
F	0 1 0 0 1 -	F 0 1 0 0 0 -	F 0 0 0 0 1 -	F 0 0 0 0 1 -	F 0 0 0 0 0 -

	M_1	M_2	M_3	M_4	M_5

	M_6	M_7	M_8	M_9	M_{10}

Figure 2. Sequence of connectivity matrixes of the dynamic network model in Figure 1.

The Delay matrix (D) of existing routes between a source device s and a destination device d in the domain G_T can be obtained as follows:

$$D(s, d) = \sum_{i=1}^{|V|-1} \sum_{j=2}^{|V|} \{\delta_{ij} x_{ij}\}_{t_{ij}} \quad (1)$$

The Load matrix (L) of all existing routes between a source device s and a destination device d in the domain G_T can be obtained as follows:

$$L(s, d) = \sum_{i=1}^{|V|-1} \sum_{j=2}^{|V|} \{\lambda x_{ij}\}_{t_{ij}} + \lambda \quad (2)$$

where

δ_{ij} represents the forwarding delay from device i to device j .

λ represents the message size.

$x_{ij} = 1$ if devices $\{i, j\}$ contact at any time and that link is used in a route between the source device s and the destination d (i.e., device i decides to forward a copy of the message to device j).

$x_{ij} = 0$ if devices $\{i, j\}$ do not connect or if the link is not used in any route between the source device s and destination d .

t_{ij} represents the encounter time of device i and device j .

We suggest the Dynamic Shortest Path Method, drawing on the aforementioned factors. This approach aims to achieve a compromise between decreasing communication costs, guaranteeing equitable load distribution, and obtaining the ideal delivery delay. To do this, we employ a tuning parameter α that enables the three factors to be considered when determining the optimum route between source and destination nodes. The variables $\{\alpha_1, \alpha_2, \alpha_3\}$ can also be changed depending on the analysis to give the three variables different relative weights. Next, we list the equations that describe the dynamic shortest path approach that is suggested:

- Delivery Delay:

$$\alpha_1 \min(D(s, d)) = \alpha_1 \min \left(\sum_{i=1}^{|V|-1} \sum_{j=2}^{|V|} \{ \delta_{ij} x_{ij} \}_{t_{ij}} \right) \quad (3)$$

- Load Balancing:

$$\alpha_2 \min \left(\frac{1}{\sum_{i=1}^{|V|-1} \sum_{j=2}^{|V|} x_{ij}} \sum_{i=1}^{|V|-1} \sum_{j=2}^{|V|} \left(\{ \delta_{ij} x_{ij} \}_{t_{ij}} - \frac{1}{\sum_{i=1}^{|V|-1} \sum_{j=2}^{|V|} x_{ij}} \sum_{i=1}^{|V|-1} \sum_{j=2}^{|V|} \{ \delta_{ij} x_{ij} \}_{t_{ij}} \right)^2 \right)^{1/2} \quad (4)$$

- Communication Overhead:

$$\alpha_3 \min(L(s, d)) = \alpha_3 \min \left(\sum_{i=1}^{|V|-1} \sum_{j=2}^{|V|} \left(\{ \lambda x_{ij} \}_{t_{ij}} \right) + \lambda \right) \quad (5)$$

subject to the following restrictions:

$$\delta_{ij} > 0, \forall i, j \in V$$

$$\lambda > 0$$

$$x_{ij} \in \{0, 1\}, \forall i, j \in V$$

$\sum_{j=2}^{|V|} x_{1j} = 1$ (the shortest path only uses one link from the source device).

$\sum_{i=1}^{|V|-1} x_{i|V|} = 1$ (the shortest path only uses one link to the destination device).

$\sum_{i=1}^{|V|-1} x_{ik} = \sum_{j=2}^{|V|} x_{kj}, \forall k \in \{2, 3, \dots, |V| - 1\}$ (in the shortest path, if a link arriving at device k is used, then a single link leaving k will be used).

$t_{1,2} \leq t_{2,3} \leq \dots \leq t_{(T-1),T} \leq$ (represents the connections of intermediary devices based on the time order).

The example of Figure 3 lists the values of the three variables for the four pathways (in different colors) from device A to device B depending on the time order to demonstrate the efficacy of the suggested strategy.

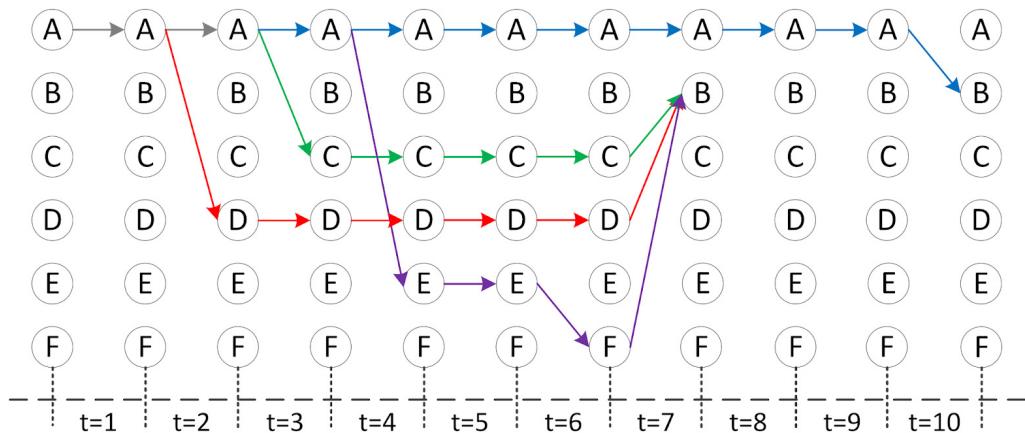


Figure 3. Illustration of the paths from device A to device B based on the time order in Figure 2.

- Number of Paths between devices A and B is $p(A, B) = \sum_{i=1}^{|V|-1} M_T^i(A, B) = 4$.
- Delay matrix: $D(A, B) = \sum_{i=1}^{|V|-1} \sum_{j=2}^{|V|} \{\delta_{ij} x_{ij}\}_{t_{ij}} =$

$$\begin{aligned} \{A, B\} &\Rightarrow \delta_{AB} x_{AB} = 10t \text{ seconds} \\ \{A, C, B\} &\Rightarrow \delta_{AC} x_{AC} + \delta_{CB} x_{CB} = 3t + 4t = 7t \text{ seconds} \\ \{A, D, B\} &\Rightarrow \delta_{AD} x_{AD} + \delta_{DB} x_{DB} = 2t + 5t = 7t \text{ seconds} \\ \{A, E, F, B\} &\Rightarrow \delta_{AE} x_{AE} + \delta_{EF} x_{EF} + \delta_{FB} x_{FB} = 4t + 2t + t = 7t \text{ seconds} \end{aligned}$$

- Load balancing (LB):

$$\begin{aligned} \{A, B\} &\Rightarrow LB = \left(\frac{1}{4} ((10t - 10t)^2) \right)^{1/2} = 0t \text{ seconds} \\ \{A, C, B\} &\Rightarrow LB = \left(\frac{1}{2} ((3t - 3.5t)^2 + (4t - 3.5t)^2) \right)^{1/2} = 0.5t \text{ seconds} \\ \{A, D, B\} &\Rightarrow LB = \left(\frac{1}{2} ((2t - 3.5t)^2 + (5t - 3.5t)^2) \right)^{1/2} = 1.5t \text{ seconds} \\ \{A, E, F, B\} &\Rightarrow LB = \left(\frac{1}{3} ((4t - 2.33t)^2 + (2t - 2.33t)^2 + (1t - 2.33t)^2) \right)^{1/2} = \\ &= 1.5764t \text{ seconds} \end{aligned}$$

- Load matrix: $L(A, B) = \sum_{i=1}^{|V|-1} \sum_{j=2}^{|V|} \left(\{\lambda x_{ij}\}_{t_{ij}} \right) + \lambda =$

$$\{A, B\} \Rightarrow load = \lambda x_{AB} + \lambda = 2\lambda \text{ bytes}$$

$$\{A, C, B\} \Rightarrow load = \lambda x_{AC} + \lambda x_{CB} + \lambda = 3\lambda \text{ bytes}$$

$$\{A, D, B\} \Rightarrow load = \lambda x_{AD} + \lambda x_{DB} + \lambda = 3\lambda \text{ bytes}$$

$$\{A, E, F, B\} \Rightarrow load = \lambda x_{AE} + \lambda x_{EF} + \lambda x_{FB} + \lambda = 4\lambda \text{ bytes}$$

As shown in Table 2 considering the influence of delivery delay, communication overhead, and load balancing on routing performances, we observe that there are three paths with the shortest delivery delay ($7t$) from device A to device B: $\{A, C, B\}$, $\{A, D, B\}$ and $\{A, E, F, B\}$. However, the length of paths $\{A, C, B\}$ and $\{A, D, B\}$ is shorter than that of path $\{A, E, F, B\}$. Moreover, the longest forwarding delay of the path $\{A, C, B\}$ is three time periods, from $t = 4$ to $t = 6$ (load balancing = $0.5t$), which is shorter than that of path $\{A, D, B\}$, with four time periods, from $t = 3$ to $t = 6$ (load balancing = $1.5t$), so $\{A, C, B\}$ have better

load balancing. Finally, we observe that path $\{A, B\}$ has the shortest hop number but the longest delivery delay ($10t$). In summary, path $\{A, C, B\}$ is the best path from device A to device B considering the trade-off between delivery delay, communication overhead, and load balancing.

Table 2. Dynamic shortest path identification from device A to device B.

Title 1	Latency	Load Balancing	Overhead
$\{A, B\}$	$10t$	$0t$	2λ
$\{A, C, B\}$	$7t$	$0.5t$	3λ
$\{A, D, B\}$	$7t$	$1.5t$	3λ
$\{A, E, F, B\}$	$7t$	$1.5764t$	4λ

Using flooding or epidemic routing, device B should receive 4 copies of the message. One from device A at time $t = 10$ and three copies from C, D and F at $t = 7$, assuming a total load of $CO - p_{AB}\lambda + \lambda = \sum_{i=1}^{|V|-1} \sum_{j=2}^{|V|} (\{\lambda x_{ij}\}_{t_{ij}} + \lambda) - p_{AB}\lambda + \lambda = 12\lambda - 4\lambda + \lambda = 9\lambda$ bytes.

However, by selecting the best route, overhead improvement can be obtained as $\sum_{i=1}^{|V|-1} \sum_{j=2}^{|V|} (\{\lambda x_{ij}\}_{t_{ij}} + \lambda) - p_{AB}\lambda + \lambda - \min(\sum_{i=1}^{|V|-1} \sum_{j=2}^{|V|} \{\lambda x_{ij}\}_{t_{ij}} + \lambda) = 6\lambda$ bytes, which is an improvement of 33%.

3. Influence Metrics

Node centrality or impact is the process of classifying nodes or devices in a network according to their importance or effect. This statistic evaluates a device's significance or effect on the network as a whole. There are several ways to determine device centrality, and each one takes a different strategy to pinpoint the most important nodes. These methods are intended to identify the importance and function of each device inside the network. Our goal is to order the network nodes according to their impact, allowing for the creation of new routing protocols that improve the QoS of the network. We can increase QoS levels by using just the most powerful nodes for data forwarding.

Degree and Closeness are two conventionally used common centrality measurements. While the Closeness metric considers the global topological information, Degree centrality is based on local topological information and assesses the node's local importance in the network. These influence measures are defined as follows for a network $G = (V, E)$:

3.1. Local Influence

To determine a node's local influence, it is straightforward to assess the centrality of the node within the network. The quantity of direct connections a node has to other nodes determines its degree of centrality. The following is the mathematical formula for determining the degree centrality of a given node j :

1. If it is an unweighted and undirected network,

$$D(j) = \sum_i^{|V|} x_{ij} \quad (6)$$

where $x_{ij} = x_{ji} = 1$ if and only if nodes i and j are connected; $x_{ij} = x_{ji} = 0$ otherwise.

2. If it is a weighted and undirected network,

$$D(j) = \sum_i^{|V|} \omega_{ij} x_{ij} \quad (7)$$

where

$\omega_{ij} = \omega_{ji} = 1$ is the cost of the link (i, j) .

$x_{ij} = x_{ji} = 1$ if and only if nodes i and j are connected; $x_{ij} = x_{ji} = 0$ otherwise.

Dynamic Degree Metric

Since nodes in static binary networks may only be either linked or disconnected, the traditional degree centrality statistic was initially created for those types of networks. The interactions between nodes in DSCN networks are not binary, though, and the topology of these networks is continually changing. Individuals in DSCNs often have a small number of regular connections in addition to sporadic encounters. Connections made often have a tendency to be stronger than those made infrequently. As a result, if a person interacts with their neighbors regularly and has a larger number of neighbors, they are more likely to be able to engage with new individuals. Therefore, it is crucial to consider the following three behavioral traits in DSCNs in order to effectively evaluate a device's local influence: a large number of neighbors, a high number of neighbor contact instances, and a positive evolution in the frequency of neighbor contact over time.

Regression analysis is a commonly used method for investigating data distribution patterns in the fields of information science and statistical modeling [27,28]. This theory has been used by us to investigate the connections between the temporal changes in the connection time distributions among devices.

Considering a device v which has $\sum_{k=1, k \neq v}^{|V|} \{x_{vk}\}_T$ connections with neighbors during the time span T , $\sum_{k=1, k \neq v}^{|V|} \{x_{vk}\}_{t_i}$ is the number of neighbors of v during the snapshot t_i and $\sum_{k=1, k \neq v}^{|V|} \{\varphi_{vk} x_{vk}\}_T$ is the frequency of contact times between device v and its neighbors during the time span T . Then, the evolution trend of connections of device v during the time span T is defined as follows:

$$\tau_D(v) = \frac{F_T \sum_{i=1}^T \left(t_i \sum_{k=1, k \neq v}^{|V|} \{x_{vk}\}_{t_i} \right) - \sum_{i=1}^T t_i \sum_{i=1}^T \left(\sum_{k=1, k \neq v}^{|V|} \{x_{vk}\}_{t_i} \right)}{F_T \sum_{i=1}^T t_i^2 - \left(\sum_{i=1}^T t_i \right)^2} \quad (8)$$

The device tends to increase the frequency of interaction with its neighbors over the time period T if the value of $\tau_T(v)$ is positive. An equitable distribution of contact frequency with a specific node's neighbors is indicated by a value of 0, while a negative value denotes a reduction in contact frequency over time. Therefore, devices will have a more favorable contact dynamic if they have more neighbors and increased contact probabilities with those neighbors. For that, we propose the Dynamic Degree metric of a device v ($DTE(v)$), which takes into account the following properties, as more interactions with neighbors lead to stronger links with them:

$$DTE(v) = \alpha \tau_D(v) + (1 - \alpha) \sum_{i=1}^{F_T} \frac{\sum_{k=1, k \neq v}^{|V|} \{x_{vk}\}_{t_i} - F_T^{-1} \sum_{k=1, k \neq v}^{|V|} \{\varphi_{vk} x_{vk}\}_T}{\left(F_T^{-1} \sum_{i=1}^{F_T} \left(\sum_{k=1, k \neq v}^{|V|} \{x_{vk}\}_{t_i} - F_T^{-1} \sum_{k=1, k \neq v}^{|V|} \{\varphi_{vk} x_{vk}\}_T \right)^2 \right)^{1/2}} \quad (9)$$

The user-defined parameter value $\alpha \in [0, 1]$ regulates the importance or weight of connection evolution and the frequency of contact moments. Please note that this value adheres to a zero-sum condition, meaning that increasing the weight of one element would inherently decrease the weight of the other factor. If significant patterns of growth or decline in the metric's values are observed, then $\alpha \rightarrow 1$. If the data show uniform distributions across time, $\alpha \rightarrow 0$.

3.2. Global Influence

A typical global centrality metric known as 'Closeness' uses the shortest routes to calculate distances between each node and every other node in the network. However, due to the specific characteristics of DSCNs, this statistic often results in inaccurate estimates. To address these issues, we have developed a unique approach to calculating the shortest

paths that more accurately represents the information propagation patterns within DSCNs over time. With this approach, we subsequently formulated a refined definition of the global influence measure, accounting for these distinctive qualities of DSCNs.

The frequent partitioning of topologies and intermittent connections that characterize DSCNs often lead to higher storage capacity utilization. The constrained storage space on a device can present a hurdle for efficient routing, particularly if it receives messages more quickly than it can transmit them to the next relay device. This situation can result in uneven load balancing, which significantly impacts the overall routing efficiency within DSCNs. Furthermore, employing an excessive number of devices as relays for a message can introduce unnecessary communication overhead, exacerbating routing performance issues. Hence, achieving a balance between delivery delay, load distribution, and communication overhead becomes imperative when making routing decisions in DSCNs.

3.2.1. Dynamic Closeness Metric

One centrality measure that relies on distance is Closeness. It is determined by averaging the shortest distances (involving the fewest nodes, thus minimizing overhead) from a specific node to all other nodes in the network. This is equivalent to summing the shortest distances (d_{short}) and dividing by the number of nodes (referred to as the network order, denoted as $|V|$), minus one, as node j itself is excluded from this calculation:

$$\begin{aligned} \text{Average path lengths}(v) &= \frac{\sum_{k=1, k \neq v}^{|V|-1} \min(D(v,k))}{|V|-1} = \\ &= \frac{\sum_{k=1, k \neq v}^{|V|-1} \min\left(\sum_{i=1}^{|V|-1} \sum_{j=2}^{|V|} \left(\{\lambda x_{ij}\}_{t_{ij}}\right) + \lambda\right)}{|V|-1} \end{aligned} \quad (10)$$

where $x_{ij} = 1$ if the link $\{i, j\}$ is used in a route between the source node v and the destination node k and $x_{ij} = 0$ otherwise.

The lower the above value, the closer a node is to the center of the network. For this reason, closeness is defined as the reciprocal of Equation (10), so that the more centered a node v is in the network, the higher its closeness metric is:

$$C_{CLO}(v) = \frac{|V| - 1}{\sum_{k=1, k \neq v}^{|V|-1} \min\left(\sum_{i=1}^{|V|-1} \sum_{j=2}^{|V|} \left(\{\lambda x_{ij}\}_{t_{ij}}\right) + \lambda\right)} \quad (11)$$

The Closeness measure in the network is strongly related to the rate of information propagation between devices as well as the timeframes at which messages are transmitted over the network. This metric offers a means to evaluate how accessible a device is within the network.

Let us examine an example calculation for the connected devices $\{A, B, C, D, F\}$ during G_7 , as shown in Figure 4.

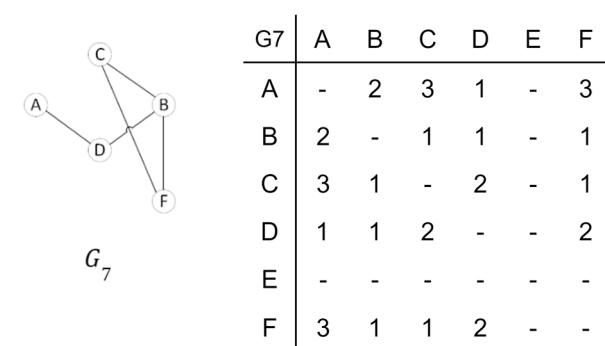


Figure 4. Network connections during instant G_7 , based on the example of Figure 1.

$$C_{CLO}(A) = \frac{6-1}{\lambda(2+3+1+3)+\lambda} = \frac{5}{10\lambda}$$

$$C_{CLO}(B) = \frac{6-1}{\lambda(2+1+1+1)+\lambda} = \frac{5}{6\lambda}$$

$$C_{CLO}(C) = \frac{6-1}{\lambda(3+1+2+1)+\lambda} = \frac{5}{8\lambda}$$

$$C_{CLO}(D) = \frac{6-1}{\lambda(1+1+2+2)+\lambda} = \frac{5}{7\lambda}$$

$$C_{CLO}(F) = \frac{6-1}{\lambda(3+1+1+2)+\lambda} = \frac{5}{8\lambda}$$

Therefore, the ranking of nodes according to their closeness index is as follows:

$$\text{Ranking} = \{B, D, C, F, A\} \quad (12)$$

Regrettably, calculating the Closeness metric (as indicated in Equation (11)) for each node in a network requires knowledge of the distances between all pairs of vertices. In disconnected networks, where nodes belong to distinct components or subnetworks that lack any larger linked subnetwork, the distance between two nodes is traditionally considered infinite, as depicted in the example shown in Figure 1, rendering Closeness inapplicable. As a result, the reciprocal becomes 0, and the sum in the equation (Equation (11)) diverges. Devices often belong to various components, rendering Closeness values irrelevant for all devices in the network except those within the largest component. Consequently, the computation of the Closeness metric must exclude devices that are part of smaller components.

Through the utilization of dynamic shortest paths, we can overcome the limitation of the conventional Closeness metric in disconnected networks. By employing a method that accumulates the reciprocal of path costs instead of the reciprocal of the total path cost, we can redefine the Closeness metric. This approach takes into consideration communication costs or overhead. As a result, the Dynamic Closeness metric is defined as the sum of the reciprocals of distances, rather than the reciprocal of the sum of distances:

$$C_{CLO}(v) = \sum_{k=1, k \neq v}^{|V|} \left(\frac{1}{\min \left(\sum_{i=1}^{|V|-1} \sum_{j=2}^{|V|} \left(\{\lambda x_{ij}\}_{t_{ij}} \right) + \lambda \right)} \right) \quad (13)$$

The adoption of the Dynamic Closeness measure prevents situations where an infinite distance dominates over other distances. Additionally, this measure can be standardized by considering that in a network with a star topology, the maximum value is achieved by the central node, which is equal to $|V| - 1$ (the longest distance possible in a network with $|V|$ nodes is $|V| - 1$, i.e., in a chain-connected network). The standardized value of the central node in a star network is 1, while the value for the leaf nodes is

$$\frac{1}{|V|-1} \left(\frac{1}{1} + (|V|-2) \frac{1}{2} \right) = \frac{|V|}{2(|V|-1)} \quad (14)$$

Thus, the centrality index is now defined by

$$C_{CLO}(v) = \frac{1}{|V|-1} \sum_{k=1, k \neq v}^{|V|} \left(\frac{1}{\min \left(\sum_{i=1}^{|V|-1} \sum_{j=2}^{|V|} \left(\{\lambda x_{ij}\}_{t_{ij}} \right) + \lambda \right)} \right), \quad (15)$$

subject to $t_{k,k+1} \leq t_{k+1,k+2} \leq \dots \leq t_{|V|-1,|V|}$.

Thus, considering $(|V|-1)^{-1} \sum_{k=1, k \neq v}^{|V|} \left(\min \left(\sum_{i=1}^{|V|-1} \sum_{j=2}^{|V|} \left(\{\lambda x_{ij}\}_{t_{ij}} \right) + \lambda \right) \right)^{-1}$, the closeness evolution of device v during the time span T is defined as

$$\tau_C(v) = \frac{F_T \sum_{l=1}^T \left(t_l (|V|-1)^{-1} \sum_{k=1, k \neq v}^{|V|} \left(\min \left(\sum_{i=1}^{|V|-1} \sum_{j=2}^{|V|} \left(\{\lambda x_{ij}\}_{t_l} \right) + \lambda \right) \right)^{-1} \right)}{F_T \sum_{l=1}^T t_l^2 - (\sum_{l=1}^T t_l)^2} - \frac{\sum_{l=1}^T t_l \sum_{l=1}^T \left((|V|-1)^{-1} \sum_{k=1, k \neq v}^{|V|} \left(\min \left(\sum_{i=1}^{|V|-1} \sum_{j=2}^{|V|} \left(\{\lambda x_{ij}\}_{t_l} \right) + \lambda \right) \right)^{-1} \right)}{F_T \sum_{l=1}^T t_l^2 - (\sum_{l=1}^T t_l)^2} \quad (16)$$

In conclusion, we put forward the Dynamic Closeness for device v ($CTE(v)$), which takes into account the fluctuations in the Closeness measure, as defined:

$$CTE(v) = \alpha \tau_C(v) + (1 - \alpha) \frac{1}{F_T} \sum_{l=1}^{F_T} \left(\frac{1}{|V|-1} \sum_{k=1, k \neq v}^{|V|} \left(\frac{1}{\min \left(\sum_{i=1}^{|V|-1} \sum_{j=2}^{|V|} \left(\{\lambda x_{ij}\}_{t_l} \right) + \lambda \right)} \right) \right) \quad (17)$$

where $x_{ij} = 1$ if the link $\{i, j\}$ is used in any route between nodes v and k and $x_{ij} = 0$ otherwise.

3.2.2. Social Closeness Metric

Human social relationships typically display greater stability than transmission links between mobile devices due to the complex network conditions present, for instance, in Opportunistic Mobile Social Networks (OppMSNs), characterized by intermittent connectivity that results in unstable end-to-end paths between devices. As a result, OppMSN routing decisions may be made more efficient using social indicators.

It is noticed that people keep both regular and sporadic interactions within their social surroundings. Information propagation greatly depends on the degree of contact between nodes. If the sender often communicates with the destination device, the sender may be aware of the times when they are most likely to run across the destination or nodes that are very likely to cross paths with the destination in the future [29,30]. Conversely, the likelihood of two devices knowing one another improves if they have a greater number of friends in common.

We examine the devices' past contacts in order to develop the Opportunistic Relationship Index (ORI), a social metric that is derived from important structural characteristics of a complex network, specifically the contact durations between devices, their shared neighbors, and distances [31]. In order to reflect the possibility of establishing a connection between devices v and k , the score is calculated as shown in Equation (17) for each pair of unconnected devices v and k . In this equation, $\sum_{i=1}^T \{\varphi_{vk} x_{vk}\}_{t_i}$ represents the frequency of contact occurrences between devices v and k within the time span T , and $\sum_{i=1}^{|V|-1} \sum_{j=2}^{|V|} x_{ij}$ denotes the distance matrix of existing paths between the two devices:

$$ORI_T(v, k) = \begin{cases} \left(\sum_{i=1}^T \{\varphi_{vk} x_{vk}\}_{t_i} \right)^{\frac{|\Gamma(v) \cap \Gamma(k)| + 1}{2}} & \text{if } \Gamma(v) \cap \Gamma(k) \neq 0 \\ \left(\sum_{i=1}^T \{\varphi_{vk} x_{vk}\}_{t_i} \right)^{\frac{1}{\min(\sum_{i=1}^{|V|-1} \sum_{j=2}^{|V|} x_{ij})}} & \text{otherwise,} \end{cases} \quad (18)$$

where $\Gamma(v) = \sum_{i=1}^T \sum_{l=1, l \neq v}^{|V|} \{x_{vl}\}_{t_i}$ and $\Gamma(k) = \sum_{i=1}^T \sum_{l=1, l \neq k}^{|V|} \{x_{kl}\}_{t_i}$ represent the respective sets of neighbors for devices v and k , respectively, over the time span T .

Figure 5 in this context shows the subnetwork created from Appendix A Figure A1, only showing the four current routes connecting device A and device B , with the weights denoting the calculated ORI.

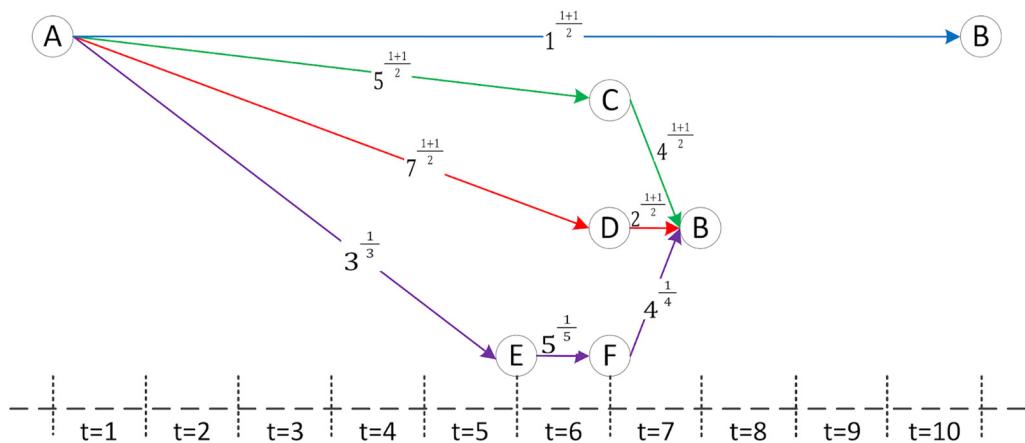


Figure 5. Illustration of the paths from device A to device B of example in Figure 1, where the weights represent the ORI.

Then, we present the shortest path based on opportunistic relationships, which include both opportunistic relationships and communication costs. We invert the weights to find the path with the lowest weight since the weight denotes the ORI. We use a tuning parameter to ensure that the ORI and the number of intermediary devices affect the choice of the best path, incorporating both communication cost and ORI. The following is the social measure used in Equation (18) to determine the opportunistic cost ($Cost_{opp}$) of a path between a source device (s) and a destination device (d):

$$Cost_{opp}(s, s+1, \dots, d) = \sum_{i=s}^{d-1} \left(\frac{1}{ORI_T(i, i+1)} \right)^\alpha, \quad (19)$$

subject to $t_{s,s+1} \leq t_{s+1,s+2} \leq \dots \leq t_{d-1,d}$.

The ORI between devices i and j before their interaction with the next relay device is indicated by the symbol ORI_T in Equation (18). The moment at which device i and device j first made contact is represented by $t_{i,j}$, and the sequence in which connections between intermediary devices are made is indicated by $t_{s,s+1} \leq t_{s+1,s+2} \leq \dots \leq t_{d-1,d}$.

We further normalize the series by dividing the geometric mean of ORI_T by the highest ORI_T , as the routing choice may not function effectively with a low ORI_T value across devices. The tuning parameter is this normalized value, which enables us to gauge the degree of dispersion between low and high ORI_T values. As a result, the following is how the definition of α is stated:

$$\alpha_{s,s+1,\dots,d} = \frac{\left(\prod_{i=s}^{d-1} ORI_T(i, i+1) \right)^{\frac{1}{|s,d|}}}{\max(ORI_T(s, s+1), ORI_T(s+1, s+2), \dots, ORI_T(d-1, d))}, \quad (20)$$

where $|s,d|$ represents the number of hops in the path from the source device s to the destination device d .

By adding considerations of the shortest pathways, social interactions between nodes, and communication cost, we use the opportunistic-based shortest path approach to expand the Closeness measure in this way. The Social Closeness metric (CLO_{opp}) is calculated in the manner described in the following:

$$CLO_{opp}(v) = \frac{1}{|V|-1} \sum_{k=1, k \neq v}^{|V|} \left(\frac{1}{\min(Cost_{opp}(v,k))} \right) = \\ = \frac{1}{|V|-1} \sum_{k=1, k \neq v}^{|V|} \left(\frac{1}{\min\left(\sum_{i=v}^{k-1} \left(\frac{1}{ORI_T(i,i+1)} \right)^\alpha\right)} \right), \quad (21)$$

subject to $t_{v,v+1} \leq t_{v+1,v+2} \leq \dots \leq t_{k-1,k}$, in which $\min(Cost_{opp}(v, k))$ first computes all the paths between v and k , then calculates the $Cost_{opp}$ of each path, and finally selects the minimum $Cost_{opp}$ among all.

Hence, taking into consideration the expression

$(|V| - 1)^{-1} \sum_{k=1, k \neq v}^{|V|} \left(\min \left(\sum_{i=v}^{k-1} \left(\frac{1}{\{ORI_T(i,i+1)\}_{t_i}} \right)^\alpha \right) \right)^{-1}$, we can evaluate a device v 's social closeness at the specified snapshot t_l . As a result, Equation (22) describes how the social closeness of device v changed throughout time T :

$$\tau_{opp}(v) = \frac{\frac{F_T \sum_{l=1}^T \left(t_l (|V|-1)^{-1} \sum_{k=1, k \neq v}^{|V|} \left(\min \left(\sum_{i=v}^{k-1} \left(\frac{1}{\{ORI_T(i,i+1)\}_{t_i}} \right)^\alpha \right) \right)^{-1} \right)}{F_T \sum_{l=1}^T t_l^2 - (\sum_{l=1}^T t_l)^2} - \frac{\sum_{l=1}^T t_l \sum_{l=1}^T \left((|V|-1)^{-1} \sum_{k=1, k \neq v}^{|V|} \left(\min \left(\sum_{i=v}^{k-1} \left(\frac{1}{\{ORI_T(i,i+1)\}_{t_i}} \right)^\alpha \right) \right)^{-1} \right)}{F_T \sum_{l=1}^T t_l^2 - (\sum_{l=1}^T t_l)^2}}{(|V|-1)^{-1} \sum_{k=1, k \neq v}^{|V|} \left(\min \left(\sum_{i=v}^{k-1} \left(\frac{1}{\{ORI_T(i,i+1)\}_{t_i}} \right)^\alpha \right) \right)^{-1}} \quad (22)$$

In conclusion, we provide the definition of the Social Closeness metric of a node v ($CTE_{opp}(v)$), which combines the dynamics of the Social Closeness metric in:

$$CTE_{opp}(v) = \tau_{opp}(v)^\alpha + \frac{1}{F_T} \sum_{l=1}^{F_T} \left(\frac{1}{|V|-1} \sum_{k=1, k \neq v}^{|V|} \left(\frac{1}{\min \left(\sum_{i=v}^{k-1} \left(\frac{1}{\{ORI_T(i,i+1)\}_{t_i}} \right)^\alpha \right)} \right) \right), \quad (23)$$

subject to $t_{v,v+1} \leq t_{v+1,v+2} \leq \dots \leq t_{k-1,k}$.

4. Results and Discussion

Our goal is primarily to discuss the dataset used to assess various QoS measures and the analytical process once we have shown the production of rankings with the most significant nodes based on different centrality metrics. We will next go through how we integrated these rankings into message routing.

Using a collection of Reality Mining datasets [32], the recommended algorithm's efficacy has been evaluated. These datasets embody a complex social system by capturing data from 100 mobile phones over a span of 9 months. The authors demonstrate how common Bluetooth-enabled mobile phones can be used to measure information access and utilization in a variety of settings, detect social patterns in users' daily activities, infer relationships, identify socially significant locations, and model organizational patterns.

However, it is worth noting that we utilize a modified version of the Reality dataset provided by the authors of [33]. As stated in the same reference, there is no significant activity before and after the timestamp ranges 1,094,545,041 and 1,111,526,856. Therefore, the simulations presented in this paper exclusively employ the data within that time interval, as shown in Table 3.

Table 3. Characteristics of the dataset.

Feature	Value
Number of devices	97
Environment	Campus
Dataset duration	246 days
Dataset duration used	196 days
Encounter prob. 1st 1/4 day	0.0003
Encounter prob. 2nd 1/4 day	0.0011
Encounter prob. 3rd 1/4 day	0.0019
Encounter prob. 4th 1/4 day	0.0012
Percentage of dataset duration for the Training Graph (GT)	75%
Percentage of dataset duration for the Probe Graph (GP)	25%
Network density	<0.5%
Number of contacts of the top 20 devices	4–9

Regarding the methodology for analyzing the dataset, it has undergone processing using a similar approach employed when implementing Machine Learning models. In this manner, the dataset is partitioned into two distinct non-overlapping graphs known as the training (GT) and probe (GP) graphs. The Training Graph (GT) is constructed by selecting a subset that represents the initial 75% of node interactions within the dataset. The remaining edges, not included in GT, constitute the Probe Graph (GP). Likewise, the edges included in GT are denoted as ET, while those in GP are referred to as EP, i.e., $E = ET + EP$. It should be noted that ET and EP are mutually exclusive; however, there may be overlapping nodes between GT and GP. For our experiments, we have allocated 75% of the edges to ET and the remaining 25% to EP.

The simulations have been conducted by simulating the GP (Probe Graph) with an implementation of a total of five routing algorithms. On one hand, we include the conventional ones typically used to evaluate QoS in OppNets, that is Spray and Wait (S&W), Prophet versions 1 and 2, and Epidemic (four algorithms). On the other hand, we incorporate a modified version of the S&W algorithm, which is evaluated three times based on a parameterized ranking of the most significant nodes according to the metrics described in the previous sections (Dynamic Degree, Dynamic Closeness, and Social Closeness), which sums a total of five routing algorithms. The results of the simulations will be presented in Sections 4.3.4–4.3.6.

The algorithms have been developed using The ONE (Opportunistic Network Environment) simulator [34] and can be accessed from a public repository located at <https://github.com/sito25/pubtesis.git>, (accessed on 21 September 2023) under the GNU Lesser General Public License v.3.0. This simulator is specifically designed for opportunistic networks and was initially developed at Aalto University in 2009. The ONE provides a wide range of capabilities, including the generation of node movements using various models, replication of message traffic and routing, cache management, and visualization of both mobility and message transmission through its graphical user interface. Additionally, it offers diverse reporting options, such as node movements, message transmission, and general statistics. Currently, it is collaboratively maintained by Aalto University and Technische Universität München, boasting a robust user community. The version utilized in our research is 1.6.0, implemented in Java.

4.1. Network Density

As mentioned earlier, opportunistic networks are a type of low-density networks that traditionally focus on self-organized and ad hoc mobile networks. These networks often experience frequent disruptions, delays, and intermittent connectivity, leading to a lack of end-to-end connections within the environment. In such scenarios, wireless devices can temporarily store information and forward it to other devices that are more likely to be within communication range of the intended destination when an opportunity for connection arises.

The density of an opportunistic network is determined by the ratio of edges present in a graph to the maximum number of edges the graph can contain. This ratio provides a conceptual idea of the network's connectivity in terms of link density. Specifically, network density is defined as the ratio of the number of connections to the maximum possible connections.

A network is considered dense when the number of links is close to the maximum possible, where every pair of devices is connected by a single link. Conversely, a network with few links is considered sparse. This concept provides an understanding of the level of connectivity and density within the network [35,36]. Therefore, to determine the maximum number of connections in the network, we can derive it as follows:

$$MaxConn = \frac{|V|(|V| - 1)}{2} \quad (24)$$

Let us now introduce the formula for calculating network density. The network density is calculated by dividing the total number of connections existing in the network $G(V, E)$ by the maximum possible number of connections that could potentially exist within the network. Let us examine the formula in detail:

$$\text{Density} = \frac{\frac{|E|}{|V|(|V|-1)}}{2} = \frac{2|E|}{|V|(|V|-1)} \quad (25)$$

In our study, Figure 6 displays the likelihood of the presence of complete pathways, which is linked to the density of connections within the Reality Mining datasets. This likelihood, denoted as $P(EE)$, can be defined as the ratio of the number of established end-to-end routes to the total number of possible connections.

$$P(EE) = \left(\sum_{s=1}^{|V|-1} \sum_{d=1}^{|V|-1} \sum_{i=1}^{|V|-1} M_T^i(s, d) \right) \left(\frac{|V|(|V|-1)}{2} \right)^{-1} \quad (26)$$

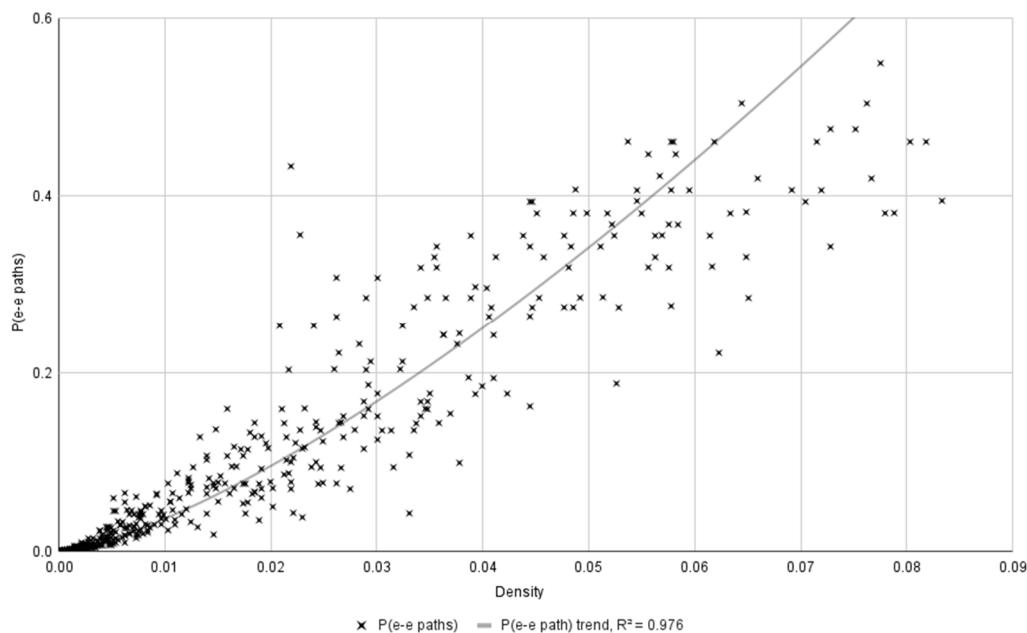


Figure 6. End-to-end paths probability in the datasets as a function of the connection density.

Figure 6 illustrates the network structure of the used dataset, which is characterized as a sparsely connected network with a consistent connection density among devices that does not exceed 8.5% throughout its dynamic nature. The majority of density values are below 1%. As a result, the upper limit for the probability of end-to-end connections remains below 55%, with a significant concentration below 0.5%. Therefore, it is imperative to consider that the utilized datasets define an opportunistic network with a very low density of connections.

4.2. Effectiveness Analysis of the Proposed Metrics

We may examine how successfully suggested metrics capture and quantify the intended attributes or characteristics of network devices by evaluating their efficacy. It enables us to assess how well these metrics capture the underlying ideas of ranking, connection, or social impact. As a result, we can evaluate their effectiveness and decide which metrics are more appropriate for achieving our study goals. Based on their capacity to capture the necessary elements of device centrality, these analyses aid in the selection of the most suitable centrality metrics. We can also establish whether these metrics provide useful information and can be relied upon when making network design decisions. For

performance evaluation, comparative analysis, hypothesis validation, and determining their practical relevance, evaluating the efficacy of local and global centrality measures in the network is essential.

4.2.1. Local Metrics

We compared Dynamic Degree during the studies to two benchmark measures, namely, Degree and Weighted Degree. Figure 7 uses the Reality Mining datasets to show the outcomes of these studies. The top-N devices are sorted according to their Weighted Degree and Dynamic Degree, and Figure 7a shows the plotted curves reflecting the average number of nearby devices among those devices.

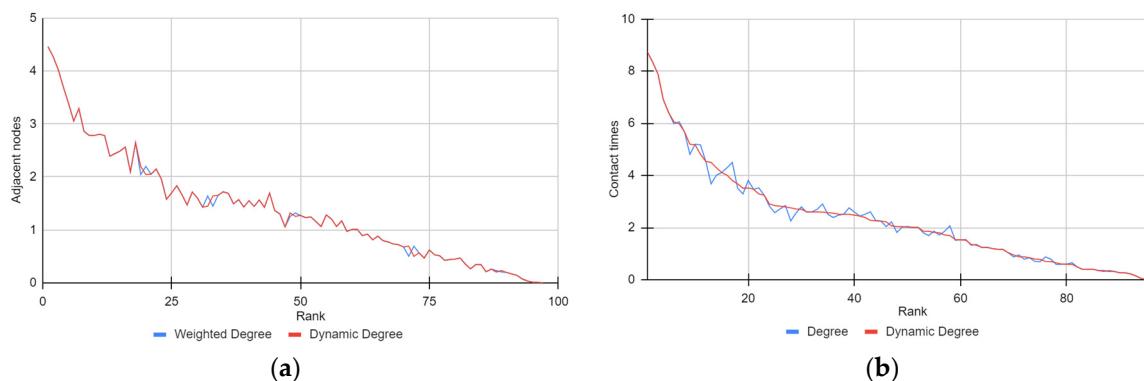


Figure 7. Effectiveness of local metrics: (a) Number of neighboring devices among the top-N devices; (b) average number of contacts among the top-N devices.

An effective local impact metric in Figure 7a should show a diminishing trend since a node with strong local influence often shows a high number of nearby nodes. The curves for the two measures shown in the picture, however, are rather near to one another. This resemblance could result from certain traits or distinctive qualities that the dataset itself possesses.

Figure 7b shows the plotted curves for the top-N devices ordered by their Degree and Dynamic Degree, which reflect the average number of connections. In light of the fact that a device with a strong local effect should interact with its neighbors often, the Dynamic Degree curve has a smoother downward slope and performs better than the Degree metric. Therefore, of the three influence measures, the Dynamic Degree meter performs the best since it establishes a balance between the number of nearby devices and the frequency of encounters.

4.2.2. Global Metrics

Using real-world datasets, we analyzed the distribution properties of suggested global influence indicators. Figure 8 displays the Complementary Cumulative Distribution Functions (CCDF) for the suggested global centralities, with the horizontal axis denoting the order of device effect. Here, it is clear that the distributions of the Closeness measure are not uniform when taking into account various centrality techniques. This suggests that the selection of the centrality approach affects the measures' distributions. These results help us determine the best tools to increase the effectiveness of data transmission in OppMSNs. These features can be used by routing algorithms to choose the most reliable device as a relay. As a result, the impact measures suggest various possible contributions of the same device to information propagation when combined with various centrality approaches. This connection between device impact and route design, in our opinion, is a key element in enabling effective information propagation inside OppMSNs.

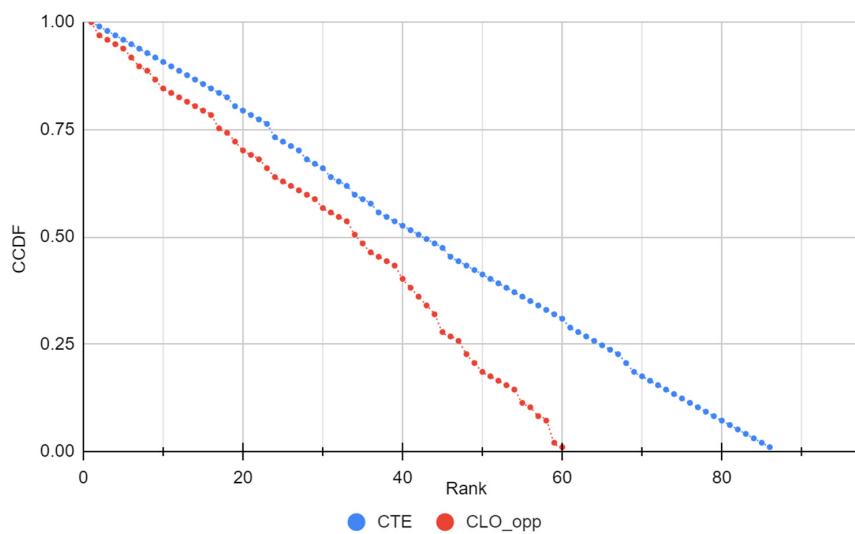


Figure 8. Illustration of Complementary Cumulative Distribution Function of proposed global centrality metrics.

4.2.3. Correlation Analysis between Local and Global Metrics

Our goal is to study the probable association between the ranking of global influence and the ranking of local impact through time and to address the consequences of anticipating global influence by looking at the consistency and predictability of human social qualities.

A statistical metric used to quantify the strength of the association between the relative changes of two variables is the correlation coefficient. It includes values between -1 and 1 . A complete negative correlation is represented by a correlation coefficient of -1 , and a perfect positive correlation is represented by a correlation coefficient of 1 . The absence of a linear link between the changes in the two variables is shown by a correlation value of 0 .

A statistical metric used to quantify the degree of linear association between two variables is the Pearson correlation coefficient. This coefficient reveals the nature and strength of the relationship, given that a change in one variable leads to a proportionate change in the other. When there is no apparent association, the Pearson coefficient returns a value of 0 .

Another correlation statistic used to assess rank correlation, which indicates the statistical dependence between the ranks of two variables, is Spearman's rank correlation coefficient. The Spearman coefficient measures the extent to which the relationship between variables can be characterized by a monotonic function, in contrast to a linear relationship where the rate of increase or decrease is constant. Without necessarily adhering to a consistent rate of change, it assesses how well the data align monotonically. The Spearman correlation evaluates the monotonic relationship between two variables, whether they are continuous or ordinal in nature, by considering the ranked values of each variable rather than the raw data. In a monotonic relationship, one of the following is true:

- As one variable increases, the value of the other variable decreases; or
- Conversely, as one variable increases, the value of the other variable also increases.

The methods employed by the two correlation coefficients are fundamentally distinct. While the Spearman coefficient considers both linear and monotonic correlations, the Pearson coefficient focuses exclusively on linear relationships between variables. Furthermore, Spearman utilizes rank-ordered variables, whereas Pearson employs the raw data values of the variables.

It is advisable to employ the Spearman coefficient instead of the Pearson coefficient when a scatterplot reveals a potential link that could be either monotonic or linear. Using the Spearman coefficient does not cause any harm, even if the data eventually demonstrate a perfect linear relationship. Nevertheless, choosing Pearson's coefficient might lead to missing crucial insights that Spearman could provide in cases where the connection is not

exactly linear. Therefore, as illustrated in Figure 9, we utilize the Spearman correlation coefficient to analyze the relationship between the ranks of Closeness and Degree throughout their dynamics. By establishing a consistent relative order of observations within each variable (e.g., first, second, third, and so on), it is intuitively understandable that the Spearman correlation between two variables becomes strong when observations possess similar (or identical, resulting in a correlation of 1) ranks. Conversely, the correlation is low when observations exhibit disparate ranks across the two variables (or entirely opposite rankings, leading to a correlation of -1).

Let $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ represent a collection of composite rankings from two distinct ranking lists, X and Y. The n raw scores (x_i, y_i) are converted to ranks $(R(x_i), R(y_i))$, and r_s is derived as follows:

$$r_s = \rho(R(X), R(Y)) = \frac{\text{cov}(R(X), R(Y))}{\sigma(R(X))\sigma(R(Y))} = \frac{(2n^2)^{-1} \sum_{i=1}^n \sum_{j=1}^n (R(x_i) - \bar{R}(x_i))(R(y_j) - \bar{R}(y_j))}{\left(\frac{1}{n} \sum_{i=1}^n (R(x_i) - \frac{1}{n} \sum_{i=1}^n R(x_i))^2\right)^{\frac{1}{2}} \left(\frac{1}{n} \sum_{i=1}^n (R(y_i) - \frac{1}{n} \sum_{i=1}^n R(y_i))^2\right)^{\frac{1}{2}}}, \quad (27)$$

where

ρ represents the application of the Pearson correlation coefficient to the rank-transformed variables.

$\text{cov}(R(X), R(Y))$ is the covariance of the rank variables.

$\sigma(R(X))$ and $\sigma(R(Y))$ represent the standard deviations of the rank-transformed variables.

With mean and standard deviation statistical significances (p -values) of 0.0035 and 0.0381, respectively, the majority of Spearman's rank correlation coefficients for the proximity measure in Figure 9 exhibit values near 0.75. This observation underscores the strong relationship between this measure and the Dynamic Degree. As the window period lengthens, the curve remains largely stable despite minor fluctuations in the correlation coefficients. Given that the Degree measure influences the Closeness metric in the dynamic context, it becomes conceivable to formulate an approximation of Closeness metric values through a combination of social-based and dynamic-based methods. These characteristics offer opportunities to enhance the accuracy of our closeness prediction strategies and to devise more effective forwarding algorithms in OppMSNs.

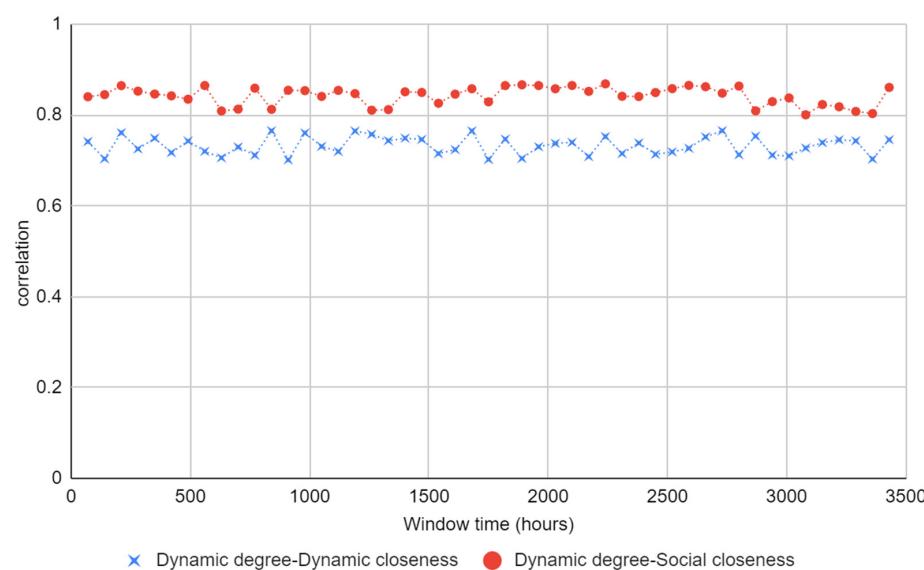


Figure 9. Ranking of Spearman correlations by varying snapshots during the time span.

4.3. Quality of Service Metrics Analysis

Latency, overhead, or hop count are QoS metrics that allow us to assess the performance and efficiency of the network. By quantifying the latency, we can determine how quickly data or messages are transmitted from the source to the destination. An overhead calculation helps assess the additional resources or data required to support the communication process. By quantifying overhead, we can identify potential inefficiencies or resource-intensive aspects of the network. This information is valuable for optimizing network performance and ensuring efficient resource utilization.

On the other hand, a hop count calculation helps evaluate the number of network devices or “hops” required for data to travel from the source to the destination. Lower hop counts generally imply a more direct and efficient routing path, resulting in reduced latency and improved overall network performance.

Overall, accurate calculation and analysis of latency, overhead, and hop count are crucial for performance evaluation, resource optimization, reliability assessment, routing efficiency analysis, identifying areas for improvement, and making informed decisions regarding network design and configuration.

In summary, precise calculations and analyses of latency, overhead, and hop count are pivotal for performance evaluation, resource optimization, reliability assessment, and routing efficiency analysis. To fulfill these requirements, we have implemented a set of algorithms to evaluate these QoS metrics and incorporate our proposals into The ONE—an opportunistic network simulator that is well-suited for simulating and studying such networks, as mentioned earlier.

4.3.1. Updating Training Matrix on Contact

The analysis for the used dataset involves simulating data analysis with a similar methodology to the one that is used to train Machine Learning models. This simulation includes dividing the data into a training subset and a test subset. As mentioned earlier, we allocate 75% of the simulation data to train a set of adjacency matrices. In these matrices, we simply increment the counter for each connection between pairs of nodes (device, another host). Algorithm 1 generates adjacency matrices that not only indicate connected nodes but also capture the connectivity capacity of nodes, making them likely to be chosen as message transporters to other nodes.

It is important to note that matrices are calculated with a specific frequency to ensure that different adjacency matrices are collected, reflecting the evolving connectivity of the nodes.

Algorithm 1. Updating training matrix on contact

Input:	TM (Training Matrix), N1 and N2 (new contact between two nodes)
Output:	TM (a new version of Training Matrix)

```

1 begin
2   |   if TM[N1, N2] == null then
3   |   |   TM[N1, N2] = 0;
4   |   if TM[N2, N1] == null then
5   |   |   TM[N2, N1] = 0; TM[N1, N2] +=1;
6   |   TM[N1, N2] +=1;
7   |   TM[N1, N2] +=1;
8 return TM

```

Computational complexity analysis allows us to evaluate the efficiency of an algorithm in terms of resource usage, such as time and memory [37]. By understanding the complexity, we can estimate how the algorithm will respond to varying input sizes. In the case of Algorithm 1, the complexity is simply O(1) because when two nodes meet each other, the corresponding cells of the adjacency matrix (TM[N1, N2] and TM[N2, N1]) are updated,

indicating how many times those nodes have been found throughout the considered period. The spatial complexity, on the other hand, is determined by the need to store contact information in an $N \times N$ matrix, where N is the number of nodes in the topology. However, since we also store M matrices to reflect the evolution of the adjacency matrix between nodes, the spatial complexity turns out to be $O(M \times N^2)$.

For each matrix, we generate an intermediate ranking of nodes with good connectivity. These rankings are then combined to obtain a final ranking, as described in Section 3, Influence Metrics, depending on the metric used. The rankings assist the routing algorithm in determining whether to forward a new copy of the package to the identified node.

4.3.2. Calculation of Friends Nodes upon Contact

In our proposal, the concept of a friend refers to a node that has connected with another node and is likely to reconnect within a relatively short time, based on the principle of temporal locality. To implement this idea of temporal locality, we introduce the friend concept, which involves setting a timer for each pair of connecting devices. This timer is activated after the nodes establish a connection. If they reconnect before the timer expires, we consider them friends who frequently connect. On the other hand, if the timer has expired by the time they encounter each other again, they are still friends but do not connect frequently.

The concept of a friend represents a list of nodes to which the given node has previously connected. This list is closely related to the adjacency matrices described in Algorithm 1, as both implementations rely on node connections. However, unlike the adjacency matrices, the friends list is not reset with each new matrix. Instead, it is continually updated throughout the simulation as the node forms new connections.

It is important to note that the friends list is not utilized during the training phase (when running Algorithm 2). Instead, it is used during testing, which will be explained further in Algorithm 3.

Moreover, the concept of a friend can be utilized to assess the extent to which these friends adhere to the notion of temporal locality. For instance, a connection counter between them can be employed within the context of their unexpired timer. In these simulations, we use the concept of friends as a list of nodes to which a specific node has connected throughout the entire simulation.

Algorithm 2. Calculation of Friend Nodes on contact

Input: $N1$ LNF (List of $N1$ friends), $N1$ and $N2$ (new contact between two nodes)
Output: $N1$ LNF (a new version of $N1$ friends), $T(N1,N2)$ (timeout between $N1$ and $N2$) and $TL(N1,N2)$ (temporal locality between $N1$ and $N2$)

```

1 begin
2   MAX_TIMEOUT = 20,000;
3   if  $N2$  in  $N1$  LNF then
4     elapsed_time = (current_time -  $T(N1,N2)$ );
5     if elapsed_time < MAX_TIMEOUT then
6        $TL(N1, N2) += 1;$ 
7     else
8       add  $N2$  to  $N1$  LNF;
9        $T(N1, N2) = current\_time;$ 
10  return  $N1$  LNF,  $T(N1,N2)$ ,  $TL(N1,N2)$ 
11  repeat with the input:  $N2$  LNF,  $N1$  and  $N2$ 
```

The complexity is also $O(1)$ because this algorithm simply updates the list of friends (LNF) of a node and vice versa when it connects to another. The space complexity is $O(N^2)$ because the friend list of each node is not renewed during the simulation, as is the case

with TM adjacency matrices during training. Instead, a friend is added to each contact in case they both meet and connect for the first time.

4.3.3. Routing Decision on Contact

The main objective of Algorithm 3 is to determine whether node A , given the connection between nodes A and B , should send a copy of the messages it carries to node B . This decision aims to minimize overhead. Instead of sending copies to every encountered node B , it is preferable to choose a node with better connectivity. Such a node is more likely to have greater access to a larger network.

Algorithm 3. Routing decision on contact

```

Input: N1 and N2 (new contact between two nodes), R (best nodes ranking), N1 LNF
Output: N2 messages queue
1 begin
2   for each message in N1 queue do
3     N3 = obtain message destination;
4     is_one_of_best_nodes = (N2 in R);
5     are_friends = (N3 in N1 LNF);
6     arrived_to_destination = (N2 == N3);
7     if is_one_of_best_nodes or are_friends or arrived_to_destination then
8       forward message to N2 queue;
9   end for
10  return N2 messages queue

```

This algorithm requires the use of different data structures to determine whether node N1 forwards the messages in its queue to N2. Therefore, these data structures are accessed in a single loop, where the processing of each one is decided. The time complexity is $O(N)$. In terms of spatial complexity, we need a list M for each node N to act as a message buffer, and a unique list to store the ranking of nodes. Consequently, the spatial complexity is $O(N \times M)$.

This approach restricts the generation of message copies by node A (a finite number in the case of S&W or unlimited in the case of Epidemic) to specific connections where node B exhibits one of the following three characteristics:

- Node B ranks among the top positions in the ranking obtained through Algorithm 1. Being a node with good connectivity, it is more likely to successfully deliver the packet to the intended recipient or another node that can assist in reaching the message's destination;
- The source and destination nodes of the message are friends. This indicates that they have previously connected and are likely to reconnect. Therefore, node A , carrying the message, is allowed to deliver a copy to node B ;
- Node B is the intended destination of the message. In this scenario, it is logical for node A to deliver the message to node B .

4.3.4. Packet Latency

Based on Figure 10, it becomes apparent that our algorithms achieve a decreased average packet delay in comparison to the original Spray and Wait protocol, with an average reduction of approximately 2% with respect to S&W and more than 10% with respect to Epidemic or Prophets v1 and v2 protocols. By integrating equivalent buffer sizes, minimizing overhead, and intelligent packet forwarding selection (unlike S&W, which disseminates packets to all encountered nodes), our approach facilitates expedited packet delivery to their intended destination.

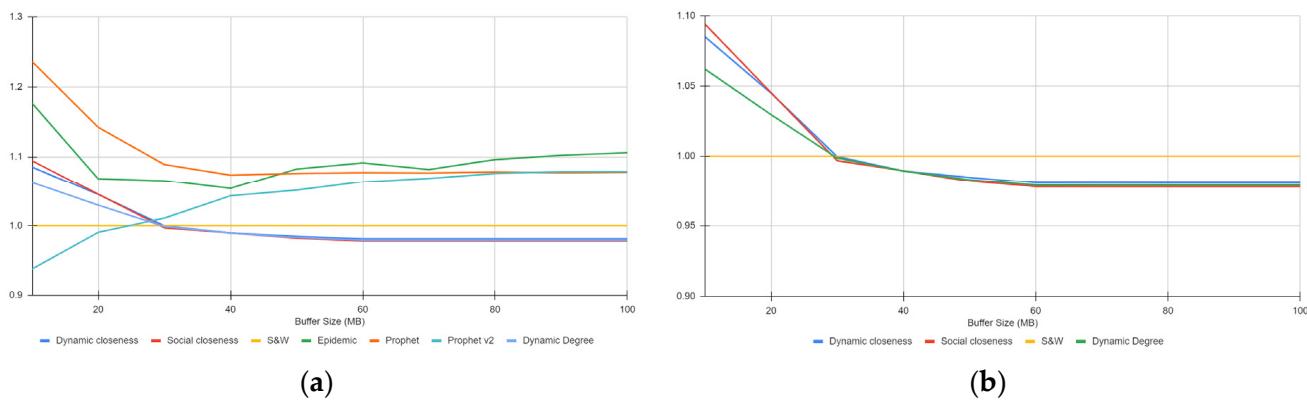


Figure 10. Average latency for different routing protocols, depending on the buffer size, normalized with respect to the results of the S&W routing protocol: (a) comparison among all the analyzed routing protocols; (b) comparison of the proposed metrics and S&W.

4.3.5. Path Length

As illustrated in Figure 11, the average number of hops taken by packets is influenced by the node selection procedure. While this effect may not be readily apparent from the graph, our algorithms have exhibited a slight decrease in the number of hops, surpassing S&W by more than 3%, and significantly outperforming the Epidemic or Prophet v1 and v2 routing protocols. This reduction in hops is accomplished through our meticulous node selection process and the principles we employ to determine packet forwarding. Consequently, the packets take fewer diversions along their routing path.

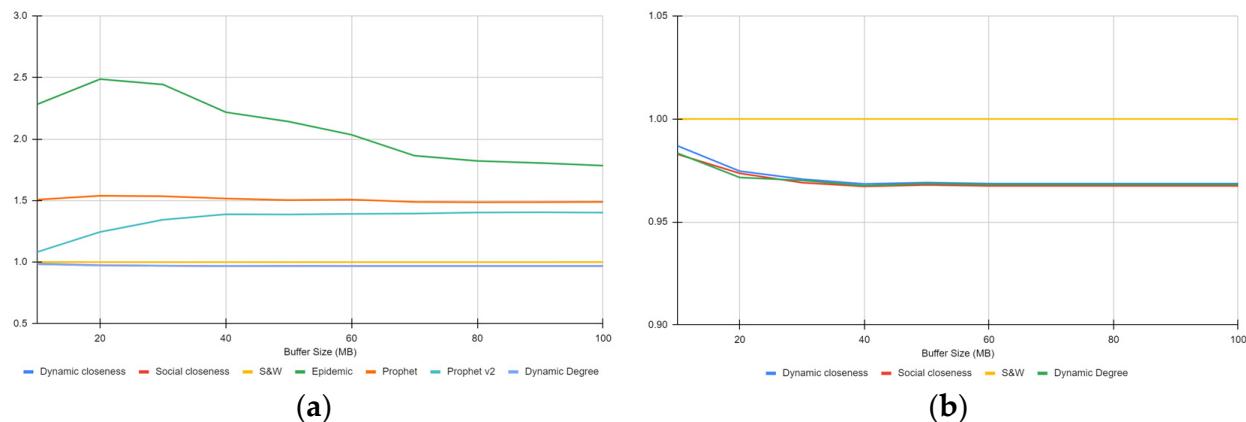


Figure 11. Hop count of delivered packets for different routing protocols, depending on the buffer size, normalized with respect to the results of the S&W routing protocol: (a) comparison among all the analyzed routing protocols; (b) comparison of the proposed metrics and S&W.

4.3.6. Route Overhead

The execution of Algorithm 3 within the proposed routing protocol determines the quality of the connection between the source device and destination node. This leads to a more restricted packet transmission approach compared to S&W, Epidemic, and Prophet v1 and v2 protocols, resulting in a great reduction in overhead (more than 32% less), as shown in Figure 12. Although the proposed algorithm retains the same number of message copies, they are no longer forwarded to all nodes but only to those that satisfy the conditions specified in Algorithm 3.

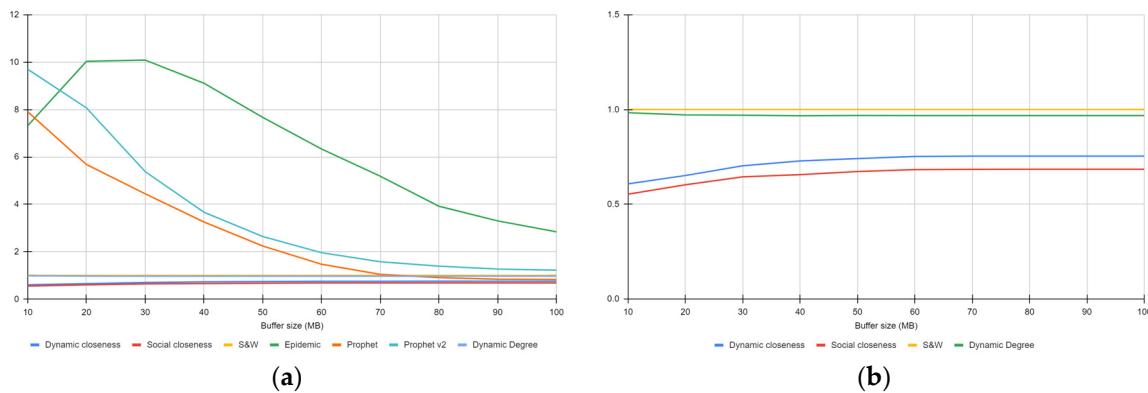


Figure 12. Overhead comparison of different routing protocols based on buffer size relative to the results of the S&W routing protocol: (a) comparison among all the analyzed routing protocols; (b) comparison of the proposed metrics and S&W.

4.3.7. Discussion of the Results

In the realm of networking protocols and QoS optimization, effective data transmission and low-latency routing are of utmost importance. This study deeply delves into the evaluation and refinement of routing algorithms to address these critical aspects, specifically comparing the performance of our proposed algorithms against well-established ones such as Spray and Wait, Epidemic, Prophet v1, and Prophet v2. The comparative analysis encompasses essential metrics, including packet latency, the number of hops, and overhead. It sheds light on the superior efficiency and effectiveness achieved by our meticulously designed algorithms. Subsequently, the discussion encapsulates noteworthy findings and implications of our research, illuminating the promising advancements in network optimization and reliability.

Concerning packet latency, our algorithms significantly reduce average packet delay compared to the original Spray and Wait protocol, with an average reduction of approximately 2% compared to Spray and Wait, and more than 10% compared to Epidemic or Prophets v1 and v2 protocols. Additionally, our algorithms demonstrate a reduction in the number of hops, surpassing Spray and Wait by more than 3%, and significantly outperforming the Epidemic or Prophet v1 and v2 routing protocols. This reduction in hops is achieved through our meticulous node selection process and the principles we employ to determine packet forwarding. Regarding overhead, our algorithms adopt a more restricted approach to packet transmission compared to Spray and Wait, Epidemic, and Prophet v1 and v2 protocols, resulting in a substantial reduction in overhead (more than 32% less).

In summary, our study highlights the remarkable performance enhancements achieved by our proposed algorithms. The reductions in packet latency, number of hops, and overhead represent significant advancements over established protocols like Spray and Wait, Epidemic, Prophet v1, and Prophet v2. These improvements are attributed to our meticulous node selection process and refined packet forwarding principles. The results underscore the potential impact of our algorithms in optimizing QoS for routing in various network scenarios, emphasizing their significance in advancing network efficiency and reliability.

5. Conclusions and Future Work

The extraction of the most influential nodes in a Complex Network is crucial for seeking more efficient data transmission within the network, evaluating its resilience, making better routing decisions, and gaining a deeper understanding of its dynamics. Our study arises from the need to find metrics that measure the influence of nodes in a DSCN, aiming to enhance the network's QoS metrics.

Initially, we employed a network model that transformed the operation of OppMSN over time into a discretized time series of CSNs, to analyze the network's dynamic topology

and the pattern of connections among devices. This approach provides a more accurate framework to analyze the evolution of these patterns, based on regression analysis, rather than using a single static aggregated network. In fact, the connections between devices have been analyzed from the perspective of dynamic centrality metrics, as well as from the perspective of a social complex network, extracting relationship patterns to detect the most influential devices over others throughout the network's operation.

In this study, real datasets have been used for validation to showcase the effectiveness of the conducted experiments. The efficacy of different metrics employed on the datasets and potential correlations between them have been verified. Finally, based on influence dynamic rankings, our algorithms have facilitated better decision-making regarding the selection of nodes most suitable for routing data toward their destination in the datasets, leading to enhancements in standard QoS metrics.

Moving forward, our future work involves analyzing the evolutionary characteristics of influence distribution using additional real datasets with more devices and enhanced connectivity among them. Furthermore, we will explore other combinations of centrality metrics and similarity indices to enhance the accuracy of classifying devices in an importance ranking. Additionally, we aim to investigate the concept of "friend" as a measure of temporal locality between each pair of nodes, evaluating its relationship with the connection capacity of a node with others.

Author Contributions: Conceptualization, F.J.R.-P.; Methodology, F.J.R.-P.; Formal analysis, F.J.R.-P.; Investigation, M.A.L.-R.; Resources, M.A.L.-R.; Data curation, M.A.L.-R.; Writing—review & editing, M.A.L.-R. and F.J.R.-P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded in part by TED2021-131699B-I00 / AEI/10.13039/501100011033 / European Union NextGenerationEU/PRTR.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

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

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

Appendix A

We quantify the Dynamic Degree of the devices in the example presented in Figure 1 for clarity. The conclusions are drawn from Figure A1, where edge weights represent contact times, and $\alpha = 0.5$ is employed to give equal relative weight to the evolutionary trend of connections and the frequency of contact times.

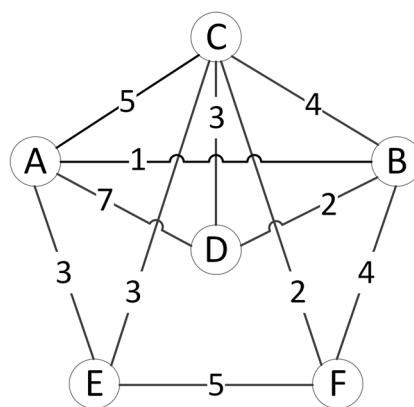


Figure A1. Aggregated network derived from contact frequency of Figure 1.

$$DTE_T(A) = 0.5 \cdot 0.0364 + (1 - 0.5) \frac{(5 + 1 + 7 + 3)}{10} = 0.0182 + 0.8 = 0.8182$$

$$DTE_T(B) = 0.5 \cdot (-0.0788) + (1 - 0.5) \frac{(4 + 1 + 2 + 4)}{10} = (-0.0394) + 0.55 = 0.5106$$

$$DTE_T(C) = 0.5 \cdot 0.1151 + (1 - 0.5) \frac{(5 + 3 + 3 + 2 + 4)}{10} = 0.05755 + 0.85 = 0.90755$$

$$DTE_T(D) = 0.5 \cdot (-0.0788) + (1 - 0.5) \frac{(7 + 3 + 2)}{10} = (-0.0394) + 0.6 = 0.5606$$

$$DTE_T(E) = 0.5 \cdot (-0.0182) + (1 - 0.5) \frac{(3 + 3 + 5)}{10} = (-0.0091) + 0.55 = 0.5409$$

$$DTE_T(F) = 0.5 \cdot (-0.0545) + (1 - 0.5) \frac{(5 + 2 + 4)}{10} = (-0.02725) + 0.55 = 0.52275$$

For instance, based on the following details, a preliminary ranking of nodes can be established: Device A interacts with its neighbors more frequently than device B, despite both having the same three neighbors. As a result, device A displays a higher Dynamic Degree than device B. Similar to the previous example, device E receives a better score due to its greater trend of contact development, even if device D contacts its neighbors more frequently. Device C achieves a higher Dynamic Degree than Device A due to having more neighbors and a faster rate of contact development. Consequently, the nodes are ranked as follows according to their Dynamic Degree index: $\text{Ranking}_{\text{DTE}} = \{C, A, D, E, F, B\}$.

Let us examine now an example calculation in Figure A2, showcasing an unconnected network during G_2 .

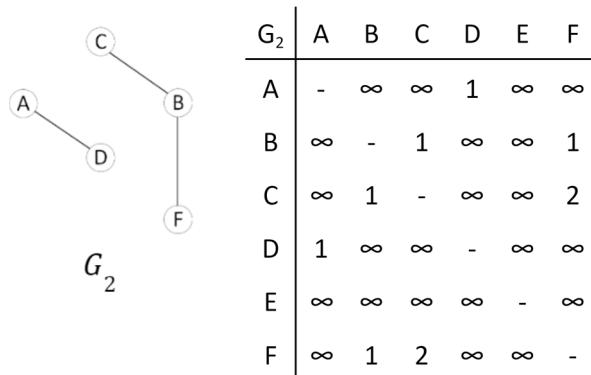


Figure A2. Network connections during instant G_2 , based on the example of Figure 1.

$$C_{CLO}(A) = \lim_{x \rightarrow \infty} \left(\frac{1}{6-1} \left(\frac{1}{x} + \frac{1}{x} + \frac{1}{2\lambda} + \frac{1}{x} + \frac{1}{x} \right) \right) = \frac{1}{10\lambda}$$

$$C_{CLO}(B) = \lim_{x \rightarrow \infty} \left(\frac{1}{6-1} \left(\frac{1}{x} + \frac{1}{2\lambda} + \frac{1}{x} + \frac{1}{x} + \frac{1}{2\lambda} \right) \right) = \frac{1}{5\lambda}$$

$$C_{CLO}(C) = \lim_{x \rightarrow \infty} \left(\frac{1}{6-1} \left(\frac{1}{x} + \frac{1}{2\lambda} + \frac{1}{x} + \frac{1}{x} + \frac{1}{3\lambda} \right) \right) = \frac{1}{6\lambda}$$

$$C_{CLO}(D) = \lim_{x \rightarrow \infty} \left(\frac{1}{6-1} \left(\frac{1}{2\lambda} + \frac{1}{x} + \frac{1}{x} + \frac{1}{x} + \frac{1}{x} \right) \right) = \frac{1}{10\lambda}$$

$$C_{CLO}(E) = \lim_{x \rightarrow \infty} \left(\frac{1}{6-1} \left(\frac{1}{x} + \frac{1}{x} + \frac{1}{x} + \frac{1}{x} + \frac{1}{x} \right) \right) = 0$$

$$C_{CLO}(F) = \lim_{x \rightarrow \infty} \left(\frac{1}{6-1} \left(\frac{1}{x} + \frac{1}{2\lambda} + \frac{1}{3\lambda} + \frac{1}{x} + \frac{1}{x} \right) \right) = \frac{1}{6\lambda}$$

Upon analyzing this small example, it becomes apparent that when computed on an unconnected network, the Closeness metric tends to result in lower values, indicating the difficulty of communication between devices belonging to different components. Additionally, the devices within the same component experience an increase in their centrality, as all values are non-zero in the calculation. Consequently, the metric places greater emphasis on nodes that are well-connected.

Let us see the same example of calculation as before for connected devices $\{A, B, C, D, F\}$, from G_7 as shown in Figure 4, but now using the new closeness equation (Equation (15)):

$$C_{CLO}(A) = \lim_{x \rightarrow \infty} \left(\frac{1}{6-1} \left(\frac{1}{3\lambda} + \frac{1}{4\lambda} + \frac{1}{2\lambda} + \frac{1}{x} + \frac{1}{4\lambda} \right) \right) = \frac{4}{15\lambda}$$

$$C_{CLO}(B) = \lim_{x \rightarrow \infty} \left(\frac{1}{6-1} \left(\frac{1}{3\lambda} + \frac{1}{2\lambda} + \frac{1}{2\lambda} + \frac{1}{x} + \frac{1}{2\lambda} \right) \right) = \frac{11}{30\lambda}$$

$$C_{CLO}(C) = \lim_{x \rightarrow \infty} \left(\frac{1}{6-1} \left(\frac{1}{4\lambda} + \frac{1}{2\lambda} + \frac{1}{3\lambda} + \frac{1}{x} + \frac{1}{2\lambda} \right) \right) = \frac{19}{60\lambda}$$

$$C_{CLO}(D) = \lim_{x \rightarrow \infty} \left(\frac{1}{6-1} \left(\frac{1}{2\lambda} + \frac{1}{2\lambda} + \frac{1}{3\lambda} + \frac{1}{x} + \frac{1}{3\lambda} \right) \right) = \frac{1}{3\lambda}$$

$$C_{CLO}(E) = \lim_{x \rightarrow \infty} \left(\frac{1}{6-1} \left(\frac{1}{x} + \frac{1}{x} + \frac{1}{x} + \frac{1}{x} + \frac{1}{x} \right) \right) = 0$$

$$C_{CLO}(F) = \lim_{x \rightarrow \infty} \left(\frac{1}{6-1} \left(\frac{1}{4\lambda} + \frac{1}{2\lambda} + \frac{1}{2\lambda} + \frac{1}{3\lambda} + \frac{1}{x} \right) \right) = \frac{19}{60\lambda}$$

The new ranking of devices based on their closeness index, which is determined by using the sum of reciprocal distances instead of the reciprocal sum of distances, is in line with the ranking obtained by using the traditional equation (Equation (11)) on a network that is well-connected (Equation (12)):

$$\text{Ranking} = \{B, D, C, F, A\}$$

To maintain clarity in the example illustrated in Figure 1, we opt for a value of 0.5 when assessing the closeness of the devices. This choice ensures that the Closeness measure and its dynamics carry equal relative weight.

$$CTE_T(A) = 0.5 \cdot 0.0174 + (1 - 0.5) \frac{2.0667(\lambda^{-1})}{10} = 0.1167\lambda^{-1}$$

$$CTE_T(B) = 0.5 \cdot (-0.0026) + (1 - 0.5) \frac{1.43344(\lambda^{-1})}{10} = 0.0704\lambda^{-1}$$

$$CTE_T(C) = 0.5 \cdot 0.0056 + (1 - 0.5) \frac{2.3501(\lambda^{-1})}{10} = 0.1203\lambda^{-1}$$

$$CTE_T(D) = 0.5 \cdot (-0.0122) + (1 - 0.5) \frac{1.9833(\lambda^{-1})}{10} = 0.0931\lambda^{-1}$$

$$CTE_T(E) = 0.5 \cdot (-0.0001) + (1 - 0.5) \frac{1.8167(\lambda^{-1})}{10} = 0.0908\lambda^{-1}$$

$$CTE_T(F) = 0.5 \cdot (-0.0065) + (1 - 0.5) \frac{1.7667(\lambda^{-1})}{10} = 0.0851\lambda^{-1}$$

The new ranking of nodes according to their Dynamic Closeness index is as follows:

$$\text{Ranking} = \{C, A, D, E, F\}$$

The best path from device A to device B in Figure 5 is then determined using the opportunistic-based shortest path method. Table A1 presents the findings of this investigation.

$$ORI_T(A, B) = \left(\sum_{i=1}^T \{\varphi_{AB} x_{AB}\}_{t_i} \right)^{\frac{|\Gamma(A) \cap \Gamma(B)|+1}{2}} = 1^{\frac{1+1}{2}}$$

$$ORI_T(A, C) = \left(\sum_{i=1}^T \{\varphi_{AC} x_{AC}\}_{t_i} \right)^{\frac{|\Gamma(A) \cap \Gamma(C)|+1}{2}} = 5^{\frac{1+1}{2}}$$

$$ORI_T(C, B) = \left(\sum_{i=1}^T \{\varphi_{CB} x_{CB}\}_{t_i} \right)^{\frac{|\Gamma(C) \cap \Gamma(B)|+1}{2}} = 4^{\frac{1+1}{2}}$$

$$ORI_T(A, D) = \left(\sum_{i=1}^T \{\varphi_{AD} x_{AD}\}_{t_i} \right)^{\frac{|\Gamma(A) \cap \Gamma(D)|+1}{2}} = 7^{\frac{1+1}{2}}$$

$$ORI_T(D, B) = \left(\sum_{i=1}^T \{\varphi_{DB} x_{DB}\}_{t_i} \right)^{\frac{|\Gamma(D) \cap \Gamma(B)|+1}{2}} = 2^{\frac{1+1}{2}}$$

$$ORI_T(A, E) = \left(\sum_{i=1}^T \{\varphi_{AE} x_{AE}\}_{t_i} \right)^{\frac{|\Gamma(A) \cap \Gamma(E)|+1}{2}} = 3^{1/3}$$

$$ORI_T(E, F) = \left(\sum_{i=1}^T \{\varphi_{EF} x_{EF}\}_{t_i} \right)^{\frac{|\Gamma(E) \cap \Gamma(F)|+1}{2}} = 5^{1/5}$$

$$ORI_T(F, B) = \left(\sum_{i=1}^T \{\varphi_{FB} x_{FB}\}_{t_i} \right)^{\frac{|\Gamma(F) \cap \Gamma(B)|+1}{2}} = 4^{1/4}$$

$$\alpha_{A,B} = \frac{(ORI_T(A, B))^{\frac{1}{|\Gamma(A, B)|}}}{\max(ORI_T(A, B))} = \frac{\left(1^{\frac{1+1}{2}}\right)^{\frac{1}{1}}}{{1^{\frac{1+1}{2}}}} = 1$$

$$\alpha_{A,C,B} = \frac{(ORI_T(A, C) \cdot ORI_T(C, B))^{\frac{1}{2}}}{\max(ORI_T(A, C), ORI_T(C, B))} = \frac{\left(5^{\frac{1+1}{2}} \cdot 4^{\frac{1+1}{2}}\right)^{\frac{1}{2}}}{5^{\frac{1+1}{2}}} = 0.8944$$

$$\alpha_{A,D,B} = \frac{(ORI_T(A, D) \cdot ORI_T(D, B))^{\frac{1}{2}}}{\max(ORI_T(A, D), ORI_T(D, B))} = \frac{\left(7^{\frac{1+1}{2}} \cdot 2^{\frac{1+1}{2}}\right)^{\frac{1}{2}}}{7^{\frac{1+1}{2}}} = 0.5345$$

$$\alpha_{A,E,F,B} = \frac{(ORI_T(A, E) \cdot ORI_T(E, F) \cdot ORI_T(F, B))^{\frac{1}{3}}}{\max(ORI_T(A, E), ORI_T(E, F), ORI_T(F, B))} = \frac{\left(3^{\frac{1}{3}} \cdot 5^{\frac{1}{5}} \cdot 4^{\frac{1}{4}}\right)^{\frac{1}{3}}}{3^{\frac{1}{3}}} = 0.9789$$

$$Cost_{opp}(A, B) = \left(\frac{1}{1^{\frac{1+1}{2}}} \right)^1 = 1$$

$$Cost_{opp}(A, C, B) = \left(\frac{1}{5^{\frac{1+1}{2}}} \right)^{0.8944} + \left(\frac{1}{4^{\frac{1+1}{2}}} \right)^{0.8944} = 0.5226$$

$$Cost_{opp}(A, D, B) = \left(\frac{1}{7^{\frac{1+1}{2}}} \right)^{0.5345} + \left(\frac{1}{2^{\frac{1+1}{2}}} \right)^{0.5345} = 1.0437$$

$$Cost_{opp}(A, E, F, B) = \left(\frac{1}{3^{\frac{1}{3}}} \right)^{0.9789} + \left(\frac{1}{5^{\frac{1}{5}}} \right)^{0.9789} + \left(\frac{1}{4^{\frac{1}{4}}} \right)^{0.9789} = 2.1408$$

Table A1. Results of the opportunistic social shortest path method to identify the best path from device A to device B .

Path	Latency	Load Balancing	Overhead	ORI_T	ORI_T Balancing	$Cost_{opp}$
$\{A, B\}$	$10t$	$0t$	2λ	$1^{2/2}$	0	1
$\{A, C, B\}$	$7t$	$0.5t$	3λ	$5^{2/2}, 4^{2/2}$	4.5	0.5226
$\{A, D, B\}$	$7t$	$1.t$	3λ	$7^{2/2}, 2^{2/2}$	4.5	1.0437
$\{A, E, F, B\}$	$7t$	$1.5764t$	4λ	$3^{1/3}, 5^{1/5}, 4^{1/4}$	1.5087	2.1408

The path $\{A, B\}$ represents the shortest distance, but it possesses the lowest $ORI_T(A, B)$ value among all routes. On the other hand, the path $\{A, E, F, B\}$ has the longest path distance with a $ORI_T(A, E, F, B)$ value of 4.2362 (the sum of the three ORI_T), which is the second lowest. Comparatively, both the paths $\{A, C, B\}$ and $\{A, D, B\}$ share the same path distance, with $ORI_T(A, C, B) = ORI_T(A, D, B) = 9$. However, $Cost_{opp}(A, C, B)$ is lower than $Cost_{opp}(A, D, B)$. As a consequence, when both social relationships (ORI_T) and communication costs are taken into consideration, the path $\{A, C, B\}$ is shown to be the best option for moving from device A to device B .

Therefore, to obtain the ranking based on CLO_{opp} , one must compute the CLO_{opp} of each node with respect to the rest:

$$\begin{aligned}
 CLO_{opp}(A) &= \frac{1}{6-1} \left(\frac{1}{\min(Cost_{opp}(A,B))} + \frac{1}{\min(Cost_{opp}(A,C))} + \frac{1}{\min(Cost_{opp}(A,D))} \right. \\
 &\quad \left. + \frac{1}{\min(Cost_{opp}(A,E))} + \frac{1}{\min(Cost_{opp}(A,F))} \right) \\
 &= \frac{1}{6-1} \left(\frac{1}{0.5264} + \frac{1}{0.2} + \frac{1}{0.1429} + \frac{1}{0.6934} + \frac{1}{1.4289} \right) = 3.5343 \\
 CLO_{opp}(B) &= \frac{1}{6-1} \left(\frac{1}{\min(Cost_{opp}(B,A))} + \frac{1}{\min(Cost_{opp}(B,C))} + \frac{1}{\min(Cost_{opp}(B,D))} \right. \\
 &\quad \left. + \frac{1}{\min(Cost_{opp}(B,E))} + \frac{1}{\min(Cost_{opp}(B,F))} \right) \\
 &= \frac{1}{6-1} \left(\frac{1}{0.5264} + \frac{1}{0.25} + \frac{1}{0.5} + \frac{1}{0.4378} + \frac{1}{0.7071} \right) = 2.0019 \\
 CLO_{opp}(C) &= \frac{1}{6-1} \left(\frac{1}{\min(Cost_{opp}(C,A))} + \frac{1}{\min(Cost_{opp}(C,B))} + \frac{1}{\min(Cost_{opp}(C,D))} \right. \\
 &\quad \left. + \frac{1}{\min(Cost_{opp}(C,E))} + \frac{1}{\min(Cost_{opp}(C,F))} \right) \\
 &= \frac{1}{6-1} \left(\frac{1}{0.2} + \frac{1}{0.25} + \frac{1}{0.4497} + \frac{1}{1.2428} + \frac{1}{1.2523} \right) = 2.5654 \\
 CLO_{opp}(D) &= \frac{1}{6-1} \left(\frac{1}{\min(Cost_{opp}(D,A))} + \frac{1}{\min(Cost_{opp}(D,B))} + \frac{1}{\min(Cost_{opp}(D,C))} \right. \\
 &\quad \left. + \frac{1}{\min(Cost_{opp}(D,E))} + \frac{1}{\min(Cost_{opp}(D,F))} \right) \\
 &= \frac{1}{6-1} \left(\frac{1}{0.1429} + \frac{1}{0.5} + \frac{1}{0.4497} + \frac{1}{1.2603} + \frac{1}{1.3055} \right) = 2.5562 \\
 CLO_{opp}(E) &= \frac{1}{6-1} \left(\frac{1}{\min(Cost_{opp}(E,A))} + \frac{1}{\min(Cost_{opp}(E,B))} + \frac{1}{\min(Cost_{opp}(E,C))} \right. \\
 &\quad \left. + \frac{1}{\min(Cost_{opp}(E,D))} + \frac{1}{\min(Cost_{opp}(E,F))} \right) \\
 &= \frac{1}{6-1} \left(\frac{1}{0.6394} + \frac{1}{1.4378} + \frac{1}{1.2428} + \frac{1}{1.2603} + \frac{1}{0.7248} \right) = 1.0474 \\
 CLO_{opp}(F) &= \frac{1}{6-1} \left(\frac{1}{\min(Cost_{opp}(F,A))} + \frac{1}{\min(Cost_{opp}(F,B))} + \frac{1}{\min(Cost_{opp}(F,C))} \right. \\
 &\quad \left. + \frac{1}{\min(Cost_{opp}(F,D))} + \frac{1}{\min(Cost_{opp}(F,E))} \right) \\
 &= \frac{1}{6-1} \left(\frac{1}{1.4289} + \frac{1}{0.7071} + \frac{1}{1.2523} + \frac{1}{1.3055} + \frac{1}{0.7248} \right) = 1.0116
 \end{aligned}$$

So, the new ordered ranking of nodes based on their Social Closeness index is as follows:

$$Ranking_{CLO_opp} = \{A, C, D, B, F, E\}$$

References

- Number of Smartphone Mobile Network Subscriptions Worldwide from 2016 to 2022, with Forecasts from 2023 to 2028. Available online: <https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/> (accessed on 30 June 2023).
- Zhang, H.; Chen, Z.; Wu, J.; Liu, K. FRRF: A Fuzzy Reasoning Routing-Forwarding Algorithm Using Mobile Device Similarity in Mobile Edge Computing-Based Opportunistic Mobile Social Networks. *IEEE Access* **2019**, *7*, 35874–35889. [CrossRef]
- Gantha, S.S.; Jaiswal, S.; Ppallan, J.M.; Arunachalam, K. Path Aware Transport Layer Solution for Mobile Networks. *IEEE Access* **2020**, *8*, 174605–174613. [CrossRef]
- Yuan, X.; Yao, H.; Wang, J.; Mai, T.; Guizani, M. Artificial Intelligence Empowered QoS-Oriented Network Association for Next-Generation Mobile Networks. *IEEE Trans. Cogn. Commun. Netw.* **2021**, *7*, 856–870. [CrossRef]
- He, B.; Wang, J.; Qi, Q.; Sun, H.; Liao, J. RTHop: Real-Time Hop-by-Hop Mobile Network Routing by Decentralized Learning with Semantic Attention. *IEEE Trans. Mob. Comput.* **2023**, *22*, 1731–1747. [CrossRef]
- Soelistijanto, B.; Howarth, M.P.; Qi, Q.; Sun, H.; Liao, J. Transfer Reliability and Congestion Control Strategies in Opportunistic Networks: A Survey. *IEEE Commun. Surv. Tutor.* **2014**, *16*, 538–555. [CrossRef]
- Xiong, F.; Xia, L.; Xie, J.; Sun, H.; Wang, H.; Li, A.; Yu, Y. Is Hop-by-Hop Always Better Than Store-Carry-Forward for UAV Network? *IEEE Access* **2019**, *7*, 154209–154223. [CrossRef]
- Anh Duong, D.V.; Kim, D.Y.; Yoon, S. TSIRP: A Temporal Social Interactions-Based Routing Protocol in Opportunistic Mobile Social Networks. *IEEE Access* **2021**, *9*, 72712–72729. [CrossRef]
- Hajarathaiah, K.; Enduri, M.K.; Dhuli, S.; Anamalamudi, S.; Cenkeramaddi, L.R. Generalization of Relative Change in a Centrality Measure to Identify Vital Nodes in Complex Networks. *IEEE Access* **2023**, *11*, 808–824. [CrossRef]
- Qiu, L.; Zhang, J.; Tian, X.; Zhang, S. Identifying Influential Nodes in Complex Networks Based on Neighborhood Entropy Centrality. *Comput. J.* **2021**, *64*, 1465–1476. [CrossRef]
- Ibrahim, M.H.; Missaoui, R.; Vaillancourt, J. Cross-Face Centrality: A New Measure for Identifying Key Nodes in Networks Based on Formal Concept Analysis. *IEEE Access* **2020**, *8*, 206901–206913. [CrossRef]
- Zhu, Y.; Ma, H. Ranking Hubs in Weighted Networks with Node Centrality and Statistics. In Proceedings of the Fifth International Conference on Instrumentation and Measurement, Computer, Communication and Control (IMCCC), Qinhuangdao, China, 18–20 September 2015.
- Liu, M.; Zeng, Y.; Jiang, Z.; Liu, Z.; Ma, J. Centrality Based Privacy Preserving for Weighted Social Networks. In Proceedings of the 13th International Conference on Computational Intelligence and Security (CIS), Hong Kong, China, 15–18 December 2017.
- Niu, J.; Fan, J.; Wang, L.; Stojmenovic, M. K-hop centrality metric for identifying influential spreaders in dynamic large-scale social networks. In Proceedings of the IEEE Global Communications Conference, Austin, TX, USA, 8–12 December 2014.
- Rogers, T. Null models for dynamic centrality in temporal networks. *J. Complex Netw.* **2015**, *3*, 113–125. [CrossRef]
- Opsahl, T.; Agneessens, F.; Skvoretz, J. Node centrality in weighted networks: Generalizing degree and shortest paths. *Soc. Netw.* **2010**, *32*, 245–251. [CrossRef]
- Kim, H.; Anderson, R. Temporal node centrality in complex networks. *Phys. Rev. E* **2012**, *85*, 026107. [CrossRef] [PubMed]
- Trajkovic, L. Complex Networks. In Proceedings of the IEEE 19th International Conference on Cognitive Informatics & Cognitive Computing (ICCI*CC), Beijing, China, 26–28 September 2020.
- Zhang, S.-S.; Liang, X.; Wei, Y.-D.; Zhang, X. On Structural Features, User Social Behavior, and Kinship Discrimination in Communication Social Networks. *IEEE Trans. Comput. Soc. Syst.* **2020**, *7*, 425–436. [CrossRef]
- Elmezain, M.; Othman, E.A.; Ibrahim, H.M. Temporal Degree-Degree and Closeness-Closeness: A New Centrality Metrics for Social Network Analysis. *Mathematics* **2021**, *9*, 2850. [CrossRef]
- Wang, Y.; Li, H.; Zhang, L.; Zhao, L.; Li, W. Identifying influential nodes in social networks: Centripetal centrality and seed exclusion approach. *Chaos Solitons Fractals* **2022**, *162*, 112513. [CrossRef]
- Caschera, M.C.; D’Ulizia, A.; Ferri, F.; Grifoni, P. MONDE: A method for predicting social network dynamics and evolution. *Evol. Syst.* **2019**, *10*, 363–379. [CrossRef]
- Ugurlu, O. Comparative analysis of centrality measures for identifying critical nodes in complex networks. *J. Comput. Sci.* **2022**, *62*, 101738. [CrossRef]
- Dey, P.; Bhattacharya, S.; Roy, S. A Survey on the Role of Centrality as Seed Nodes for Information Propagation in Large Scale Network. *ACM/IMS Trans. Data Sci.* **2021**, *2*, 1–25. [CrossRef]
- Omar, Y.M.; Plapper, P. A Survey of Information Entropy Metrics for Complex Networks. *Entropy* **2020**, *22*, 1417. [CrossRef]
- Gao, Z.; Shi, Y.; Chen, S. Measures of node centrality in mobile social networks. *Int. J. Mod. Phys. C* **2015**, *26*, 1550107. [CrossRef]
- Wei, B.; Kawakami, W.; Kanai, K.; Katto, J. A History-Based TCP Throughput Prediction Incorporating Communication Quality Features by Support Vector Regression for Mobile Network. In Proceedings of the 13th International Conference on Information and Communication Technology Convergence (ICTC), Taichung, Taiwan, 11–13 December 2017.
- Awane, H.; Ito, Y.; Koizumi, M. Study on QoS Estimation of In-vehicle Ethernet with CBS by Multiple Regression Analysis. In Proceedings of the IEEE International Symposium on Multimedia (ISM), Jeju Island, Republic of Korea, 19–21 October 2022.
- Chen, W.; Zhou, Y. A Link Prediction Similarity Index Based on Enhanced Local Path Method. In Proceedings of the 40th Chinese Control Conference (CCC), Shanghai, China, 26–28 July 2021.
- Varma, S.; Shivam, S.; Thumu, A.; Bhushanam, A.; Sarkar, D. Jaccard Based Similarity Index in Graphs: A Multi-Hop Approach. In Proceedings of the IEEE Delhi Section Conference (DELCON), New Delhi, India, 11–13 February 2022.

31. Ahmad, I.; Akhtar, M.; Noor, S.; Shahnaz, A. Missing Link Prediction using Common Neighbor and Centrality based Parameterized Algorithm. *Nat. Sci. Rep.* **2020**, *10*, 364. [[CrossRef](#)] [[PubMed](#)]
32. Eagle, N.; Pentland, A. Reality mining: Sensing complex social systems. *Pers. Ubiquitous Comput.* **2006**, *10*, 255–268. [[CrossRef](#)]
33. Orlinski, M.; Filer, N. The rise and fall of spatio-temporal clusters in mobile ad hoc networks. *Ad Hoc Netw.* **2013**, *11*, 1641–1654. [[CrossRef](#)]
34. Anulakshmi, S.; Anand, S.; Ramesh, M.V. Impact of Network Density on the Performance of Delay Tolerant Protocols in Heterogeneous Vehicular Network. In Proceedings of the International Conference on Wireless Communications Signal Processing and Networking (WiSPNET), Chennai, India, 21–23 March 2019.
35. Sati, M.; Shanab, S.; Elshawesh, A.; Sati, S.O. Density and Degree Impact on Opportunistic Network Communications. In Proceedings of the IEEE 1st International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering MI-STA, Tripoli, Libya, 25–27 May 2021.
36. Dede, J.; Förster, A.; Hernández-Orallo, E.; Herrera-Tapia, J.; Kuladinithi, K.; Kuppusamy, V.; Vatandas, Z. Simulating opportunistic networks: Survey and future directions. *IEEE Commun. Surv. Tutor.* **2017**, *20*, 1547–1573. [[CrossRef](#)]
37. Khan, M.; Liu, M.; Dou, W.; Yu, S. vGraph: Graph Virtualization towards Big Data. In Proceedings of the Third IEEE International Conference on Advanced Cloud and Big Data (CBD), Yangzhou, China, 30 October–1 November 2015.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.