 ISSN NO: 2770-4491
December 2024

Designing And Optimizing Opportunistic Communication Networks

Jolimbetova Elyanora
PhD student of the Department of "Data Communication Networks and Systems"
Tashkent University of Information Technologies named after Muhammad al-Khwarizmi
elyanorajolimbetova@gmail.com

Abstract

This paper tries to address the challenge of detecting and managing overlay topologies in opportunistic networks. As a first contribution, it introduces the concept of "link" in opportunistic networks, which is different from the traditional perspective in classic MANETs: instead of an instantaneous communication relation between two nodes, links are defined here as the cumulative contacts of nodes over a time interval. This redefinition lets opportunistic networks be modeled as graphs that evolve.

A key insight is that this approach enables the regulation of how nodes handle contacts to construct overlays according to a desired topology. That is important because the statistical properties of the overlay topology can form a basis for assessing a network's capacity to disseminate content. The feasibility of the proposed method is shown by a trace-driven simulation using various datasets, which range from dense urban networks to highly sparse environments. These experiments confirm that the approach effectively manages overlay topologies in opportunistic networks.

Keywords: MANET, DTN, QoS, OMN,OppNets, routing, neighborhood.

Introduction

Modern routing approaches in ad hoc networks assume that there is always a continuous path from the source node to the destination node [1]. However, this assumption is hardly correct because it has been disproved by common connection protocols such as ZigBee and Bluetooth connections [2]. The reason for this discrepancy is the extensive physical range of mobile networks and a high degree of node mobility caused by the random movement of these mobile networks. In sparse networks [3], node mobility is constant over time. Driven by the increasing demand of human-centric applications, such networks are classified as Opportunistic Networks (OppNets) [4] or Delay-Tolerant Networks (DTNs) [5], as in Fig. 1.

In these networks, the unpredictable mobility of users and network fragmentation prevent immediate connection establishment between source and destination nodes using conventional MANET routing protocols such as AODV [6] and DSR [7]. Specifically, reactive routing protocols [8] fail in identifying a viable path due to network dynamics, while proactive routing protocols [9] fail to adapt to the network's random topology. Fortunately, the lack of continuous connectivity does not necessarily imply packet delivery failure. With ever-changing network topology, opportunities for temporary connections may occur. Packets can be forwarded over the available paths until a connection break occurs, at which time packets are buffered awaiting establishment of the next link. Thus, this store-and-forward philosophy allows packets to eventually reach their destination. In practical applications, like short message services, the communication links should allow the messages to be delivered with least delays.

In contrast, our work extends these studies by providing a comprehensive analysis of routing algorithms under various buffer sizes, supplemented by experimental evaluations to demonstrate their performance.

A framework for modeling opportunistic networks.

An opportunistic network can be represented as a dynamic graph where links between nodes correspond to the cumulative contacts they have within a specific time interval *D*. At a given time *t*, the neighborhood of a node *i*, denoted as $\Pi_i(t)$, includes all nodes directly connected to *i*. Over time, this

ISSN NO: 2770-4491
December 2024

neighborhood evolves as nodes move and interact. The *D*-neighborhood, $\Pi_i^D(t)$, models all nodes with which *i* has been in contact during the interval $[t - D, t]$.

This *D*-neighborhood captures the essence of delay tolerance in opportunistic networks, establishing a relationship between two nodes once both have been in contact over the last *D* time window. Links where data can be forwarded actively, considering such links are required for communication, allow the sending of messages between nodes who may be in contact in this *D* - However, such links turn out to be temporary because mobile nodes have limited computational resources.

In this context, links are kept at the bundle or application layer. A link from node *i* to node *j* means that *i* keeps some information about *j* , such as its ID, profile and shared data. If *i* later encounters another node *k*, it may forward data of *j* to *k*, but such communication is inherently asymmetric unless *j* and *i* meet again.

Due to resource limitations, nodes keep only a subset of their *D* -neighborhood as active neighbors, and the nodes in this set are dynamically replaced over time. Links are thus directed, in that one node may keep information about another node without vice-versa. For convenience, a link is usually defined to exist if a single contact of sufficient duration to exchange data occurs, although other definitions, such as sustained communication over *D*, may also be used.

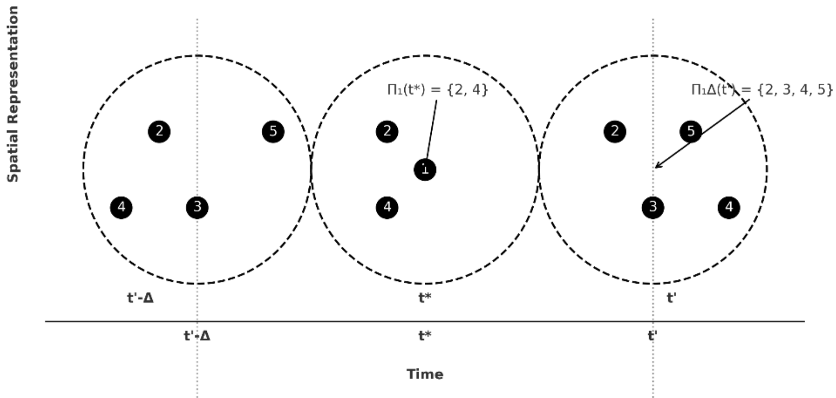

Fig. 1. Δ -neighborhood of node 1.

A Protocol for Structuring Opportunistic Networks

Given a model of how nodes interact, one can develop protocols that go beyond the nearly ubiquitous assumption that opportunistic networks are fundamentally unpredictable; rather, one can shape a network by specifying a desired topology, in a way that will be explained based on the definition of a "link". Concretely, the intended topology of an opportunistic network is specified as the probability distribution of the degree of connectivity a node may attain. This statistical representation provides a means to automatically estimate important network metrics, including network diameter-the longest shortest path between any two nodes-the average number of neighbors within a certain distance from a node, among others [3].

To achieve a specified network topology and desired node degrees, an interaction algorithm is executed locally at each node. This study proposes a straightforward algorithm that builds upon the conventional "store-carry-forward" paradigm typically used in opportunistic networks. In this approach, the size of the node's cache (representing its neighbor table) is configured to match the desired degree of connectivity. In other words, when a node is assigned a desired degree *d*, it creates a data structure that can store data from *d* neighbors in its *D* -neighborhood. Thus, the active links a node has at any point in time are a subset of its *D* neighborhood.

The link management is straightforward and works as follows (see Algorithm1). Each node chooses a target degree at the beginning of the network operations, according to a probability distribution of the desired topology. This degree is computed locally. Depending on the topology and the method used to generate the degree distribution, the node may need an estimate of the network size in order to compute its degree. For example, for random graphs, a node usually connects to about $-pN$ other nodes, where p is the connection probability and N is the network size. Existing approaches enable nodes to estimate the size of the network [7]; when the network is very large, such estimations may be rough. For example, in many large random graphs, node degrees are effectively modeled by a Poisson distribution. Nodes are also in charge of detecting any drastic changes in network size and updating their target degree if necessary.

```
Algorithm 1: Interaction Protocol for Node i

Require: Interaction with node j
lastContact[j] ← getCurrentTime()
if isActiveContact(j) then
    newData ← exchangeContents(j)
    addToCache(newData, j)
end if
Require: Timeout Δ of no interactions with node j
removeFromCache(j)

function addToCache(data, node):
if isNeighbor(node) then
    updateContents(node)
else
    if degree < desiredDegree then
        push(data, node)
        degree ← degree + 1
    else
        if random() < ω then
            k ← randomNeighborToReplace()
            deleteFromCache(k)
            push(data, node)
        end if
    end if
 end if
end function
```

Network Evolution and Link Formation

Nodes in opportunistic networks form links opportunistically according to their active contacts, also known as their **D-neighborhood**. Each node maintains the list of encountered nodes and continues building its links until its desired degree is achieved. For example, every time two nodes, say i and j, meet, i updates the contact information first and then assesses the value of the interaction by calling a function like **ACTIVECONTACT()**. This decision process will differ depending on the specific application running over the network.

The most frequently adopted is the **greedy strategy**: the node interacts with any peer that comes into contact. In fact, this is suitable for networks that are sparse, thus characterized by very rare contacts. However, under dense settings, like urban scenarios, other strategies could prefer some interactions over others based on the application requirements. For simplicity, this chapter considers a greedy approach. Having interacted, the nodes exchange data, which i may cache if its cache has space. If i is already at its desired degree, new links can replace old ones with some probability, often targeting the ones that have fewer recent contacts.

Link Duration and Optimization

Links between nodes are temporary, expiring after a period (D) of no contact. Tuning the duration (D) is critical: longer durations allow nodes to maintain more stable connections and retain neighbor data for

ISSN NO: 2770-4491
December 2024

extended periods, but excessively high values may result in outdated or irrelevant information. Proper optimization ensures a balance, allowing nodes to maintain their desired degree while fostering efficient data exchange. Section 5 of the original paper focuses on the analysis of changing D's influence on network performance, node degree stability included.

Active Contact Selection and Neighbor Strategies

Neighbor selection is one of the most crucial factors that influence network efficiency. The traditional approach has been a greedy one: a node will exchange data with every peer encountered. This strategy works for sparse networks but may not meet the more stringent requirements of some applications [10]. For instance,

- **Urban Networks**: People often interact based on mobility patterns, encountering strangers by chance or familiar individuals like friends and colleagues more frequently. Applications such as commuter-based file sharing or news exchange could favor interactions among frequently meeting nodes, ensuring consistent content updates.
- **Adaptive Link Creation**: In certain contexts, nodes may prefer rare contacts for the sake of content diversity. In this case, links are established with peers met less frequently, increasing the chances of exchanging novel data rather than redundant content.

By adapting neighbor selection strategies, opportunistic networks can be optimized for particular applications, such as social networking or location-based services.

Experimental Setup and Evaluation

For the purpose of evaluating the proposed interaction algorithm, a trace-driven simulator was implemented based on several real-world datasets. This provides a realistic insight into the behavior of the algorithm under diverse settings, avoiding the limitations of synthetic simulations[9]. The datasets include:

- **NUS Dataset**: Collected from students at the National University of Singapore, it represents a large, urban scenario with periodic, structured interactions, such as those during lectures.
- **UniMi Dataset**: Fine-grained, high-frequency interactions among students, faculty, and staff internal to the University of Milano represent dense, localized networks.
- **DieselNet Dataset**: Sparse, low-frequency interactions among buses over a large area simulate delaytolerant networks. Each one of these datasets contributes differently to both temporal and spatial natures and thus helps to evaluate the effectiveness of the algorithm under various conditions.

This work has shown that opportunistic networks, based on modifications of link duration (D) and neighboring selection strategies, can provide designed performance in various scenarios. Applications vary from urban file sharing to sparse delaytolerable networks; all exploit the adaptive strategy in pursuit of a good trade-off between stability and diversity when creating links.

Conclusion

The opportunistic networks' dynamic nature and the ability of tolerating delays mean that the modeling links cannot be treated as an instantaneous active connection between two nodes, but instead, they have to represent cumulative contacts happening within a certain time interval. In this context, a link in the evolving graph indicates that a node maintains application-related data from its neighbor in its "neighbor table" and thus forwards this data when an opportunity arises. The contact between two mobile nodes needs to be sufficiently long to permit data exchange, and the actual duration required depends on the application involved.

The work proposed a simple algorithm, underpinned by the store-carry-forward paradigm commonly utilized in opportunistic networks. This will be the algorithm that will then enable each node to manage its degree, shaping the network into the topology desired, and therefore impacting the way content will be disseminated across the network. The approach is evaluated with three kinds of datasets: a large and crowded opportunistic network, a small network with frequent interactions between nodes, and a sparse network with rarely interacting nodes.

Based on this interaction protocol, a trace-based simulator has been developed, enabling the running of simulations on these datasets. These results clearly showed how tweaking of the time interval *D* can shape the network to meet any arbitrary topology. This fact is important because the statistical properties of a given topology yield critical insight into the performance of content dissemination across a network.

REFERENCES:

1. Fall, K. (2003). A Delay-Tolerant Network Architecture for Challenged Internets. SIGCOMM '03: Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, 27–34. [DOI: 10.1145/863955.863960]

2. Kempe, D., Kleinberg, J., & Kumar, A. (2000). Connectivity and Inference Problems for Temporal Networks. STOC '00: Proceedings of the 32nd ACM Symposium on Theory of Computing, 513–522. [DOI: 10.1145/335305.335347]

3. Burgess, J., Gallagher, B., Jensen, D., & Levine, B. N. (2006). MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks. Proceedings of IEEE INFOCOM, 11(4), 1–11. [DOI: 10.1109/INFOCOM.2006.228]

4. Perkins, C. E., Royer, E. M., & Das, S. R. (2003). Ad Hoc On-Demand Distance Vector (AODV) Routing. RFC Editor. [https://doi.org/10.17487/rfc3561]

5. Bluetooth SIG. (2009). Bluetooth Core Specification Version 4.0. [https://www.bluetooth.com/specifications/specs/core-specification/]

6. Hossmann, T., Spyropoulos, T., & Legendre, F. (2011). From Contacts to Graphs: Pitfalls in Using Complex Network Analysis for DTN Routing. IEEE INFOCOM, 858–866. [DOI: 10.1109/INFCOM.2011.5934898]

7. Vahdat, A., & Becker, D. (2000). Epidemic Routing for Partially-Connected Ad Hoc Networks. Technical Report CS-2000-06, Duke University. [https://www.cs.duke.edu/ari/courses/fall06/cps296.3/papers/2000-vahdat-epidemic-routing.pdf]

8. Scellato, S., Mascolo, C., Musolesi, M., & Latora, V. (2011). Distance Matters: Geo-social Metrics for Online Social Networks. Proceedings of the 3rd Conference on Online Social Networks (WOSN), 8–14. [DOI: 10.1145/1991023.1991027]

9. Liang, Y., & Lee, J. (2006). Spatiotemporal Dataset Analysis for Wireless Mobile Networks. National University of Singapore. [https://nuswirelessmobilitydatasets.com]

10. Newman, M. E. J. (2003). The Structure and Function of Complex Networks. SIAM Review, 45(2), 167–256. [DOI: 10.1137/S003614450342480]