Energy Saving on DTN using Trajectory Inference Model

Antônio Rodrigo D. De Vit
PUCRS
Porto Alegre – RS
Brazil
antonio.vit@acad.pucrs.br

César Marcon
PUCRS
Porto Alegre – RS
Brazil
cesar.marcon@pucrs.br

Raul Ceretta Nunes
UFSM
Santa Maria - RS
Brazil
ceretta@inf.ufsm.br

Thais Webber
UNISC
Santa Cruz do Sul - RS
Brazil
thaiscs@unisc.br

Gustavo Sanchez
PUCRS
Porto Alegre – RS
Brazil
gustavo.sanchez @acad.pucrs.br

Carlos Oberdan Rolim
UFRGS
Porto Alegre – RS
Brazil
carlos.oberdan@inf.ufrgs.br

ABSTRACT

Delay or Disruption Tolerant Networks (DTN) are characterized by long delays and intermittent connectivity, requiring efficient energy consumption for increasing the mobile nodes lifetime. The movements of nodes modify the network topology, changing the number of connection opportunities between nodes. This paper proposes a new technique for energy saving on DTN by using a trajectory inference model for mobile nodes powered by machine learning techniques. The objective of this work is to reduce the energy consumption of DTN using a mobility prediction method. Experimental results indicate more than 47% of energy saving on data communication applying the trajectory inference model.

CCS CONCEPTS

• **General and reference** → General literature • **Networks** → Mobile ad hoc networks • **Hardware** → Power and energy • **Hardware** → Power estimation and optimization.

KEYWORDS

Delay or Disruption Tolerant Network, Opportunistic Networking, Energy Saving, Trajectory Inference Model

ACM Reference format:

A. Vit, C. Marcon, R. Nunes, T. Webber, G. Sanchez, C. Rolim, 2018. In *Proceedings of ACM SAC Conference, Pau, France, April 9-13, 2018 (SAC'18)*, 4 pages. DOI: https://doi.org/10.1145/3167132.3167437

1 INTRODUCTION

Delay or Disruption Tolerant Networks (DTN) are subject to long delays, intermittent connectivity as well as energy restrictions, limited bandwidth, and high error rates. There are different terminologies for DTN in the literature, like opportunistic networking. The DTN is also one of the most interesting evolutions of MANET (Mobile Ad hoc NETworks). Several applications of DTN include sensor networks for monitoring ecological environments, vehicular networks, ad hoc military networks, among others.

The movement of nodes alters network topology, increasing the number of connection opportunities among nodes. One fundamental problem in this context is how to route messages from source to destination with high delivery rate, low latency, low overhead and low energy usage since end-to-end paths might be absent for the whole lifetime of the message. We argue that the nodes' mobility could be used to infer future trajectory patterns resulting in significant energy savings for DTN.

This work is organized as follows. Section 2 presents the related work. Section 3 comprises the proposed approach with detailed description of the trajectory inference model. Section 4 details the experimental setup and experimental results. Finally, conclusions are summarized in Section 5.

2 RELATED WORK

Our work demonstrates that energy saving approaches applied to DTN regards two essential aspects: (i) the protocol used to route messages through the network nodes, and (ii) the knowledge of nodes' position for efficient message delivery. This section shows how our work relates to other scientific approaches in view of these two aspects.

2.1. Message Routing Protocols

The routing protocols used in DTN have been based on the node's mobility awareness and store-carry-forward mechanisms [1].

The Epidemic protocol [2] operates similarly to the spread of an infectious disease. Nodes transmit the message blindly (i.e. disregarding the content of the message) to all other nodes that do not yet have a copy of this message until the message reaches its destination. The purpose of having an unlimited amount of copies is to increase the message delivery rate using each possible path, even with a high resource consumption and overhead.

Spray and Wait [3] is an epidemic protocol with transmission of a limited number of copies of each message, consisting of two phases. In the Spray phase, the nodes send N copies to N neighbors, which transport these messages during the Wait phase until they arrive at their destination. In the second version of this protocol, called Binary Spray and Wait, each node sends half the copies of a message to any node found along the path. When the number of copies becomes equal to one, the node waits for the opportunity to deliver the message directly to the receiver.

The PRoPHET protocol [4] uses data of encounters among nodes that exchange state vectors for increasing the predictability of message deliveries. These state vectors contain information about other nodes that have already been encountered. The authors propose using probabilistic routing assuming non-random node's mobility to improve the delivery rate of messages while keeping buffer usage and communication overhead at a low level.

Since people or vehicles can transport a node, it is important to manage the energy costs for sending/receiving messages and probe connections from other nodes within transmission range. Our contribution is to demonstrate that exploring mobility information in DTN is a promising way to save energy. The probing connections close to the nodes' transmission range is one factor of significant energy consumption [5-6]. A node that can predict a mobility pattern in the DTN can infer current and future trajectories, enabling to explore techniques to reduce the energy consumption. Besides, different routing algorithms can implement our proposal without changing its operating characteristics. These details are discussed in the following sections.

2.2. Node's Trajectory Inference Principles

Several criteria must be taken into consideration to understand the principles for inference of mobile nodes trajectory. The models Manhattan mobility [7], Random Waypoint [8] and Random Walk [9] introduce a *memoryless* pattern, where the next move is entirely independent of the previous one. Models with *memoryless* pattern have no dependence between current and future movements; thus, they are not suitable to model DTN mobility. In addition, some of them have exclusively random movements, making it difficult to use analytical modeling to predict future positions. Various authors choose traces based on the mobility of humans, animals, and vehicles to model real scenarios, using varied techniques to predict nodes' movement, such as those that use machine learning, neural networks, genetic algorithms and classification techniques [10-13]. Our proposed work is based on traces from the real-world scenario, collected from buses fleets. We based our tests on a real instance of a specific mobility dataset, and we use it as input to the trajectory prediction method with real-world buses trace from CRAWDAD dataset [14].

3 PROPOSED APPROACH

This work proposes the blend of nodes trajectory prediction, and the knowledge of the node encounters to control the activation and deactivation of communication mechanisms to save network energy. Our trajectory prediction method uses a data mining technique to build decision trees. They are capable of classifying nodes trajectories and predict nodes positions. We highlight two main aspects: (i) how to handle with energy consumption for a node in a DTN, and (ii) how to build the trajectory inference model. This section shows how these two aspects are built.

3.1. Energy Saving by Node Trajectory Inference

For each node in the DTN, from an initial **Scanning** state, two other states can be reached. If the node found another one in its range of connectivity, it goes to the **Communicating** state, remaining there while having data to transmit and/or receive. When the communication finishes, the node goes to a low energy consumption state (i.e., **Sleeping**), remaining there until it reaches a timeout defined by the *sleep time*. Then, the node restarts scanning for neighboring nodes during a predefined period (i.e., *scan time*). If the scanning procedure fails during the *scan time*, the node goes to sleep again to reduce its energy consumption.

Without the knowledge of all nodes placement, a tradeoff appears – with a significant *sleep time*, the sleeping node may lose several opportunities to find other nodes. Nevertheless, with a small *sleep time*, the node stays much time spending an enormous amount of energy scanning for communications, which is a typical characteristic of node behavior in DTN; i.e., a node spends more time disconnected and searching for new connections, then connected and transferring information. In fact, to guarantee connections with all meeting opportunities, the **Sleeping** state has to be avoided, implying substantial energy consumption. Therefore, we propose a new way to save energy without losing communication opportunities based on the inference of nodes trajectory in DTN.

The proposal inserts the **Inferring** state before going to the **Sleeping** state that is responsible for estimating future connections based on nodes trajectory. Once the previous communication finishes, the **Inferring** state allows the node to estimate if the next possible connection is timely distant or near. In the case of a near connection, the node restarts scanning for this near connection; otherwise, the connectivity-inferring algorithm set a lesser *sleep time* but near to the time inferred for the next meeting and reduces the energy consumption going to the **Sleeping** state.

The *sleep time* knowledge is the main reason for this new approach reduce energy without losing connections. While the classical approach does not have mechanisms to estimate the *sleep time*, this approach can estimate a reasonable where the communication circuits may remain turned off. The success of the proposed approach depends on the quality of the connectivity-inferring algorithm that depends on several aspects like quantity of nodes, movement patterns, and time spent learning the movement of the nodes. Therefore, the success of our approach can be measured with three metrics: (i) loss of meetings, (ii) quantity of unsuccessful scanning and (iii) energy consumption. A loss of meeting or an unsuccessful scanning happens when the connectivity-inferring algorithm sets a greater or lesser *sleep time* than the required, respectively.

3.1. Trajectory Inference Model

Let $\prod = \{\boldsymbol{p_1}, \ldots, \boldsymbol{p_N} \mid 1 < \boldsymbol{N}\}$ be an asynchronous and distributed system model consisting of a finite set of $\boldsymbol{N}$ mobile nodes. There is no global clock, and the nodes communicate through message exchange using radio broadcast. The hardware architecture of a mobile node has the following components: (i) processing unit; (ii) memory; (iii) input/output devices, responsible for data traffic; (iv) radio for sending/receiving messages; (v) sensors and actuators for communicating with the external environment. The software architecture considers the protocol stack of a DTN reference model, consisting of a set of protocols, which can be seen as a layered model. Each layer is responsible for a group of tasks, providing a well-defined set of services for the upper layer protocol [15]. The bundle layer, implemented by the bundle protocol, handles intermittent connectivity (storing a message and carrying this message until a contact takes place).

Our communication model regards radio messages with a limited transmission range and no signal loss. A message $\boldsymbol{m}$ sent by a node $\boldsymbol{p_i}$ is only received by a node $\boldsymbol{p_k}$ that is within its transmission range when $\boldsymbol{p_k}$ can identify the signal power of $\boldsymbol{m}$.

The Trajectory Inference Computing (TIC) requires hardware and software resources. A Positioning System (PS) module allows the node to obtain and notify information about its mobility. The data from this module are composed of 2D (horizontal and vertical) positions with time tags. These data are conveyed to the TIC in the software module. Figure 1 shows the TIC algorithm that receives the data correspondent to the nodes coordinates with their respective time tag (**readMobilityTrace**). Then, the algorithm calculates the connections between the nodes and uses this information to control the operations to optimize the node's energy management.

Each instance of TIC, placed in each node of the DTN, executes two tasks: (i) **makingConnectivityMap** (**Task0**), which is a static task performed at the beginning of the simulation, and (ii) **inferringStateOperation** (**Task1**), which is a dynamic task performed during the simulation.

Task0 produces the connectivity map of a node. The task starts reading the node mobility trace (line 1). Then, the algorithm creates a map (**connectivityMap**) containing all time intervals a node is in a connectivity range of neighbor nodes (lines 2-10). This procedure uses the 2D positioning of all nodes, for each simulation step, and signal strength. To build this task, we used data mining techniques for discovering patterns in the massive amounts of data; and we used these patterns for predicting the values of dependent variables by identifying regularities and building generalizations in the dataset attributes [16].

To construct our model and training set, we use the following information: X and Y axes coordinates; length, calculated from the sum of the distance formed between all pairs of consecutive trajectories points $\boldsymbol{p_n}$ and $\boldsymbol{p_{n-1}}$; X and Y axes lower values; X and Y axes greater values; X and Y axes difference; travel duration time; displacement, calculated from the distance between the first and last trajectory points; average speed between all trajectory pairs points; average acceleration between two speeds, calculated for all trajectory points; mean slope between all points; and medium change direction between all points. Additionally, all trajectories were divided into trajectory segments multiple of 100 seconds (i.e., 100, 200, 300, etc.), up to the total value of the trajectory itself.

Task0: makingConnectivityMap of node$_i$ ∈ ∏

```
1.  readMobilityTraces(nodei)
2.  for all 1 ≤ k ≤ N | nodek ∈ ∏ {
3.      if k ≠ i {
4.          readMobilityTraces(nodek)
5.          for each simulation time {
6.              if nodei(X,Y) in connectivity range of nodek(X,Y)
7.                  connectivityMapi(time)
8.          }
9.      }
10. }
```

Task1: inferringStateOperation of node$_i$ ∈ ∏

```
1.  at each transition to the Inferring state {
2.      timeToNextConnection ← connectivityMapi(time)
3.      if timeToNextConnection < predefinedThresholdTime {
4.          timeout ← predefinedScanningTime
5.          go to Scanning state
6.      }
7.      else {
8.          timeout ← timeToNextConnection – predefinedScanningTime
9.          reduce energy consumption of nodei
10.         go to Sleeping state
11. }
```

Figure 1: TIC algorithm for a *node$_i$*

After training, we obtained the decision trees that are models built through data mining, frequently used when high accuracy and low complexity execution are required [17]. The Waikato Environment for Knowledge Analysis (WEKA) version 3.8 [18], was used to train the decision trees. The training was performed using the J48 algorithm, which is an open-source implementation of the C4.5 algorithm available on WEKA. From the use of decision trees, **Task0** is implemented.

Task1 is performed each time a node goes to the **Inferring** state. The algorithm computes the time inferred for the meeting, which is stored in the variable **timeToNextConnection** (line 2), using the **connectivityMap** and simulation time. The algorithm compares this value with a predefined threshold (**predefinedThresholdTime**), whose definition depends on the quality of the prediction method (lines 3-11). Hence, for small **timeToNextConnection** a value, the node goes to the **Scanning** state to search for a new connection; else the node goes to the **Sleeping** state allowing saving energy.

4 RESULTS

ONE simulator [19] is used to evaluate the impact of saving energy through the inference of nodes trajectory in a DTN. The experiments encompass a dataset originated from the bus fleet movement of the city of Seattle (Washington, USA) [14] to simulate the nodes mobility. The raw data refers to date and time, bus and route identifications, and the XY coordinates of the buses. Following the instructions in [19], to use the simulator the date and time were converted to seconds and time events sort all lines. Finally, the sampling interval (time difference between two-time events) was one second for the entire file, which was obtained by interpolating the original values of the dataset.

This work uses the energy model consumption developed by [19]. The energy values used by [20] were adapted for this case. The simulation area, number of nodes and simulation time are based on real data extracted from [14]. The range values and transmission speed are compatible with IEEE 802.11 interfaces for outdoor settings, considering it can transmit up to 50 meters away, at a speed of 54 Mbps. The routing protocols used [2-4] implement different functionalities and, consequently, present different energy consumptions.

In "send and receive" operations, the energy consumption of the algorithms [2-4] is the same, using or not using the TIC algorithm because TIC does not interfere in this type of operation. Similarly, our approach does not affect the two metrics (i) loss of meetings and (ii) quantity of unsuccessful scanning, because of the connectivity-inferring algorithm, which uses trained decision trees, always sets an ideal sleep time.

In the traditional model, a node is always scanning for new connections. **It does not happen in TIC model, which is the main difference of our work**. Table 1 highlights that considering the total scanning operations for all network nodes the TIC model accounted for 47.51% fewer operations than the traditional model.

Table 1: Energy consumption on scanning operations.

Routing algorithm	Energy consumption of a single node (mJ)	
	Total	(%)
Traditional	352,291.00	100.0
TIC	184,917.17	52.5
Difference	167,373.83	47.5

5 CONCLUSIONS

We proposed a method for saving energy through the inference of nodes trajectory in DTN. Three different algorithms were studied indicating interesting results. We showed an energy saving of more than 47% in certain operations when compared to traditional methods. The results obtained in this work display that, regardless of the algorithm used, the inference of the nodes trajectory could be used for energy saving in a DTN with promising results. We showed the node's behavior was simulated activating its energy to transmit or to scan only when it is needed, i.e., when there is another node nearby. In any other situation, the node turns off its functions for this activity. This procedure is equivalent to a simple principle: an "energy consumption-aware user" who only activates the network interface of his device when he discovers a connection point. If the user finds few connection points, then the device turns off the interface most of its time, and then it saves more energy from the equipment. Finally, this work demonstrated that our approach provides significant energy savings for a DTN. As future work, we intend to use new machine learning methods, such as neural networks. We also intend to improve the energy model that we use, through tests on real hardware.

REFERENCES

[1] C. Feng, R. Wang, Z. Bian, T. Doiron, J. Hu, "Memory Dynamics and Transmission Performance of Bundle Protocol (BP) in Deep-Space Communications", IEEE Transactions on Wireless Communications, v. 14, n. 5, pp. 2802-2813, May 2015.

[2] A. Vahdat, D. Becker, "Epidemic Routing for Partially-Connected Ad Hoc Networks", Department of Computer Science, Duke University, Technical Report, CS-2000-06, pp. 1-14, Apr. 2000.

[3] T. Spyropoulos, K. Psounis, C. Raghavendra, "Spray and Wait: an Efficient Routing Scheme for Intermittently Connected Mobile Networks", in Proceedings of the ACM SIGCOMM Workshop on Delay-Tolerant Networking (WDTN), pp. 252-259, 2005.

[4] A. Lindgren, A. Doria, O. Schelén, "Probabilistic Routing in Intermittently Connected Networks", ACM SIGMOBILE Mobile Computing and Communications Review, v. 7, n. 3, pp. 19-20, 2003.

[5] K. Dantu, M. Rahimi, H. Shah, S. Babel, A. Dhariwal, G. Sukhatme, "Robomote: Enabling Mobility in Sensor Networks", in Proceedings of the International Symposium on Information Processing in Sensor Networks (IPSN), article 55, pp. 1-6, 2005.

[6] N. Banerjee, M. Corner, B. Levine, "Design and Field Experimentation of an Energy-Efficient Architecture for DTN Throwboxes", IEEE/ACM Transactions on Networking, v. 18, n. 2, pp. 554-567, Apr. 2010.

[7] S. Buruhanudeen, M. Othman, M. Othman, B. Mohd Ali, "Mobility Models, Broadcasting Methods and Factors Contributing Towards the Efficiency of the MANET Routing Protocols: Overview", in Proceedings of the IEEE International Conference on Telecommunications and Malaysia International Conference on Communications, pp. 226-230, 2007.

[8] D. Johnson, D. Maltz, "Dynamic Source Routing in Ad hoc Wireless Networks", Mobile Computing, Kluwer Academic Publishers, v. 353, pp. 153-181, 1996.

[9] K. Chiang, N. Shenoy, "A Random Walk Mobility Model for Location Management in Wireless Networks", in Proceedings of the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), v.2, pp. E-43-E-48, 2001.

[10] J. Torkestani, "Mobility Prediction in Mobile Wireless Networks", Journal of Network and Computer Applications, v. 35, n. 5, pp. 1633-1645, Sep. 2012.

[11] L. Ghouti, T. Sheltami, K. Alutaibi, "Mobility Prediction in Mobile Ad Hoc Networks Using Extreme Learning Machines", Procedia Computer Science, v. 19, pp. 305-312, 2013.

[12] R. Suraj, S. Tapaswi, S. Yousef, K. Pattanaik, M. Cole, "Mobility Prediction in Mobile Ad Hoc Networks Using a Lightweight Genetic Algorithm", Wireless Networks, v. 22, n. 6, pp. 1797-1806, Aug. 2016.

[13] W. Xiao, B. Vallet, K. Schindler, N. Paparoditis, "Street-Side Vehicle Detection, Classification and Change Detection Using Mobile Laser Scanning Data", Journal of Photogrammetry and Remote Sensing, v. 114, pp. 166-178, Apr. 2016.

[14] J. Jetcheva, Y. Hu, S. PalChaudhuri, A. Saha, D. Johnson, "CRAWDAD dataset rice/ad_hoc_city (v. 2003-09-11)", available at crawdad.org/rice/ad_hoc_city/20030911/, Mar. 2017.

[15] K. Fall, "A Delay-Tolerant Network Architecture for Challenged Internets", in Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM), pp. 27-34, 2003.

[16] J. Han, M. Kamber, J. Pei, "Data mining: concepts and techniques", 3rd ed., Elsevier, 703p., 2012.

[17] Wu, Xindong, et al., "Top 10 algorithms in data mining." Knowledge and information systems 14.1, pp. 1-37, 2008.

[18] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA Data Mining Software: An Update". SIGKDD Explorations, Volume 11, Issue 1, pp. 10-18, 2009.

[19] A. Keränen, T. Kärkkäinen, M. Pitkänen, F. Ekman, J. Karvo, J. Ott, "The One Simulator", available at akeranen.github.io/the-one/, August 2017.

[20] D. Silva, A. Costa, J. Macedo, "Energy Impact Analysis on DTN Routing Protocols", in Proceedings of the ACM Extreme Conference on Communication (ExtremeCom), 6p., 2012.