A Bloom Filter-Based Algorithm for Routing in Intermittently Connected Mobile Networks

Jairo Sanchez-Hernandez
Centro de Investigación en Computación
Instituto Politécnico Nacional
jairojsh@gmail.com

Rolando Menchaca-Mendez
Centro de Investigación en Computación
Instituto Politécnico Nacional
rmen@cic.ipn.mx

Ricardo Menchaca-Mendez
Centro de Investigación en Computación
Instituto Politécnico Nacional
ric@cic.ipn.mx

Jesus Garcia-Diaz
Centro de Investigación en Computación
Instituto Politécnico Nacional
jesgadiaz@gmail.com

Mario E. Rivero-Angeles
Centro de Investigación en Computación
Instituto Politécnico Nacional
erivero@cic.ipn.mx

J.J. Garcia-Luna-Aceves
University of California, Santa Cruz
jj@soe.ucsc.edu

ABSTRACT

In this paper, we present a new protocol for routing in intermittently connected mobile networks that, by periodically exchanging constant-size Counting Bloom filters, assigns to every node in the network probabilities of reaching any destination. The gradients defined by these probabilities are further used to forward data packets towards any node in the network. The proposed protocol is based on two novel operations defined over the Bloom filters, namely, the unary *degradation* operation that models the loss of topological information as it gets stale or as it is propagated away from the place where it was generated; and the binary *addition* operation that is used to acquire topological information from other nodes. These two operations are used to implement a probabilistic form of soft state that is defined in terms of the content of the Counting Bloom filters. We present a series of experimental results based on extensive detailed simulations that show that the proposed protocol outperforms the Epidemic routing protocol by delivering more data packets with less delay, while inducing less total overhead in both MANET and VANET scenarios.

Keywords

Delay Tolerant Network, MANET, VANET, Routing, Bloom filters

1. INTRODUCTION

Intermittently Connected Networks (ICNs) are a generalization of the mobile ad hoc networks (MANETs) that does not assume the existence of contemporaneous end-to-end paths connecting any pair of nodes in the network. Since sources and destinations may be disconnected for arbitrarily long periods of time, applications running on this type of networks must be delay tolerant and hence, they are also referred as Delay Tolerant Networks (DTNs) [22]. Delay tolerant networks (DTNs) are a convenient alternative to provide communication services in situations where installing a communication infrastructure is not practical [9]. Examples of such scenarios are search and rescue missions, networks used to collect migration data of Zebras in Africa [10] and information dissemination in sparse vehicular ad hoc networks (VANETs) [7, 12, 21].

An ICN can be modeled by a time varying directed multigraph where each edge represents a particular type of link with a specific capacity. The latter models situations where two nodes can be joined by different types of network interfaces such as a satellital link or a modem-based telephonic link. In general, the capacity of a link can be seen as a time varying function that takes values from zero capacity when the link is not available, to a predefined maximum capacity that reflects the nature of the link. This way, an edge e_n connecting nodes u, v in the multi-graph indicates that there is a link between nodes u, v in the intermittently connected network. An edge (link) can be defined by $e = ((u, v), c_e(t), d_e(t))$ where $c_e(t)$ is a time varying capacity function, $d_e(t)$ is a time varying delay function and (u, v) is an ordered pair that indicates the direction of the link.

The problem of routing in ICNs can be defined as follows. Given a time-varying multigraph $G(t) = (V(t), E(t))$ and two nodes $o, d \in V(t)$ find a sequence of operations $\gamma_1\gamma_2 \ldots \gamma_n$ that take a packet generated by the origin o to the destination d. The set of γ_i operations are part of the alphabet Γ that is defined as follows.

- Transmission operation denoted by $\gamma_{(u,v)}$: A packet is transmitted from node u to node v
- Storing operation denoted by $\gamma_{store}(u)$: A packet is stored in u's local cache until a condition is met and the packet is either transmitted or deleted.
- Copying operation denoted by $\gamma_{copy}(u)$. Node u creates an exact copy of a data packet.

The problem of routing in intermittently connected networks is known to belong to the class of *NP-Hard* problems

MSWiM'15, November 2–6, 2015, Cancun, Mexico.

DOI: http://dx.doi.org/10.1145/2811587.2811609.

[3], even in the case where both, the history of how nodes contact each other and the traffic load are known ahead of time. The proof proceeds by reducing the problem of Edge-Disjoint Paths to the problem of routing in intermittently connected networks.

In this paper, we present a new protocol for routing in ICNs that assigns to every node in the network a probability of reaching any destination. These probabilities are computed by means of the information stored in a set of Counting Bloom Filters that condense the topological information that nodes have collected as they opportunistically encounter other nodes in the network. We say that the information stored at the filters implements a probabilistic form of soft state because it is gradually lost unless it is constantly refreshed and because it provides probabilistic clues about the likelihood of reaching a destination though any node in the network.

The remaining of this paper is organized as follows. Section 2 presents a small summary of the related work. Section 3 presents the proposed routing protocol which is based on a set of novel operations defined over Counting Bloom Filters [6, 19]. Section 4 shows the results of a detailed simulation-based performance analysis of the proposed protocol. We compare the performance of the proposed protocol against that of the Epidemic dissemination protocol which is a *de facto* baseline for performance comparisons of routing protocols for intermittently connected networks. Lastly, Section 5 presents our concluding remarks.

2. RELATED WORK

We present a small but representative sample of the papers that have addressed the problem of routing in intermittently connected networks. Our main propose is to highlight the fact that previous proposal establish individual orderings over the nodes per each of the destinations in the network, and that these orderings are either based on utility functions or probability functions. Unlike these approaches, the proposed protocol simultaneously establishes probability functions for all the destinations using a single constant-size control packet. This property makes the proposed protocol more scalable.

The Epidemic routing algorithm [20] was proposed in 2000 as a simple solution to the problem of routing in delay tolerant networks (DTN). The method used by this algorithm is similar to an infection, where packets represent the disease and the nodes that store a copy of these packets are called "carriers". This way, data packets can spread out very quickly through the connected segments of the network. From this point on, the algorithm exploits the node's mobility to reach other network segments and keep spreading the packets until they reach their intended destination. With this propagation model, packets are almost always delivered, except for the cases in which the destination is located in a completely isolated network component, when packets are discarded at the data queues or when the destination does not even exists. It is important to point out that Epidemic is a brute force algorithm that employs every possible route in the network to reach the destination and hence, is very costly in terms of bandwidth utilization and the memory space needed to store the data packets at the queues. EM-MA [8] is a recently proposed variant of epidemic routing that uses aggregation to reduce overhead, unfortunately, the scalability problems remain.

PRoPHET [13] is a probability based routing protocol for DTNs that take advantage of the repetitive behavioral patterns of the human users. For example, if a node has visited a location many times in the past, there is a high probability that this node will be there in the near future. PRoPHET employs a metric called *delivery probability* $P_{(a,b)} \in [0,1]$ that indicates how probable is that a node u can actually deliver messages to node v. Whenever two nodes come into transmission range, they exchange summaries that contain a list of packets and their delivery probabilities. The information of the summary is then used to request those packets for which the current node is a better forwarder. In [5], the authors propose the space-content adaptive-time routing (SCaTR) framework for DTN MANETs. SCaTR is an extension of traditional on demand routing that takes action when the protocol is unable to establish a connected route. When in DTN mode, SCaTR uses contact values to select the best node that can act as proxy of the destination and store packets at that node until either the destination is discovered, or another node is selected as a better proxy. Contact values are computed based on the past connectivity information that nodes keep in their contact tables. In [18], the authors propose a series of single-copy strategies which are based on utility functions that employ last encounter timers to establish an ordering over the nodes in the network. A data packet is forwarded to a next hop if its priority is larger than that of the current node plus a constant threshold. The authors also propose a hybrid routing protocol, called "Seek and Focus" that first uses randomized forwarding and then utility-based forwarding. In [17] the same authors propose similar multiple-copy strategies for routing in intermittently connected networks. GeoSpray [16] is a routing protocol for Vehicular Delay-Tolerant Networks (VDTN) that employs a hybrid approach of single and multi-copy routing. GeoSpray starts with a multiple-copy approach where a number of copies are spread in order to exploit alternative paths. Then, it switches to a forwarding scheme, where nodes forward data packets to nodes they encounter if they have a better estimated time of delivery.

For a more comprehensive analysis of the large amount of work reporting routing protocols for intermittently connected networks, please refer to the surveys by Benamar et-al [4] and Cao and Sun[7].

Bloom filters or any of their many variants have been used in a large number of applications for distributed systems such as caching, peer-to-peer networks, routing and forwarding, monitoring and measurement and data summarization [19]. Their use as a tool to compute routes, is less extensive, mainly because of the complications introduced by false positives [15]. However, recent proposals such as [14][23] are able ameliorate these complications by either assuming or inducing a tree-like topology. As discussed in Section 3.4, the probability of false positives have no impact on the correctness of the proposed protocol.

3. ROUTING USING COUNTING BLOOM FILTERS

3.1 Overview

The proposed protocol assigns to nodes a probability of reaching every other node in the network. To establish these probabilities throughout the network, nodes periodically ex-

change counting Bloom filters that condense the topological information they have been able to collect. When a node receives a filter from a neighbor, it uses the *addition* operation to merge its current topological information with the one contained in the received filter. Additionally, nodes use the *degradation* operation to implement a probabilistic form of soft state, in which the information about the destinations is gradually lost as it grows stale and as it is being propagated away from the place where it was generated.

From the filters received from each neighbor, nodes compute a probability of reaching any destination D through that particular neighbor. The gradients defined by these probabilities are further used to disseminate data packets in a store-carry-and-forward fashion to their intended destinations. A given data packet will be forwarded to a neighbor when the probability of reaching its intended destination plus a constant threshold, is larger than the probability of the current node. This way, packets tend to travel only to those regions of the network where it is more likely to find their intended destination. Please note that based on the probabilities assigned to nodes, many other forwarding strategies can be defined.

In the following sections, we present more detailed descriptions of the proposed operators and of the way the probabilities of reaching a destination are computed.

3.2 Filter Operations

A *Counting Bloom Filter* F is an array of m counters that can take values from 0 to c. Given a filter F, $F[x]$ denotes the value of the *x-th* counter (with $x \in 0, ..., m-1$) contained in the filter. We use $\mathcal{F}_m^c$ to denote the set of every filter composed of m counters with capacity c. We also denote the empty filter as F_0, where all counters are set to 0. The three operations defined over the filters are as follows.

The *insert* operation is used to add the identity of a node into a Counting Bloom filter. It takes as parameters a filter F and the identifier $a \in ADDR$ of a node (where $ADDR$ is the space of node identifiers in the network) and returns a new filter that contains the identity a. To insert the identity a in a filter F, denoted as $F + a$, the value of k independent hash functions ($h_i : ADDR \rightarrow \{0 \dots m-1\}$ with $i \in \{1 \dots k\}$) is calculated to obtain the indexes of k counters that will be set to the value of c. In this way, the operation defined by $F + a$ results in $F[h_i(a)] \leftarrow c$ with $i \in \{1 \dots k\}$. Note that this operation is different from the traditional insertion operation defined for Counting Bloom Filters where the counters are incremented in one unit.

The *degradation* operation is an unary operation that consists in decrementing in a single unit the value of each counter with probability $p_{degrade}$. This operation is denoted by $\Delta_{p_{degrade}} : \mathcal{F}_m^c \rightarrow \mathcal{F}_m^c$ and is defined by the Algorithm 1. The degradation operation is used to model situations where some of the information contained in a filter is lost; either because it is growing stale or because it is information that is being propagated away from the place it was generated.

Algorithm 1 $\Delta_{p_{degrade}}(F)$

```
for i = 0 → m − 1 do
    if F[i] > 0 then
        F[i] ← F[i] − 1; with a probability p_degrade
    end if
end for
```

The third operation is the binary *addition* operation which is denoted by $\oplus : \mathcal{F}_m^c \times \mathcal{F}_m^c \rightarrow \mathcal{F}_m^c$. This operation combines the value of two filters F_i and F_j following the Algorithm 2.

Algorithm 2 Addition(F_i, F_j)

```
F_x is a new filter;
for l = 0 → m − 1 do
    F_x[l] ← max{F_i[l], F_j[l]}
end for
return F_x
```

3.3 Probabilistic Soft State

Each node i will keep two filters that store its own topological information, namely, F_i^* and F_i^t. F_i^* is constant through time and is defined as $F_0 + i$, whereas F_i^t is time dependent and its content changes because of the updates received from neighboring nodes and from a periodical degradation process. When a node i receives a filter F_j^t from its neighbor j, it updates the value of its own filter F_i^{t+1} according to Eq. (1). Please note that in Eq. (1) the filter received from neighbor j is first degraded to reflect the loss of information and then added to the filter of node i. Additionally, every node i stores in the structure $\mathcal{N}_i$ the latest received filters from all of its neighbors.

$$F_i^{t+1} \leftarrow \Delta_{p_{degrade}}(F_j^t) \oplus F_i^t \tag{1}$$

The filter F_i^t is periodically degraded according to Eq. (2) as a way to implement a form *probabilistic soft state*. The concept of probabilistic soft state is similar to the concept of *Weak State* proposed in [2] in the sense that both provide probabilistic hints regarding the location of the destinations. Unlike Weak State that is based on geographic information, the probabilistic soft state proposed in this work is strictly based on topological information. Moreover, the probabilistic soft state has a spatial/temporal nature because it captures the fact that information is lost as times goes by, but also as it is disseminated away from the place where it was generated.

$$F_i^{t+1} \leftarrow \Delta_{p_{degrade}}(F_i^t) \oplus F_i^* \tag{2}$$

The probability of reaching a destination through node i is calculated with Eq. (3). The probability assigned to each node in the network is precisely the one that defines the probabilistic gradient used to reach the destinations. Note that under this scheme, the probability of reaching node i through itself is always 1.

$$pr_i^D \leftarrow \frac{\sum_{\forall x | F_D^*[x] > 0} F_i^t[x]}{kc} \tag{3}$$

3.4 Protocol Description

Our protocol is defined in terms of a sequence of messages that are exchanged by nodes as they opportunistically meet each other. The messages are as follows.

- BLF: Contains filter F_i^t which condenses the topological information collected by node i.
- SUV: Array of packet identifiers. It is used to publish the set of messages stored at this node.
- REQ: Array of packet identifiers. Set of messages requested by this node.

- DAT: Message that contains a set of data packets.

Every node j broadcasts a hello message m_{BLF} containing his own Bloom filter F_j^t every interval of t_h seconds. As shown in Algorithm 3, upon reception of a message m_{BLF} from node j, node i proceeds to update its own filter by means of Eq. (1) and to compute the new delivery probabilities through node j by means of Eq. (3). Then, node i adds the identifiers of the data packets with destination D, such that the delivery probability at node j is greater than the delivery probability at node i plus a threshold ($pr_j^D \geq pr_i^D + U_e$), to the list L_{suv} which is sent in a m_{SUV} message to j. This way, the packets listed in L_{suv} are such that their destinations are more likely to be reached through i than through j.

Algorithm 3 Bloom Filter Message

when m_{BLF} is received from j do:
  $L_{suv} \leftarrow \emptyset$
  $F_i^t \leftarrow F_i^t \oplus F_i^*$
  $F_i^{t+1} \leftarrow \Delta_{p_{degrade}}(F_j^t) \oplus F_i^t$
  for Message m $\in BUFFER_i$ **do**
    $D \leftarrow m.destination$
    $pr_i^D \leftarrow \frac{\sum_{\forall x | F_D^*[x]>0} F_i^{t+1}[x]}{km}$
    $pr_j^D \leftarrow \frac{\sum_{\forall x | F_D^*[x]>0} F_j^t[x]}{km}$
    if $pr_j^D \geq pr_i^D + U_e \vee pr_j^D = 1$ **then**
      $L_{suv} \leftarrow L_{suv} \cup \{m.id\}$
    end if
  end for
  send(j, L_{suv})
end when

As described by the pseudocode of Algorithm 4, when node i receives the message m_{SUV}, it creates a new list L_{req} that contains the packets listed in L_{suv} that have not been received so far. Then, node i sends a message m_{REQ} back to j requesting the messages in L_{req}.

Algorithm 4 Summary Message

when m_{SUV} is received from j do:
  $L_{req} \leftarrow \{\emptyset\}$
  for Identifier $m_i \in L_{suv}$ **do**
    if $m_i \notin BUFFER_i$ **then**
      $L_{req} \leftarrow L_{req} \cup \{m_i\}$
    end if
  end for
  send(j, L_{req})
end when

When node j receives the list L_{req}, it sends a message m_{DAT} back to i containing every data packet requested by i. This action can be seen as a series of consecutive $\gamma_{(j,i)}$ operations. Lastly, when node i receives the data packets, it passes to upper layers those packets which are intended to the node itself and stores the others in its internal buffer. This latter action can also be seen as a series of $\gamma_{store}(i)$ operations.

As already mentioned, every node i in the network periodically degrades its own filter F_i^t by means of Eq. (2). This way, node i will eventually lose any state regarding destinations that have move far away or to a different network partition.

It is important to point out that in the presence of false positives, data packets can be disseminated towards regions of the network that do not contain the destination, however, this will not prevent the proposed routing protocol to also disseminate the packets towards the true positive, namely, towards the intended destination. Therefore, while a false positive does increase the cost of delivering a data packet, it does not prevent the data packet from reaching its destination. In the worst case, our proposed protocol will behave like a pure epidemic dissemination protocol but with the extra overhead induced by exchanging the Bloom filters.

4. EXPERIMENTAL RESULTS

In this section, we present the results of a series of simulation experiments comparing the performance of the proposed protocol against that of the Epidemic routing protocol. We selected Epidemic because it has become a *de facto* baseline for performance comparisons of routing protocols for intermittently connected networks and because both are multi-copy protocols that disseminate data packets from sources to destinations. The latter allow us to highlight the performance gains obtained by disseminating data packets only towards those regions of the network where it is likely to find the destination. Both protocols use the same tail drop policy to discard data packets when a data queue has reached its maximum capacity, and the same Hello interval of 1 second. For the proposed protocol, the values used for the degradation interval, the degradation probability ($p_{degrade}$) and the U_e threshold were hand-picked. However, a further sensitivity analysis revealed that the performance of the protocol is fairly insensitive to moderate variations ($\pm 15\%$) of these values. We decided to use relatively small filters (with only 64 counters) to characterize the performance of the proposed protocol under relatively high probabilities of false positives. For all the experiments, we used the NS2 simulator version 2.34 [1].

We designed two set of experiments. The first one is intended to evaluate the performance of the protocols in MANET scenarios where nodes are sparsely spread around a large simulation area and move following the random waypoint mobility model. In the second set of experiments, we evaluate the performance of the protocols in a VANET scenario where nodes move following the street layout of the city of Murcia in Spain which is shown in Figure 1. For the two sets of experiments we use packet delivery ratio, end-to-end delay and total overhead as performance metrics. The results presented in all the figures are the average of 20 independent runs with a confidence level of 95%.

4.1 MANET Scenario

For the MANET scenario, we defined a set of three experiments to assess the impact of different variables over the performance of the proposed protocol. In Experiment 1, we vary the amount of traffic injected into the network to evaluate the effectiveness of the protocols to deliver data packets before they have to be dropped at the data queues. In Experiment 2, we decrease the node density so that the networks become more disconnected, and in Experiment 3, we evaluate the effect of decreasing the node mobility. The objective of Experiment 3 is to evaluate the effectiveness of the proposed algorithm to accurately reflect the topological changes into the counting Bloom filters. The details about the values of the simulation parameters that are fixed across the three experiments are shown in Table 1.

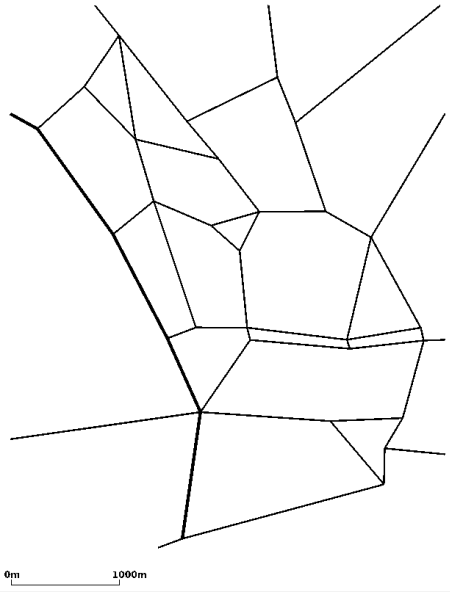

Figure 1: Street layout used in the VANET scenario.

Variable description	Value
Data flows	[1]
Simulation area (square area)	[2]
Pause time	[3]
Mobility model	Random Waypoint
Nodes in the network	100
Node placement	Random
Minimum velocity for nodes	1 m/s
Maximum velocity for nodes	20 m/s
Data flow type	CBR
Packets per data flow	1000
Tx. rate	1 packet/sec
Propagation model	Omni-directional Two-ray ground
Transmission range	250 m
Packet size	128 bytes
Simulation time	5000 seconds
Buffer capacity	2048 packets
U_e Threshold	0.1
Degradation probability ($p_{degrade}$)	0.5
Bloom filter size (m)	64
Counter capacity (c)	32
Number of hash functions (k)	4
Degradation interval	3 seconds
HELLO interval (t_h)	1 second ($\pm$0.25)

Table 1: Simulation environment.

Table 2 shows the specific values of the parameters that are varied in each experiment. The column of *reference values* corresponds to the values of the variables that are fixed during a given experiment.

Figures 2, 3 and 4 present the simulation results for the two protocols under the three different scenarios. For all the experiments, the end-points of the data flows were selected at random using a uniform probability distribution. Figure 2 shows the results of Experiment 1, where we increase the amount of traffic injected into the network by increasing the number of concurrent data flows from 10 to 20. Fig. 3 corresponds to the results of Experiment 2, where we decrease the node density by keeping constant the number of nodes and by increasing the simulation area. Lastly, Figure 4 shows the results of Experiment 3, where we reduce the mobility of the nodes by increasing the pause time from 10 seconds to 40 seconds.

Experiment	Values	Reference value
1	10,15,20 flows	10
2	2000m×2000m 3000m×3000m 4000m×4000m	3000m×3000m
3	10s,20s,40s	0s

Table 2: Specific parameters for each experiment.

Regarding packet delivery ratio, from Figures 2(a), 3(a) and 4(a) we can observe that the proposed protocol clearly outperforms Epidemic by consistently delivering more data packets. The inferior packet delivery ratio attained by Epidemic is caused by a rapid saturation of the buffers of the nodes. On the other hand, the proposed algorithm only propagates packets towards those regions of the network where it is more likely to find the destination and hence it does not waste buffer space of nodes that are located far away from the intended destinations (see Figure 2(a)). For the case of increasing traffic load, the behavior shown by the proposed protocol is as expected. When the traffic increases, the packet delivery ratio is reduced because more packets are being disseminated across the network and hence, more packets have to be dropped at the data queues. On the other hand, when the node density is increased (see Figure 3(a)), nodes tend to meet each other more often and hence, the topological information regarding the destinations percolates faster across the network. Under this situation, the probabilistic gradients are more clearly defined and the dissemination of the packets tends to be more focused towards the destinations. It is also important to point out that when the network becomes more connected, the sequence of SUV-SUV-REQ-DAT messages also becomes more costly. Moreover, we observed the appearance of the broadcast storm problem where many nodes simultaneously engage in SUV-REQ-DAT exchanges with a node transmitting a BLF packet. Lastly, Figure 4(a) shows that the two protocols are fairly insensitive to the different values used for the pause time. These results indicate that under these conditions, the probabilistic soft state used by the proposed protocol is capable of keeping up with the topological changes experienced by the network.

Regarding end-to-end delay, from Figures 2(b), 3(b) and 4(b) we can observe that the proposed protocol clearly outperforms Epidemic by consistently attaining smaller delays. This was also an expected result because Epidemic is a brute force solution that uses all possible paths to reach the intended destinations but that tend to saturate the data queues much faster. Moreover, a detailed analysis of the simulation traces revealed that the delay attained by Epidemic is also due to the fact that most of the packets were delivered to destinations that happened to be relatively close to their corresponding sources. The latter contrast with the proposed protocol that is able to deliver packets to destinations that are located far away from the sources, or in different network partitions.

Lastly, from Figures 2(c), 3(c) and 4(c) we can notice that the extra cost induced by the proposed protocol by periodically transmitting counting Bloom filters is clearly outweighed by the data overhead induced by Epidemic while disseminating data packets towards regions of the network that do not contain the intended destinations.

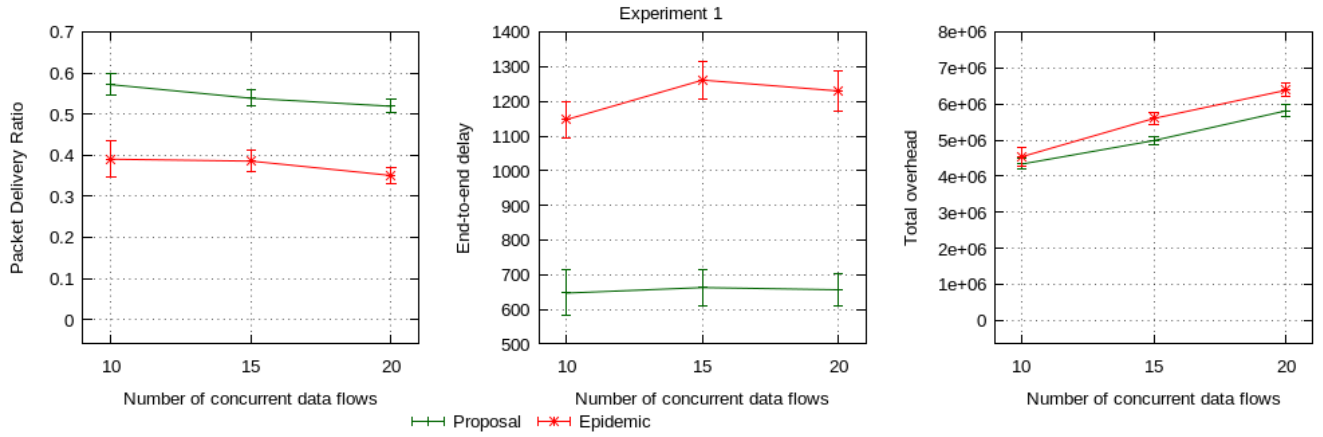

Figure 2: Performance of the protocols in a MANET scenario with increasing traffic load. (a) Packet delivery ratio. (b) End-to-end delay. (c) Total overhead.

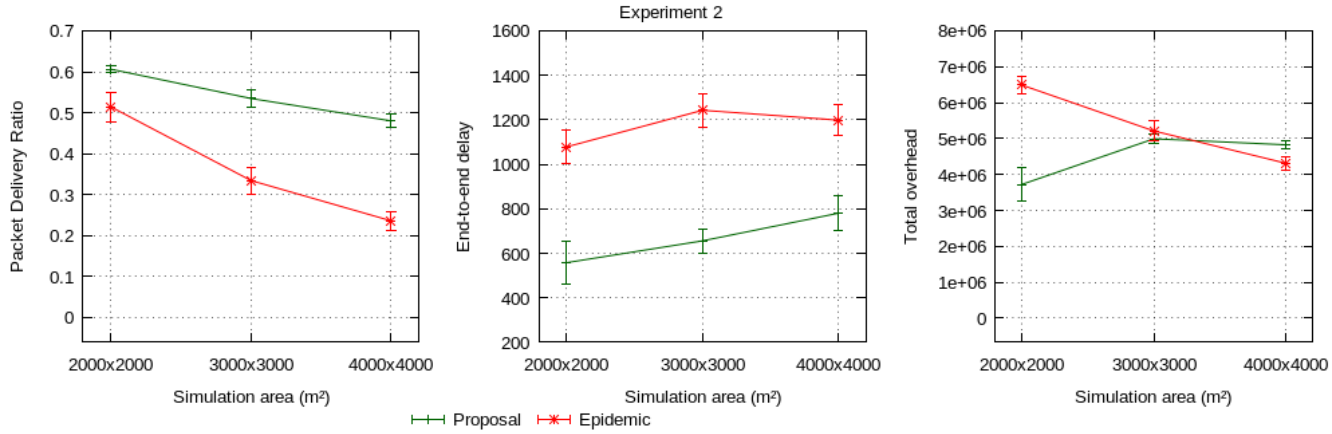

Figure 3: Performance of the protocols in a MANET scenario with decreasing node density. (a) Packet delivery ratio. (b) End-to-end delay. (c) Total overhead.

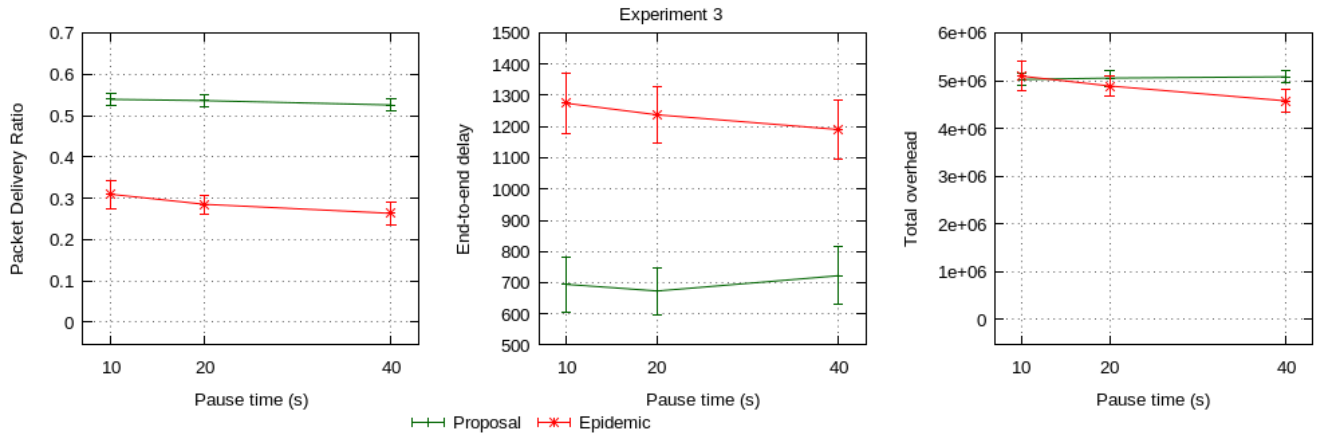

Figure 4: Performance of the protocols in a MANET scenario with decreasing node mobility. (a) Packet delivery ratio. (b) End-to-end delay. (c) Total overhead.

Vehicles entering the scenario every 45s	Flows	Routes	Avg. num. of vehicles
05	25	25	55
15	25	25	155
25	25	25	270
35	35	30	385
45	55	30	520

Table 3: Vehicle flow parameters for the VANET scenario.

Parameter	Value
Data flows	10
Packets per flow	1000
Packet size	512 bytes
Data rate	1 packet/s

Table 4: Parameters for VANET data flows

4.2 VANET Scenario

In this scenario, we evaluate the performance of the protocols when nodes move following the street layout of the city of Murcia, Spain (Figure 1). We used the SUMO simulator of urban mobility [11] to generate mobility traces of two types of vehicles, namely, *cars* and *buses*, which differ only on mobility parameters such as acceleration, maximum speed, and size. For this experiments we use the same set of values for the simulation parameters as described in Table 1, and vary the vehicle density by increasing the number of vehicles entering the simulation area per second as shown in the first column of Table 3. The second column of Table 3 corresponds to the number of different vehicle flows in the scenario. In SUMO, a flow defines the rate at which vehicles enter the scenario, as well as their type (car or bus) and the route they follow. Routes are defined in terms of a sequence of street intersections. The fourth column of Table 3 presents an estimate of the average number of active vehicles in the simulation area. This value is a function of the vehicle arrival rate and, the time vehicles remain active on the simulation, which depends on the length of the routes followed by the vehicles, the behavior of the semaphores, and the traffic congestion of the road network. As in the MANET scenario, the end-points of the CBR data flows are selected uniformly at random from the vehicles currently in the simulation. The details of the vehicle flow parameters are shown in Table 4.

Figure 5 presents the results of these experiments. From Figures 5(a) and (b), we can observe that the proposed algorithm consistently performs similar or better than Epidemic in terms of packet delivery ratio, end-to-end delay and total overhead. Our proposal delivers between 5% and 7% more packets than Epidemic, even in situations where the probability of false positives in the counting Bloom filters is quite high due to the large number of active vehicles in the simulation and the relative small size of the filters (32 counters). Under this situation, our protocol behaves similar to Epidemic because it also tend to disseminate data packets towards regions of the network that do not contain the destination but a false positive. Figure 5(b) shows that in these experiments the end-to-end delay attained by the proposed protocol is slightly better than that of Epidemic. The reason again, is the fact our protocol induces less data overhead, which reduces the waiting times at the data queues.

Lastly, Figure 5(c) shows the total overhead induced by the two protocols. As in the case of the MANET scenario, the proposed protocol incurs in far less overhead because of the reduced cost of disseminating only towards regions of the network where it is likely to find the destinations. This is true even in this scenario where the probability of false positive is high.

5. CONCLUSIONS

In this paper, we proposed a new protocol for routing in intermittently connected mobile networks that uses the concept of *probabilistic soft state* to assign to every node in the network probabilities of reaching any destination. The probabilistic soft state is defined in terms of the content of counting Bloom filters that condense the topological information that nodes have collected from other nodes as they move and that is used to compute the probabilities of reaching the destinations. These probabilities reflect the amount of information nodes have regarding the different destinations. To model the process in which nodes acquire and loss topological information, we defined two novel operations over counting Bloom filters, namely, the binary *addition* operation and the unary *degradation* operation. The addition operation is used to reflect the acquisition of new information, and the unary operation is used to reflect the loss of information, either because it is getting stale or because it is being propagated away from the place where it was generated. Then, nodes use a greedy forwarding strategy that follows the gradients defined by these probabilities to reach the intended destinations.

The proposed scheme has the advantage of establishing probabilistic gradients simultaneously for all the destinations. This contrasts with previous proposals that require the dissemination of control information for every destination node, which is not scalable. In the proposed scheme, nodes periodically exchange constant-size "hello" messages with their one-hop neighbors. Therefore, the protocol's network complexity is constant in terms of the network size. Moreover, the proposed scheme has the advantage that it combines in a simple way the concepts of probability of encounter and hop distance towards the destinations. When the network is connected, the probabilistic gradients are similar to the traditional gradients based on hop distances, but when the network is partitioned, the gradients are similar to those that compute a probability of encounter based on the time nodes last met. This way, the proposed algorithm can work in more general scenarios than previous protocols that exploit the "social behavior" of nodes.

In the presence of false positives in the Bloom filters, data packets can be disseminated towards regions of the network that do not contain the destination, however, this will not prevent the proposed protocol to also disseminate the packets towards the intended destination. We observed this situation in the VANET scenario, where due to the large number of nodes, the probability of false positive is quite high.

The results of a detailed simulation-based performance analysis showed that the proposed algorithm is a viable alternative for routing in both intermittently connected MANETs and VANETs. The proposed protocol consistently outperformed the Epidemic routing protocol in both MANET and VANET scenarios.

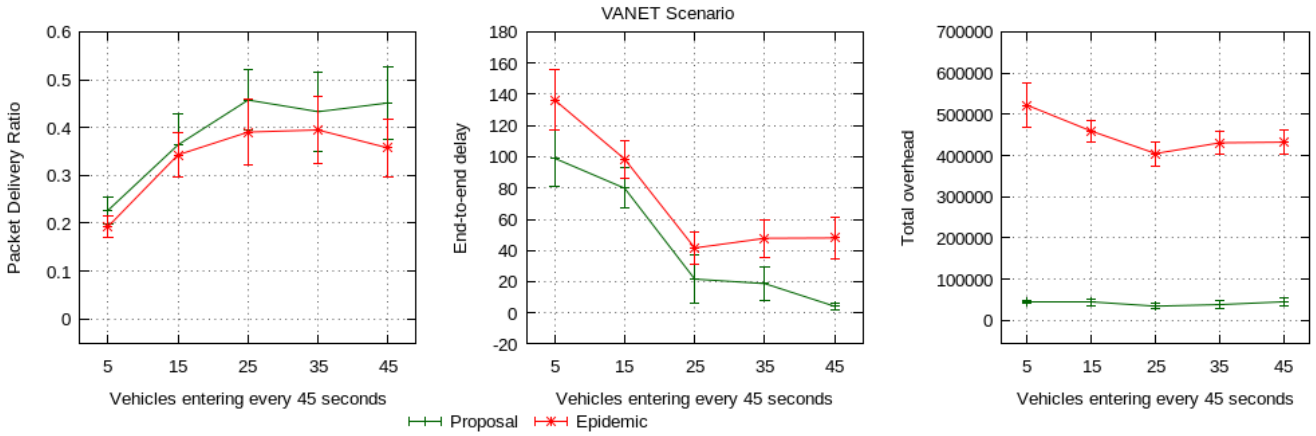

Figure 5: Performance of the protocols in a VANET scenario with increasing node density. (a) Packet delivery ratio. (b) End-to-end delay. (c) Total overhead.

6. ACKNOWLEDGMENTS

Work partially sponsored by the UC MEXUS-CONACyT program, and by the Mexican National Polytechnic Institute (IPN).

7. REFERENCES

[1] The Network Simulator, ns-2.34. http://www.isi.edu/nsnam/ns/.

[2] U. G. Acer, S. Kalyanaraman, and A. A. Abouzeid. Weak state routing for large-scale dynamic networks. *IEEE/ACM Transactions on Networking (TON)*, 18(5):1450–1463, 2010.

[3] A. Balasubramanian, B. Levine, and A. Venkataramani. DTN routing as a resource allocation problem. *SIGCOMM Comput. Commun. Rev.*, 37(4):373–384, Aug. 2007.

[4] N. Benamar, K. D. Singh, M. Benamar, D. El Ouadghiri, and J.-M. Bonnin. Routing protocols in vehicular delay tolerant networks: A comprehensive survey. *Computer Communications*, 48:141–158, 2014.

[5] J. Boice, J. Garcia-Luna-Aceves, and K. Obraczka. Combining on-demand and opportunistic routing for intermittently connected networks. *Ad Hoc Networks*, 7(1):201–218, 2009.

[6] A. Broder and M. Mitzenmacher. Network applications of bloom filters: A survey. *Internet Mathematics*, 1(4):485–509, 2004.

[7] Y. Cao and Z. Sun. Routing in delay/disruption tolerant networks: A taxonomy, survey and challenges. *IEEE Communications Surveys & Tutorials*, 15(2):654–677, 2013.

[8] T. Choksatid and S. Prabhavat. An epidemic routing with low message exchange overhead for delay tolerant networks. In *Progress in Systems Engineering*, pages 429–436. Springer, 2015.

[9] S. Jain, K. Fall, and R. Patra. Routing in a delay tolerant network. In *Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '04, pages 145–158, New York, NY, USA, 2004. ACM.

[10] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein. Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with zebranet. *SIGOPS Oper. Syst. Rev.*, 36(5):96–107, Oct. 2002.

[11] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker. Recent development and applications of SUMO - Simulation of Urban MObility. *International Journal On Advances in Systems and Measurements*, 5(3&4):128–138, December 2012.

[12] F. Li and Y. Wang. Routing in vehicular ad hoc networks: A survey. *IEEE Vehicular Technology Magazine*, 2(2):12–22, 2007.

[13] A. Lindgren, A. Doria, and O. Schelén. Probabilistic routing in intermittently connected networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 7(3):19–20, July 2003.

[14] A. Reinhardt, O. Morar, S. Santini, S. ZoÌĹller, and R. Steinmetz. Cbfr: Bloom filter routing with gradual forgetting for tree-structured wireless sensor networks with mobile nodes. In *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2012*, pages 1–9. IEEE, 2012.

[15] M. Sarela, C. E. Rothenberg, T. Aura, A. Zahemszky, P. Nikander, and J. Ott. Forwarding anomalies in bloom filter-based multicast. In *Proceedings of IEEE INFOCOM 2011*, pages 2399–2407. IEEE, 2011.

[16] V. N. Soares, J. J. Rodrigues, and F. Farahmand. Geospray: A geographic routing protocol for vehicular delay-tolerant networks. *Information Fusion*, 15:102–113, 2014.

[17] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Efficient routing in intermittently connected mobile networks: the multiple-copy case. *IEEE/ACM Transactions on Networking*, 16(1):77–90, 2008.

[18] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Efficient routing in intermittently connected mobile networks: The single-copy case. *IEEE/ACM Transactions on Networking (TON)*, 16(1):63–76, 2008.

[19] S. Tarkoma, C. E. Rothenberg, and E. Lagerspetz. Theory and practice of bloom filters for distributed systems. *IEEE Communications Surveys & Tutorials*, 14(1):131–155, 2012.

[20] A. Vahdat and D. Becker. Epidemic routing for partially-connected ad hoc networks. Technical report, Duke University, 2000.

[21] N. Wisitpongphan, F. Bai, P. Mudalige, V. Sadekar, and O. Tonguz. Routing in sparse vehicular ad hoc wireless networks. *IEEE Journal on Selected Areas in Communications*, 25(8):1538–1556, 2007.

[22] Z. Zhang. Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: overview and challenges. *IEEE Communications Surveys & Tutorials*, 8(1):24–37, 2006.

[23] H. Zheng and J. Wu. Up-and-down routing in mobile opportunistic social networks with bloom-filter-based hints. In *IEEE 22nd International Symposium of Quality of Service (IWQoS) 2014*, pages 1–10, May 2014.