Analysis and Evaluation of the Hop Expansion Routing Algorithm (HERA) for Delay-Tolerant Networks

Alexander K. Yee
Department of Electrical Engineering
San José State University
San Jose, CA 95192

Pedro E. Santacruz
Department of Electrical Engineering
San José State University
San Jose, CA 95192

***Abstract*—This paper presents an analysis and evaluation of the Hop Expansion Routing Algorithm, HERA, for delay-tolerant networks. HERA works by computing a reachability metric, akin to the metric used in PRoPHET routing, but with the major distinction that HERA expands the computation of the metric into the contributions of paths of different lengths, thus providing a more flexible metric that allows for the fine tuning of the effects of transitivity. In particular, HERA gives the flexibility to address the question of how many hops of transitivity a routing algorithm should take into account when making forwarding decisions. We present simulation results showing improvement of HERA over PRoPHET in terms of delivery ratio, number of replicas, and latency for several mobility scenarios, including real-life traces.**

I. Introduction

As the number and diversity of connected devices increases, it becomes more and more important to design more efficient and appropriate networks that will help support this connectivity. Many of the new devices entering the networked domain are portable mobile devices designed for the Internet of Things (IoT). While most of the connected devices rely on infrastructure networks such as cellular or WiFi, the characteristics of modern devices pose a new series of challenges that need to be addressed. Consider, for example, the problem of overloaded infrastructure networks due to the sheer number of connected devices, lower computational capacity of IoT devices, and mismatch between the type of data being exchanged between the new IoT devices and the traditional networked devices for which the networks were originally designed.

There is a wide variety of newer technologies that can be used to support the networks that will enable the connectivity of IoT devices. There has been extensive work on communication technologies beyond infrastructure networks with particular emphasis on device-to-device communication including Bluetooth, Bluetooth Low Energy (BLE), ZigBee, and z-Wave. Furthermore, it is important to leverage these technologies by matching them correctly to the particular characteristics of modern use case scenarios. Many new IoT devices operate under the framework of low data rate and low duty cycle, as is the case in the use of sensors, smart home appliances, and space occupancy trackers. Use case scenarios such as these provide a perfect opportunity to relieve the infrastructure networks of unnecessary traffic by leveraging alternate technologies that are more apt to handle the characteristics of IoT device traffic. In this context, the delay-tolerant network (DTN) paradigm presents a fitting system model to explore and improve the network performance for IoT devices.

In this work, we present an in-depth analysis of the Hop Expansion Routing Algorithm (HERA) for delay-tolerant networks which takes advantage of explicitly differentiating among different hops of connectivity in order to make more accurate forwarding decisions as nodes in the networks encounter each other. Similar to previous DTN routing algorithms, such as PRoPHET [1] and its various iterations [2] [3], our algorithm endows nodes in the network with a metric for each possible destination in the network and forwarding decisions are made based on that metric. The metric typically represents a "likelihood" of a specific node being able to deliver the packet to its destination, either by delivering it itself or relaying it to some other node that can ultimately deliver it. In order to capture this ability to forward the packet to its destination through several hops, PRoPHET and its variants include the concept of transitivity and add it to their metric computation. Our algorithm presents a novel approach to the concept of transitivity by keeping track of the number of hops of connectivity in a possible delivery path and adding weights to number of hops of transitivity. This results in an algorithm that provides users much more flexibility in determining how much value longer paths should contribute to the delivery likelihood metric.

Our major contributions can be summarized in the following four points:

1) We introduce and describe the details of the novel HERA algorithm
2) We produce an implementation of HERA within The ONE simulator for evaluation
3) We present simulation results comparing the performance of HERA with PRoPHET and show improvements in delivery ratio, number of replicas, and latency for realistic vehicular mobility scenarios
4) We analyze the effects of algorithms' parameters in algorithm performance

II. System Model

In this work we envision a network of mobile, short-range devices capable of device-to-device communication. Bluetooth and ZigBee are two examples of the technologies that satisfy these conditions, but in our paper we do not constrain the technology being used. There are N devices in the network $\mathcal{N}$. We model our network as a DTN with mobile nodes that come in and out of range of other nodes. The transmission range of each node $n \in \mathcal{N}$ is denoted r. When a node moves within range of another node, we assume the occurrence of a Connection Up event, and similarly when a node moves outside the range of another node a Connection Down event occurs.

Every node in the network has the ability to generate messages intended for any other node in the network. A message $x(i, j)$ is generated by a *source* node i with a *destination* node j. We define the delivery ratio as the number of messages delivered to their final destination across the entire network divided by the total number of messages generated at all sources throughout the network. We also define the number of replicas as the total number of times a message is communicated from one node to another, whether it is a relaying node, a source or destination. We measure the delay of a delivered message as the time it take to travel from source node to destination node.

III. Hop Expansion Routing Algorithm (HERA) Description

HERA is a routing protocol for DTNs that consists of three main components: 1) a reachability matrix maintained by each node in the network, 2) the mechanisms for updating the reachability matrix, and 3) a decision-making mechanism that resolves whether or not a message should be forwarded when an encounter occurs in the network. In this section we explain the details of these three components.

A. Reachability Matrix

Each node i in the network will maintain a *reachability matrix*, M_i. The entries of the matrix consist of individual metrics to each node in the network, analogous to the predictability metrics used in PRoPHET routing. The key difference in the approach presented by our proposed algorithm compared to PRoPHET routing is that for each destination in the network, HERA maintains up to $H+1$ different metrics. For example, node i will maintain $H+1$ separate reachability metrics for node j, $m^{(0)}(i,j), m^{(1)}(i,j), \ldots, m^{(H)}(i,j)$. The $H+1$ different metrics are separated in order to be able to reflect the level of confidence of each metric based on the number of hops required to reach a destination. In other words $m^{(0)}(i,j)$ is based on direct encounters between node i and node j, $m^{(1)}(i,j)$ is based on node i's encounters with nodes that have had direct contact with node j (i.e., a path with one relay), and so on. The reachability matrix M_i has the following form:

$$M_i = \begin{bmatrix} m^{(0)}(i,1) & m^{(0)}(i,2) & \ldots & m^{(0)}(i,N) \\ m^{(1)}(i,1) & m^{(1)}(i,2) & \ldots & m^{(1)}(i,N) \\ \cdots & \cdots & \cdots & \cdots \\ m^{(H)}(i,1) & m^{(H)}(i,2) & \ldots & m^{(H)}(i,N) \end{bmatrix}$$

This results in each node having an $(H+1) \times N$ matrix where H is the number of relays in the largest-hop path we will be preserving for transitivity effects and N is the number of nodes in the network. Each column in the matrix now represents all the reachability metrics associated with a specific node.

B. Metric Updates

Every time a node in the network comes within range of another node, we declare a Connection Up event and the two nodes involved in the connection must update the appropriate entries in their reachability matrix. Suppose node i encounters node j. The first step after their encounter is to exchange their reachability matrices, such that node i will receive the matrix M_j from node j and vice versa. Node i will update the metric for the node which it has directly encountered as follows:

$$m^{(0)}_{new}(i,j) = m^{(0)}_{old}(i,j) + 1 \tag{1}$$

Additionally, node i will update the metric for all nodes in the network as follows:

$$m^{(h)}_{new}(i,k) = m^{(h)}_{old}(i,k) + \beta_h m^{(h-1)}_{old}(j,k) \tag{2}$$

for all $h = 1, \ldots, H$, and for all $k \neq j \in \mathcal{N}$. This metric $m^{(h)}(i,k)$ is the value that will contribute to the reachability metric due to the transitivity effects, but with the special property that HERA separates the transitivity value on the number of hops of separation from direct contact. The value $\beta_h \in [0,1]$ allows us to determine the impact of transitivity from one hop of separation to the next.

Also, we add a metric aging process to account for the fact that after time has gone by, the metric values are not up to date. Every time a Connection Up event occurs, the metrics are also updated as follows:

$$m^{(h)}_{new}(i,j) = m^{(h)}_{old}(i,j) \times \alpha^{\delta} \tag{3}$$

where α is the aging constant and δ is the time elapsed since the last Connection Up event.

C. Forwarding Decision Process

When an encounter occurs, the nodes must make a decision about whether they should forward any of the packets they have in their queue. The decision is based on the value of the *destination reachability* of the nodes involved in the encounter. The destination reachability of a node i to a destination d is a weighted sum of all the reachability metrics maintained by the node and is computed by

$$\Omega_i(d) = \gamma_0 m^{(0)}(i,d) + \gamma_1 m^{(1)}(i,d) + \cdots + \gamma_H m^{(H)}(i,d). \tag{4}$$

Now, suppose node i encounters node j. The forwarding decision can be described as follows: if node i has a packet for destination d, it will forward that packet to node j if

$$\Omega_j(d) > \Omega_i(d). \tag{5}$$

The weight vector $\boldsymbol{\gamma} = [\gamma_0, \gamma_1, ..., \gamma_H]$ allows us to determine how much metrics of different hops of separation will affect the overall destination reachability. The ability to tune these contributions is the key factor that sets HERA apart from other similar algorithms. While in most cases, the metric of likelihood to reach a destination is condensed into a single variable from the very start, in HERA we have the capability to decide the appropriate weights due to our hop expansion approach. The reason why this is significant is because, in realistic scenarios, direct encounters with the destination are more reliable than possible indirect contacts. In fact, other algorithms add the effects of transitivity precisely for that reason, yet they do not differentiate between reliability of forwarding a message through a path with one, two, or more relaying nodes.

D. Transitivity in HERA

In this algorithm, we now have the parameters H, $\boldsymbol{\beta} = [\beta_0, \beta_1, ..., \beta_H]$, and $\boldsymbol{\gamma} = [\gamma_0, \gamma_1, ..., \gamma_H]$ at our disposal to fine tune the algorithm depending on the characteristics of the network and the mobility scenario. The value of H represents the largest-hop path we should take into account when calculating reachability to a destination node. A larger H will keep track of paths of longer lengths, but it is important to note that it also requires larger overhead of control communication exchanges. The values of $\boldsymbol{\beta}$ represent the intrinsic confidence one should put on indirect knowledge as opposed to direct encounters. The values of $\boldsymbol{\gamma}$ represent the weight each possible path of a specific length should contribute to the decision making process.

IV. Evaluation Setup

To evaluate the performance of HERA, we use The Opportunistic Network Environment (The ONE) simulator [4]. The ONE simulator presents an environment that allows us to conduct simulations using different mobility scenarios, either from traces or following specific movement models, and provides a platform to route messages between nodes in DTN environments. The ONE's architecture allows us to write our own routing algorithm following The ONEs programming conventions and plug it into the rest of the simulation infrastructure for quick evaluation.

We have developed an implementation of the HERA algorithm inside The ONE simulator using the description detailed in Section III. In our implementation, we allow for easy configuration of the parameters $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$. Our goal is to analyze the effects of different configuration parameters and also compare the performance of HERA with that of PRoPHET routing. Our metrics of performance comparison are 1) delivery ratio, 2) number of replicas, and 3) average latency.

In order to provide a comprehensive evaluation of HERA, we test its performance in 3 different mobility scenarios. The first scenario is called *Cluster*. In the Cluster scenario, we have 125 total nodes that are divided into 3 different clusters, each cluster containing 40 nodes and 5 mules. Using the mobility generation capabilities of The ONE, we configure nodes to move within each cluster creating and destroying connections to other nodes in the same cluster. The clusters are separated by enough distance that nodes from different clusters never come in contact with each other. There are 5 nodes that act as mules traveling from cluster to cluster and can deliver packets intended for destinations located outside the source's range. We use this scenario because it illustrates one of the challenges encountered by PRoPHET routing, in particular the Parking Lot problem described in [2]. Our goal is to show that HERA addresses this well-known challenge and outperforms the traditional approach.

The second mobility scenario used for evaluation is called *Taxi*. In the Taxi scenario we use real-life traces from the epfl/mobility data set from CRAWDAD [5]. The Taxi scenario uses traces of GPS locations of taxis in San Francisco, USA. We use a subset of the complete dataset and use 25 taxis over the span of 3 days. We chose the San Francisco taxi traces because they provide a realistic environment where HERA could be used and, in particular, to illustrate its potential to be used in smart city applications.

The third mobility scenario is called the *Working Day Movement (WDM)* scenario. The WDM scenario uses the mobility characteristics described in [6] and is modeled to represent accurate movement across a city (Helsinki, in this case) over a 24-hour period. This scenario includes movement in cars, buses, and by foot. The movement is overlayed on a map of roads creating a simulation with more realistic behaviors compared to other models such as random waypoint. An implementation of the WDM scenario is already included in The ONE simulator. This scenario was chosen to evaluate HERA in realistic environments but with more flexibility in the movement generation since it comes from a simulation model rather than fixed traces.

In all three different scenarios, messages to be delivered are generated randomly. The source for a given message is chosen randomly with uniform distribution over all nodes in the network. A message is generated every 25 to 35 seconds, and each message has a random size uniformly distributed from 50 kB to 150 kB. The communication capabilities of nodes in the network are simulated to be Bluetooth interfaces communicating at a rate of 250 kBps and with a range of 10 meters.

V. Results

In this section, we present the performance results of the comparison between PRoPHET and HERA routing algorithms in terms of 1) delivery ratio, 2) number of replicas, and 3) average latency.

A. Cluster Scenario

Figure 1 shows the performance comparison under the Cluster mobility scenario. As we mentioned in Section IV, the Cluster scenario is known to traditionally present challenges to the PRoPHET routing algorithm. The simulation results show that the HERA algorithm is able to outperform PRoPHET in the three metrics of performance. HERA had a larger delivery ratio, while requiring a smaller number of message replicas, and reducing the average latency.

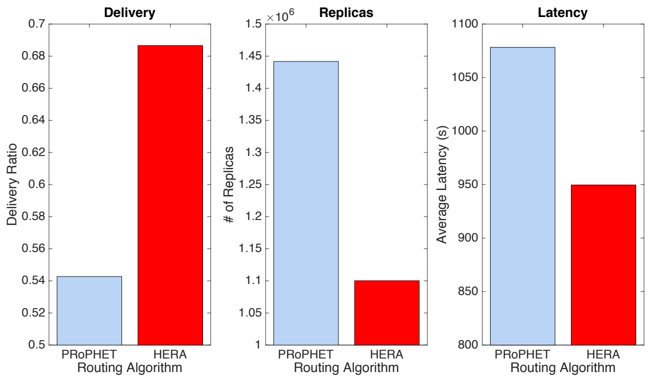

Fig. 1: Simulation results for Cluster Scenario

It is important to note that this performance of the HERA algorithm is achieved for a particular set of parameters. In this case, the results shown are for the parameters $H = 2$, $\boldsymbol{\gamma} = [0.5, 0.05]$, and $\boldsymbol{\beta} = [0.2, 0.02]$. Different parameter values offer different performance results, and in some cases, below the performance of PRoPHET. Nevertheless, this ability to fine tune parameters for performance is what sets HERA apart from other routing algorithms. We will discuss the effects of the parameters in detail in the next section.

B. Taxi Scenario

The next set of results presented in Figure 2 corresponds to the Taxi scenario. In this scenario, HERA again outperforms PRoPHET in all three metrics, although it does it by a smaller relative margin. The results shown in Figure 2 were obtained with the parameters $H = 4$, $\boldsymbol{\gamma} = [0.25, 0.25, 0.125, 0.125]$, and $\boldsymbol{\beta} = [0.8, 0.8, 0.8, 0.8]$.

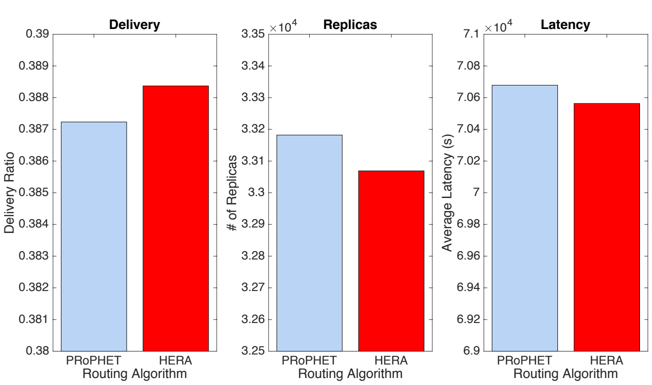

Fig. 2: Simulation results for Taxi Scenario

In this scenario, we would like to highlight that HERA offers the flexibility to operate at different points on the tradeoff space by changing parameter values. For example, if we use parameters $H = 4$, $\boldsymbol{\gamma} = [0.5, 0.5, 0.25, 0.25]$, and $\boldsymbol{\beta} = [0.9, 0.9, 0.9, 0.9]$, we obtain the performance shown in Figure 3. Using these values, HERA shows an increased improvement in terms of latency at the cost of performance in terms of delivery ratio and replicas.

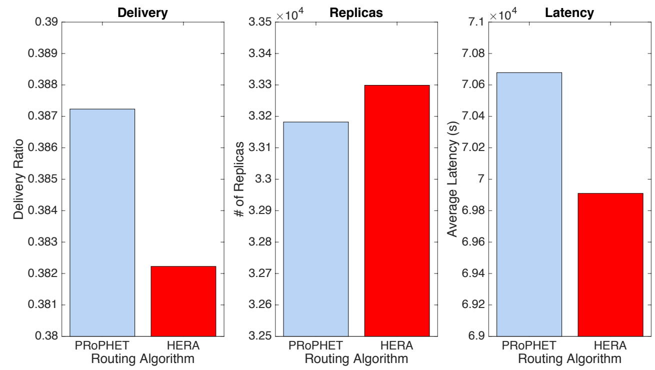

Fig. 3: Simulation results for Taxi Scenario with updated parameters

C. Working Day Movement Scenario

The third scenario studied in our comprehensive evaluation of HERA is the Working Day Movement model, which simulates 1028 nodes moving around a city, either in cars, buses, or walking. Figure 4 displays the improvement HERA achieves over PRoPHET in delivery ratio and number of replicas. For this scenario, HERA is not able to overcome the average latency performance provided by PRoPHET while maintaining gains in delivery ratio and number of replicas. The results were obtained with the following set of parameter values, $H = 4$, $\boldsymbol{\gamma} = [0.5, 0.05, 0.005, 0.0005]$, and $\boldsymbol{\beta} = [0.5, 0.05, 0.005, 0.0005]$.

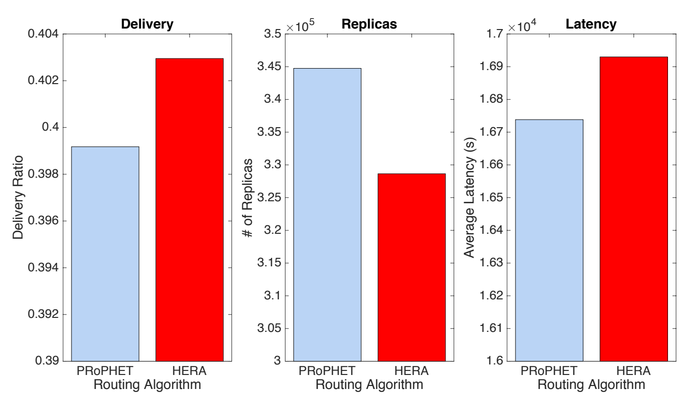

Fig. 4: Simulation results for WDM Scenario

A summary of the simulation results with the values for each one of the metrics plotted in Figures 1-4 can be found in Table I.

Scenario	Routing Alg.	Parameters, γ and β	Delivery Ratio	# of Replicas	Avg. Latency (s)
Cluster	PRoPHET	-	0.542699725	1441707	1078.2036
	HERA	$\boldsymbol{\gamma} = [0.5, 0.05]$, $\boldsymbol{\beta} = [0.2, 0.02]$	0.686639118	1100241	949.6258
Taxi	PRoPHET	-	0.387232590	33182	70678.4946
	HERA	$\boldsymbol{\gamma} = [0.25, 0.25, 0.125, 0.125]$, $\boldsymbol{\beta} = [0.8, 0.8, 0.8, 0.8]$	0.388370505	33069	70563.3299
	HERA	$\boldsymbol{\gamma} = [0.5, 0.5, 0.25, 0.25]$, $\boldsymbol{\beta} = [0.9, 0.9, 0.9, 0.9]$	0.382225762	33299	69909.8872
WDM	PRoPHET	-	0.399176955	344744	16738.2583
	HERA	$\boldsymbol{\gamma} = [0.5, 0.05, 0.005, 0.0005]$, $\boldsymbol{\beta} = [0.5, 0.05, 0.005, 0.0005]$	0.402949246	328645	16929.8889

TABLE I: Summary of Simulation Results

VI. Analysis

In this section, we analyze some of the characteristics of the HERA algorithm more in detail. First, we discuss the source of HERA's improvement over PRoPHET by showing the evolution of the decision-making metrics. Then, we present guidelines for the choice of HERA's parameters. And finally, we give some insight into the role of transitivity and how to optimize the value of H.

A. Evolution of Destination Reachability

In order to understand the mechanisms that allow HERA to show improved performance, we focus on the evolution of the Destination Reachability metric. Figure 5 illustrates the behavior of this metric for two different nodes in the Cluster scenario. One of the nodes belongs to a cluster, while the other node is a mule. We can see how the mule increases its reachability significantly when it comes into contact with the destination. On the other hand, the cluster node also increases it reachability in some instances because of transitivity effect, but because of our choices of γ and β, the increase is kept under control. More importantly, the cluster node reachability is below the reachability of the mule for an extensive part of the time, and so it will make the correct choice of forwarding a packet to the mule in most cases. This shows that HERA solves the problem of an artificially inflated predictability due to uncontrolled transitivity effects in a large number of encounters, which is typically found in PRoPHET routing.

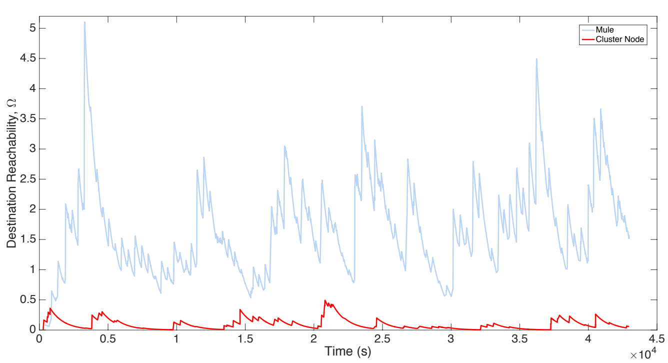

Fig. 5: Evolution of Destination Reachability (HERA) in the Cluster Scenario

Figure 6 shows the delivery predictability of the same two nodes in the Cluster scenario, but this time using the PRoPHET

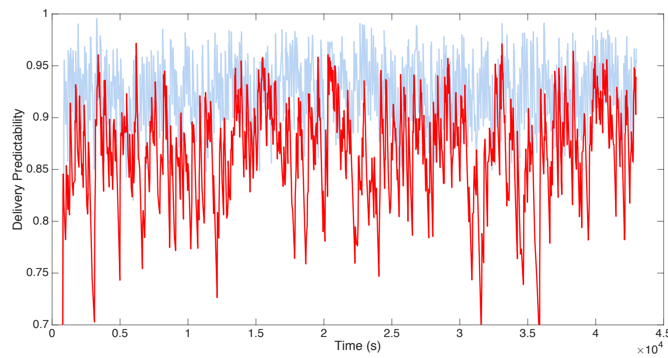

Fig. 6: Evolution of Predictability (PRoPHET) in the Cluster Scenario

algorithm. From the figure, we can see that the predictability metric of the cluster node goes above the predictability of the mule in quite a few instances. This leads to situations where, even though the cluster node will never be able to deliver a packet to the destination, it will incorrectly decide not to forward to the mule.

B. Choice of Parameters β and γ

The most prominent contribution of the HERA algorithm is the flexibility it provides to the user to choose the impact transitivity should play in routing. Because of this flexibility, it important to understand some general guidelines in choosing the appropriate parameters and how to match those parameters to the networks in which they will be used.

First, consider the parameter $\boldsymbol{\beta}$, which is used to update the reachability metrics every time a encounter occurs. Compare the best-performing values of $\boldsymbol{\beta}$ for the Cluster and Taxi Scenarios. For the Cluster scenario we have $\boldsymbol{\beta_C} = [0.2, 0.02, 0.02]$ and for the Taxi scenario $\boldsymbol{\beta_T} = [0.9, 0.9, 0.9, 0.9]$. The discrepancy between the magnitudes of the values is a reflection of the effects of the total number of encounters. In the Cluster scenario, there is a large number of encounters among cluster nodes while the Taxi scenario provides a smaller number of encounters. Therefore the Taxi γ parameter must be larger than the one for the Cluster scenario. The other point to consider is the relationship between the different β_h. In the Cluster scenario, we want to "trust" direct contacts with the destination and minimize the contributions of intra-cluster encounters, so we use an order of magnitude decrease in the values as h increases. In the Taxi scenario, the network is more homogenous and so the values chosen for $\boldsymbol{\beta}$ are kept uniform.

Now, consider the other parameter in our algorithm, γ. The parameter γ is related to the length of possible routing paths and how much they should contribute to the reachability metric. Once again, compare the best-performing values of γ for the Cluster and Taxi Scenarios. For the Cluster scenario we have $\boldsymbol{\gamma_C} = [0.5, 0.05, 0.005]$ and for the Taxi scenario $\boldsymbol{\gamma_T} = [0.25, 0.25, 0.125, 0.125]$. For this parameter, the magnitude of the values is related to the number of possible paths we would like to maintain contributing to the destination reachability metric. In the Taxi scenario, there are many possible equal paths, and therefore maintaining too many of them causes lower performance. In the Cluster scenario, there are only a few possible paths, and so we aim to maintain them with a generally higher value of γ. The effects of the relationship between the different values of β_h are more clearly defined. In the Cluster scenario, the network characteristics let us know that longer paths should be significantly undermined, and thus we use a order of magnitude decrease in value. On the other hand, in the Taxi scenario, longer paths are needed to maintain a good delivery ratio, an so the decrease of value to longer paths in only by a factor of 2.

C. Effects of the value of H

Another parameter choice HERA presents is the value of H. A larger value of H results in HERA adding contributions from longer possible paths. In other words, a larger H represents more propagated effects of transitivity. It is important to keep in mind that the higher the value of H, the more routing information needed to be exchanged since the Reachability Matrix each node maintains, and needs to exchange, is of size $(H+1)\times N$. A simple design guideline is to use the smallest H that achieves the desired performance. Figure 7 show a typical simulation result for the Cluster scenario using different values of H.

The results show that larger values of H result in a decrease of performance in terms of delivery ratio and number of replicas. While the results show that there is a gain to be obtained in terms of latency, it does come at a cost of the other two metrics. Therefore, prolonging the effects of transitivity provide a tradeoff and so a network user might want to use a small H to reduce the routing information overhead. Of course, the performance is network dependent and if a network requires longer paths than the Cluster scenario to achieve successful deliveries, the appropriate H might need to be larger. In general, the guidelines described provide a starting point in choosing the appropriate parameters for a given network. As each network is unique, HERA provides the flexibility to experiment with different parameters until the best matching parameters for a given network are found.

VII. Conclusion

In this work, we have described and analyzed the Hop Expansion Routing Algorithm (HERA) which provides extensive flexibility to control the effects of transitivity in DTN

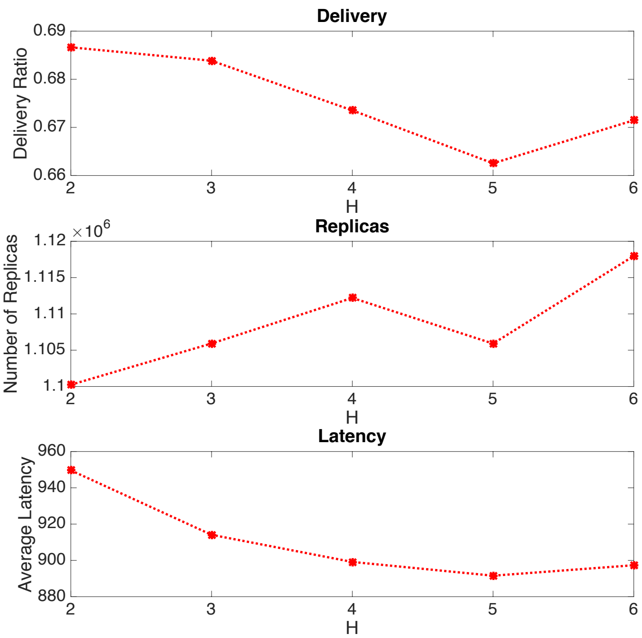

Fig. 7: Effects of H on Routing Performance

routing. We have evaluated delivery ratio, number of replicas, and average latency of our proposed algorithms and have shown that HERA outperforms PRoPHET in three different mobility scenarios, which include simulated scenarios and real-life traces. Not only does HERA outperform PRoPHET, but it provides parameter control to operate at different points in the tradeoff space between these three performance metrics. We have also provided general guidelines on the choice of the HERA parameters and how to appropriately match them to the network characteristics.

References

[1] A. Lindgren, A. Doria, and O. Schelen, "Probabilistic routing in intermittently connected networks", in *SIGMOBILE*, pp. 19-20, 2003.

[2] S. Grasic, E. Davies, A. Lindgren, and A. Doria, "The evolution of a DTN routing protocol - PRoPHETv2", in *Proceedings of the Workshop on Challenged networks (CHANTS '11)*, New York, pp. 27-30, 2011.

[3] T. K. Huang, C. K. Lee and L. J. Chen, "PRoPHET+: An Adaptive PRoPHET-Based Routing Protocol for Opportunistic Network," *2010 24th IEEE International Conference on Advanced Information Networking and Applications*, Perth, WA, pp. 112-119, 2010.

[4] A. Kernen, J. Ott and T. Krkkinen, "The ONE Simulator for DTN Protocol Evaluation", *SIMUTools'09: 2nd International Conference on Simulation Tools and Techniques*, Rome, March 2009.

[5] M. Piorkowski, N. Sarafijanovic-Djukic, M. Grossglauser, CRAWDAD dataset epfl/mobility (v. 20090224), downloaded from https://crawdad.org/epfl/mobility/20090224, https://doi.org/10.15783/C7J010, Feb 2009.

[6] F. Ekman, A. Kernen, J. Karvo, and J. Ott, "Working day movement model", in *Proceedings of the SIGMOBILE Workshop on Mobility models (MobilityModels '08)*, New York, NY, USA, pp. 33-40, 2008.