

# TTN: A *Time-to-Network* Approach to Data Reporting in Mobile Ad Hoc Networks

Phillip Sitbon, Wu-chi Feng, Nirupama Bulusu  
Portland State University  
Department of Computer Science  
E-mail: {sitbon,wuchi,nbulusu}@cs.pdx.edu

**Abstract**—For many applications in mobile wireless ad hoc networks (MANETs), forming an end-to-end data path is not always necessary; instead, the primary routing goal is often data collection or dissemination where only a data source is known. Routing algorithms must be carefully chosen in order to suit the needs of applications employing them. Our focus is on data collection applications in MANETs where limited mobility information is required to route data in a scalable manner. To address this goal, we employ the concepts of delay-tolerant networking (DTN) in which data makes progress toward a destination with high latency expectations and little knowledge of routing topology. Specifically, we present *time-to-network (TTN) forwarding*, a method of forwarding data generated by mobile nodes to a network endpoint in such a way that delivery latency is lowered without high networking cost. By segmenting mobility patterns into *trips*, we are able to apply TTN to a vehicular network using only an estimated destination arrival time for each vehicle. We evaluate TTN using mobility data from the TRANSIMS simulator for a real road network. Results show that our algorithm produces collection-to-network latencies similar to more generic algorithms but at a lower cost and with higher efficiency. Furthermore, we establish a lower bound for delivery latency in our experiments and compare it to TTN. This also helps normalize the interpretation of results specific to our mobility model.

## I. INTRODUCTION

Highly mobile ad hoc networks (MANETs) often suffer from large variations in connectivity over time. For applications within such types of networks, two-way or connected communication is not always necessary and often not possible. For example, in a wireless sensor network, eventual delivery to a sink node may be the only messaging requirement; on the other hand, an emergency notification application in a vehicular network may need to disseminate alerts among vehicles as quickly as possible. Rather than focusing on end-to-end communication where data has a source and specific destination, such applications have more generic goals. *Delay-tolerant networking (DTN)* [1] concepts provide a *store-carry-forward* approach to communicating in which peers opportunistically communicate with each other, and often provides a more robust method of delivery at the cost of redundancy and delay.

In broad terms, there are two extremes to collecting and reporting data in a MANET. First, mobile agents can collect

data and report only to a central location (such as when a vehicle destination is reached). In this case, network efficiency is high because data is never duplicated, but latency between sensing and reporting is also high due to limited connectivity. On the other hand, mobile agents can additionally report data to each other as often as possible in the hopes of reducing latency. Data duplication and network bandwidth are typically high in this case, thus lowering efficiency; however, overall latency can be much lower because data is reported sooner. The challenge in protocol design is to achieve the best of both high network efficiency and low reporting latency.

In this work, we propose a new method of forwarding data in mobile ad hoc networks utilizing a *time-to-network (TTN)* algorithm. First, we segment mobility into *trips* with waypoints indicating a beginning and end time and position for each. By considering each trip endpoint as an Internet-connected upload point (which could very well be an access point encountered during a longer trip), TTN uses metrics related to these trips in order to minimize the time between data collection and reporting. The goal of TTN is to reduce this latency of data collection, such that it can be made available to the Internet as soon as possible. Additionally, we aim to reduce the networking overhead incurred to achieve this goal. One important feature of TTN is the interchangeability of trip metrics; for example, an estimation of trip end times can be used to determine upload availability, or the amount of time spent without encountering an access point. This creates an opportunity to take advantage of in-vehicle GPS devices a driver might use for trip planning. For this work, we only assume the availability of estimated trip end times and leave other metrics for future work. Additionally, Internet-connected vehicles can act as mobile upload points, such that network availability shared during a trip can indicate the immediate availability of a data reporting sink; in this work, we focus on scenarios without mobile infrastructure and hope to revisit the topic in future work.

To evaluate TTN, we simulate its operation within a large-scale vehicular network with vehicle mobility data generated by the TRANSIMS simulator [2]. In the default case, TTN assumes that knowledge of vehicle mobility is limited to an estimated destination arrival time (which TRANSIMS provides) and uses this value to determine when and where to forward data. We compare TTN to epidemic and single-

hop forwarding methods and evaluate the efficiency of each — this helps us determine how well our algorithm performs compared to near-worst case scenarios. Results indicate that TTN provides higher data efficiency for the networking cost incurred. Additionally, we establish a lower bound for data reporting latency (using total knowledge of mobility and connectivity) and compare it to TTN, both for the best data paths TTN can choose (described later) and the overall optimal path from source to sink. Our findings show that (1) the optimal path requires too many hops to provide optimal results when simulated in the network, and (2) TTN performs within a reasonable margin of its optimal behavior, indicating that the network is not under high load.

This paper is structured as follows: in Section II, we discuss related work. In Section III, we describe TTN and our mobility scenario. In Section IV, we propose metrics for evaluating TTN and provide our simulation results. We conclude in Section V.

## II. RELATED WORK

As Jain et al. showed, the level of knowledge about mobility patterns increases routing efficiency [3]. They implement a partial-knowledge routing algorithm based on Dijkstra’s algorithm in order to determine a shortest-weighted path. Merugu et al. developed a space-time routing method utilizing predictable mobility in order to achieve an optimal path [4]. Their approach allows them to calculate routing parameters based on future delivery goals, but determining the goals require information about future mobility and delivery times. Dubois-Ferriere et al. implement an age-based encounter method for route discovery without mobility information [5]. Liu & Wu take hierarchical approach to DTN routing, which approximates an optimal space-time route in terms of delay and hop-count [6]. Rather than performing route discovery, Vahdat et al. implement an *epidemic* routing algorithm that uses intermediate storage and message duplication to ensure eventual delivery [7]. While these solutions offer effective routing, they either require near-exact mobility information or do not focus on mobility as it relates to vehicles specifically. In this work, we primarily consider only the availability of destination arrival times in the context of a vehicle network.

Different DTN forwarding methods have been created according to specific communication needs. Many methods focus on end-to-end and bidirectional communication, with goals similar to MANET routing solutions such as AODV [8] and GPSR [9]. Some methods focus more on getting data from one point to another, while making the return path a separate consideration under the same algorithm [5]–[7]. Leontiadis et al. describe an interest-based method of opportunistically disseminating data to vehicles and show that a publish-subscribe mechanism works well in such a scenario [10]. Zhao & Cao describe a method (called VADD) specifically for delivering data from vehicles to access points [11]. Our work is most similar to VADD; however, our forwarding method requires less mobility information and data destinations always correspond to vehicle destinations.

The Transportation Analysis and Simulation System (TRANSIMS), an integrated system of travel forecasting tools [2], caught the attention of MANET researchers despite its focus on transportation planning. Engelhart et al. [12] compared network measurements among TRANSIMS and variations of the random waypoint model in order to quantify the differences, finding that the naturally uneven spatial distribution of nodes set it apart even from more advanced derivations of the random waypoint model. Marfia et al. [13] analyze the performance of AODV in Portland with TRANSIMS, but their network simulator limited their time span to 200 seconds. In this work, we use TRANSIMS data for a similar-sized area but over a longer time period. Also, unlike many other TRANSIMS users, our aim is to apply delay-tolerant communication techniques to communication in vehicular ad hoc networks.

## III. APPROACH

Mobile wireless networks often suffer from frequent disconnection, and the availability of infrastructure cannot always be guaranteed or affordable. In a data collection and reporting scenario, it is possible to delay the availability of recorded information until communication opportunities arise; however, many types of data lose utility as they get older. For example, traffic data delayed by even 15 minutes can be irrelevant to those about to take a specific route, because in that amount of time congestion levels change significantly — especially during crucial peak hours. The goal of TTN is to achieve high data availability (low latency) in such scenarios without congesting wireless channels. There are many solutions in between the approaches of minimizing latency and minimizing network usage, and we believe TTN falls very close to the middle of these two extremes.

The first step of our solution is to define mobility characteristics inherent to many MANETs. Mobility patterns can be segmented into *trips* in such a way that their beginning and end times indicate changes in movement or arrival at waypoints. In this way, each discrete trip provides metrics upon which routing decisions can be made; namely, their start and end times. The imposed knowledge requirements are minimal, and meaningful routing decisions can be made depending on the application. In the case of a distributed data collection application, where nodes are moving on a map and must cooperatively provide data to a central location, trip end times can correspond to connectivity with the location. For our work we focus on such an application with trip end times corresponding to Internet-connected locations at which data is reported to. This does not exclude the possibility of roadside access points; rather, trips are segmented in such a way that an access point corresponds to the end of one trip and the beginning of another.

In the rest of this section, we first outline our assumptions that influenced the design of TTN in the context of vehicular networks, followed by a description of the mobility system we chose. We then describe the basic forwarding methods we will compare to TTN as well as a description of our algorithm.

### A. Assumptions

Many variables related to networking and mobility have a significant impact on the perceived performance of certain communication protocols. In addition to applying TTN specifically to vehicle networks, our approach includes some assumptions that we believe are either common and acceptable cases, or exclusionary of issues irrelevant to the performance of TTN:

- 1) Vehicles collect data while traveling on the road and this data is most useful when made available soon after collection,
- 2) Each networked vehicle is able to estimate the time it will reach a destination, and
- 3) Vehicle destinations are defined as points where collected data can be uploaded to the Internet, although need not be the actual trip endpoints.
- 4) Once data reaches a destination, reporting (*e.g.* uploading) delay is negligible compared to the time spent within the vehicle network.

When vehicles collect data, each discrete data item is marked with a *recording time* in order to identify its age. This information is passed along with the actual data when forwarded to other vehicles. While assumption (3) may be somewhat strong, we leave any alternatives for future work (such as the explicit availability of roadside access points).

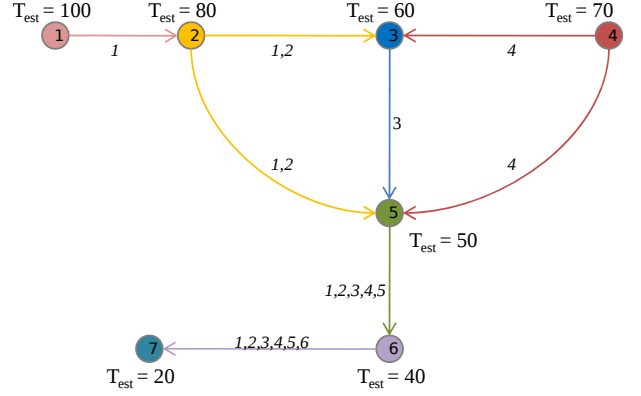
### B. Forwarding Parameters

Opportunistic communication can be effective toward minimizing latency in a delay-tolerant network, provided that forwarding choices are made wisely. Because we assume each vehicle will be able to deliver data once its destination is reached, our goal is to make forwarding choices based on parameters representing progress toward each destination. For our TTN forwarding algorithm, we outline three separate *T parameters* for each trip that might be available to the network layer:

- $T_{est}$ : The estimated time-to-network
- $T_{act}$ : The actual time-to-network
- $T_{start}$ : The starting time for each trip

Data will be opportunistically sent in one direction between each pair of vehicles depending on the difference of their *T* parameters. For example, when filtering based on  $T_{est}$ , if  $T_{est} = 300s$  for vehicle *A* and  $T_{est} = 600s$  for vehicle *B*, data will flow from vehicle *B* to vehicle *A*. We choose  $T_{start}$  as a *T* parameter under the assumption that vehicles traveling for a longer period of time might statistically be closer to their destinations and thus should receive data from vehicles with a later  $T_{start}$ . Clearly, knowing end times and estimating them accurately can play an important role in performance. For this work, we focus only on  $T_{est}$  as an available parameter for forwarding decisions and leave any study of its impact (versus the use of  $T_{act}$ ) for later work.

Due to the way TTN makes decisions based on one of the *T* parameters, data items will never loop back to any vehicles already carrying them. This only holds true if the *T* parameters



**Fig. 1:** Example data path among wireless nodes given by their connectivity and  $T_{est}$  values.

do not change for the duration of a trip—while it may be beneficial to update the value of  $T_{est}$  during a trip based on current traffic conditions, we leave such possibilities for future consideration. Additionally, the probability that a data item will be forwarded additional hops is monotonically decreasing for each data path, such that a data item is less likely to be forwarded additional hops as its carrier gets closer to the end of the trip. These properties contribute to the scalability of TTN by limiting network usage.

### C. TTN Algorithm

The TTN forwarding algorithm performs pairwise transfers based on *T* parameter comparisons. Every second during a trip, a vehicle will send a UDP beacon containing its *T* parameter value. Beacon receivers compare the value to their own and request data only if the beacon’s *T* parameter value is later (*e.g.* a beacon receiver is ending its trip sooner); otherwise, the beacon receivers will send up to two data items to the beacon source (priority is described below), one collected locally and one forwarded item. Figure 1 shows an example data path based on TTN values of each node, and Algorithm 1 outlines the general rules that dictate beacon handling.

Upon receiving a data item, an acknowledgment message is returned indicating the recording time ranges already carried for the data sources. Acknowledgments are responded to with remaining data items if any are available to send. Repeated pairwise encounters will lead to some duplicate data transmission, but peers will be able to pick up where they left off during the previous encounter.

Data is prioritized by age. The longer a data item exists, the more likely it is to be forwarded ahead of other data, provided there is an expected reduction in latency. Part of our overall goal is to minimize time-to-network, which was our motivation for this choice originally; however, there is a possibility that reconstructing data from a larger time span (*e.g.* by sending one old and one new item, and filling in between where possible) might provide more meaningful data to an application. Also, forwarding older data items first may

---

**Algorithm 1** Pseudo code for high-level beacon handling.

---

```
let LocalData( $V$ ) = Data collected by vehicle  $V$ 
let OtherData( $V$ ) = Forwarded data carried by  $V$ 
let LastSentTo( $x$ ) = Last vehicle data item  $x$  sent to

for each beacon  $M_b$  received from vehicle  $V_s$  at vehicle  $V_d$ 
do
  if  $T_{est}(V_s) < T_{est}(V_d)$  then
    if LocalData( $V_d$ )  $\neq \emptyset$  then
      for  $L_i$  in LocalData( $V_d$ ) do
        if (LastSent( $L_i$ )  $>$  SendWait)  $\wedge$ 
          ( $T_{est}(LastSentTo(L_i)) > T_{est}(V_d)$ ) then
          Send  $L_i$  to  $V_s$ 
          break
        end if
      end for
    end if
  end for

  if OtherData( $V_d$ )  $\neq \emptyset$  then
    for  $O_i$  in OtherData( $V_d$ ) do
      let  $V_{O_i}$  = Original data source
      if (LastSent( $O_i$ )  $>$  SendWait)  $\wedge$ 
        ( $T_{est}(LastSentTo(O_i)) > T_{est}(V_{O_i})$ ) then
        Send  $O_i$  to  $V_s$ 
        break
      end if
    end for
  end if
end for
```

---

not provide a greater average improvement over sending any random item first where improvement is expected. We leave such possibilities for investigation in future work.

In order to limit network congestion and to spread data evenly among peers, data items are sent no more than once every 3 seconds; this value corresponds to *SendWait* in Algorithm 1. The primary drawback of such a restriction is the possibility of missing large improvements — our subjective experience with this parameter have shown 3 seconds to provide only a small number of missed forwarding opportunities while ensuring reduced network congestion.

#### D. Epidemic & Single-hop Flooding

In order to compare TTN to other common forwarding methods, we remove T parameter filtering and implement more basic forwarding rules. Specifically, we first limit all communication to a single hop, and data is forwarded between each pair of nodes when within radio range. For the multi-hop (epidemic) case, data is forwarded multiple times by previous receivers (with the same priority as TTN). In order to avoid stale data in the network when forwarding epidemically, data items expire after 10 minutes.

#### E. Optimal Path Routing

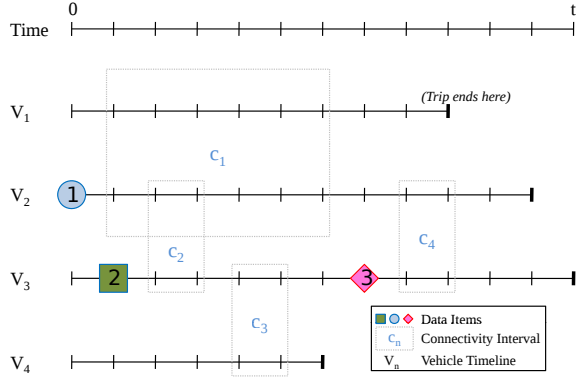
In our experiments, we also consider optimal routing of data. For TTN, the overall goal is to minimize latency, so we use that as our metric of optimality. Although such data paths may not be available in reality, we can simulate them in order to observe a lower bound for latency. To generate an optimal path for each time interval of a vehicle's trip, every possible encounter with other vehicles is evaluated starting at the end of the trip. Here, we consider two variations of the optimal latency: the optimal path along the time-varying graph based from  $T_{act}$ , and the optimal performance of TTN base from  $T_{est}$  with the additional per-hop forwarding restrictions imposed by the algorithm. These optimal data paths, when applied to a wireless simulation, allow us to establish a *simulated* lower bound on our results in the context of our specific mobility data.

For both calculations, the minimum total path weight is chosen as the best path up to that point in time for the chosen vehicle, which is defined by the minimum time data forwarded during that encounter will be reported. Note that data recorded after the availability of this optimal path cannot be forwarded, hence the need for time intervals to define each possible path. In order to avoid an instantaneous path that would never exist in a real network (e.g. data traversing 10-20 hops in one short moment of total connectivity), communication latencies between vehicles are included according to their distance. Furthermore, if two paths have the same weight, the path with fewer hops is chosen.

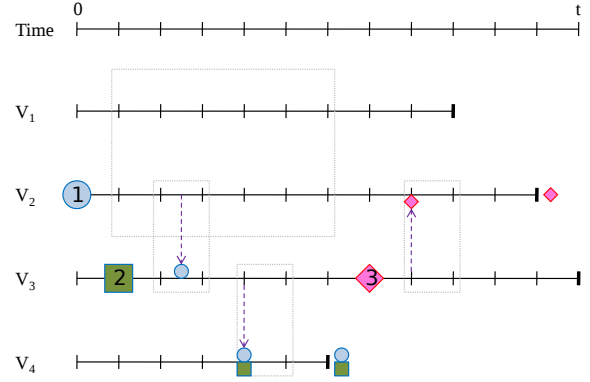
Because the optimal path is expected to change over time, we apply a modified version of the Floyd-Warshall algorithm to each time step of the simulation, starting at the end and iterating down to the first simulation time step, with the initial path latency for each vehicle set to its trip end time ( $T_{act}$ ). This allows the addition of a conditional check on each path, so that data will only be forwarded along a path at time  $t$  with latency  $l_1$  if, at time  $t + 1$  with path latency  $l_2$ , the condition  $l_1 < l_2$  is true. As noted in the previous paragraph, reverse iteration is necessary because data may not exist until the time interval between  $t$  and  $t + 1$ , and thus the path at time  $t$  cannot be taken advantage of in order to improve that data.

Finally, we impose further restrictions for separate experiments to model the optimal behavior of the TTN algorithm. The initial path weight of each vehicle is set to its  $T_{est}$  value rather than  $T_{act}$ , and no single hop that does not reduce data latency is chosen. In the optimal case, data may follow paths via vehicles with a later  $T_{act}$  value before reaching the vehicle that provides a minimal latency. For the optimal TTN implementation, such paths are excluded.

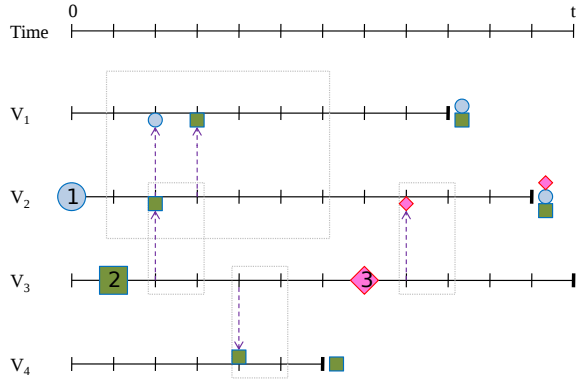
Figure 2 compares the behavior of TTN (2(c)) to the optimal (2(b)) and TTN optimal (2(d)) paths. Figure 2(a) describes the trips of each vehicle ( $V_n$ ) and connectivity between pairs of vehicles ( $c_m$ ) throughout these trips. Three data items (no. 1 for  $V_2$ , 2 and 3 on  $V_3$ ) are generated at the times marked by the numbered symbols and smaller symbols attached to arrows indicate the transmission of data. Symbols appearing after end



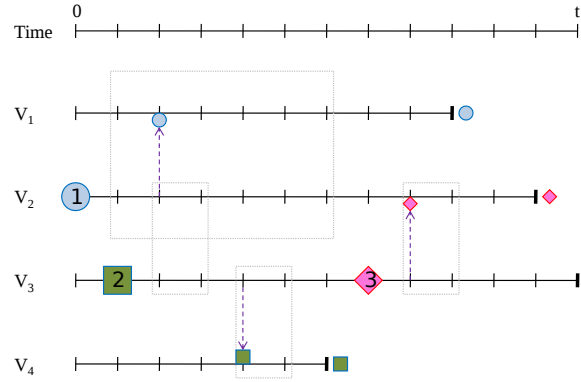
(a) Vehicle connectivity. Each box ( $c_m$ ) indicates time intervals that vehicles are able to communicate.



(b) Optimal. Here, data item 1 traverses a path that first increases reporting latency in order to obtain the minimum latency provided by  $V_4$ .



(c) TTN. This is the actual behavior of the TTN algorithm — data item 2 is sent to  $V_4$  after  $V_2$  because  $T_{est}(V_4) < T_{est}(V_2)$ .



(d) TTN Optimal. With global information, it is not necessary for  $V_3$  to send data item 2 to  $V_2$ .

**Fig. 2:** Message forwarding scenarios for three specific data items.

mark of a vehicle trip indicate that the corresponding data item was carried and subsequently reported by that vehicle. To simplify this example, we assume  $T_{est} = T_{end}$ . The difference between the optimal case and TTN for data item 1 (circle) illustrates the sub-optimality of TTN—it will never be forwarded to  $V_4$  via  $V_3$  because  $T_{est}(V_3) > T_{est}(V_2)$ , even if  $V_4$  provides the lowest latency for that data item.

Figures 2(b) and 2(d) illustrate data not being retained by their originating vehicles. For an optimal scenario, data retention is not necessary; however, when we implement these data paths in the wireless simulator, data items are kept (and not retransmitted) in case they do not end up traversing the planned path (*e.g.* due to loss). In the presence of non-infinite bandwidth (such as in a wireless simulator), finding new paths for delayed or dropped data is an NP-complete problem, and so our approach is to only remove the possibility of negative improvement.

## IV. EVALUATION

Here, we outline our experimental setup, provide a description of our metrics, and evaluate the performance of our forwarding method compared first to epidemic routing and single-hop flooding, then to optimal forwarding methods.

### A. Experimental Setup

1) *Network & Mobility:* In order to evaluate TTN, we chose the GTNetS [14] network simulator, which supports the 802.11 MAC [15]. We chose GTNetS primarily for its mobility support and 802.11b due to its wide use in simulation and real deployments. The beacon and acknowledge message sizes are 24 bytes, each data message is 512 bytes, and data is generated by selected vehicles at a rate of one data item per second (except where we evaluate different data collection rates). All vehicles have a uniform transmission range of 100m.

For our experiments, we chose the *Traffic Analysis and Simulation System (TRANSIMS)*. The goal of TRANSIMS is to provide detailed spatial and temporal simulation of travel conditions to facilitate the effective analysis of transportation

systems. The process of simulating vehicles starts with generating a synthetic population including household size, age, and income based from recent US census data. Based on this population and sets of activities, travel plans are generated and fed into the traffic microsimulator, where trips are executed concurrently within a cellular automata.

2) *Metrics*: Our goal in these experiments is to demonstrate how our TTN algorithm (described in Section III) performs in various scenarios. In order to evaluate TTN’s performance over multiple hops, we chose to compare it to the epidemic forwarding method (also described in Section III). We also compare TTN to a single-hop flooding approach. We employ the following metrics in order to describe our results:

- **Latency**: The amount of time taken to report data at a vehicle destination, measured from the time recorded,
- **Improvement**: The amount of latency reduced by forwarding data to vehicles that reach their destinations sooner,
- **Efficiency**: The improvement achieved per message copy forwarded, and
- **Overhead**: The number of duplicate messages.

In our simulations, the average data latency ranges from 10 to 15 minutes when no networking is considered, corresponding to the average trip lengths for the vehicle subsets chosen.

### B. Varying Networking & Collection Densities

In this experiment, we vary the number of vehicles networking and collecting data by selecting different random subsets for multiple simulation runs. Our goal here is to understand the behavior of TTN in situations with varied data densities and varied network availability, which represents scalability across many parameters. Table I outlines the number of vehicles participating in each role for the simulation runs.

| Percent | Varying             |              |                     |              |
|---------|---------------------|--------------|---------------------|--------------|
|         | Vehicles Collecting |              | Vehicles Networking |              |
|         | # Collecting        | # Networking | # Collecting        | # Networking |
| 25%     | 1177                | 4708         | 1177                | 1177         |
| 50%     | 2354                | 4708         | 2354                | 2354         |
| 75%     | 3531                | 4708         | 3531                | 3531         |
| 100%    | 4708                | 4708         | 4708                | 4708         |

TABLE I: Vehicle counts for different experiments.

With all vehicles networking, Figure 3 shows the average message latency when we select 25%, 50%, and 100% of all vehicles to participate in the collection of data. Figure 5 shows the average message latency when 25%, 50%, and 100% of all vehicles are networking, with all vehicles in each participating in data collection.

Figures 4 and 6 show the average number of copies per message for the entire simulation as an indicator of the amount of data duplication required in order to achieve the latencies shown.

As expected, TTN provides lower latencies than epidemic and single-hop forwarding on average. This corresponds to both its uni-directional behavior and filtering performed, but a contributing factor is also the lower network usage. Epidemic

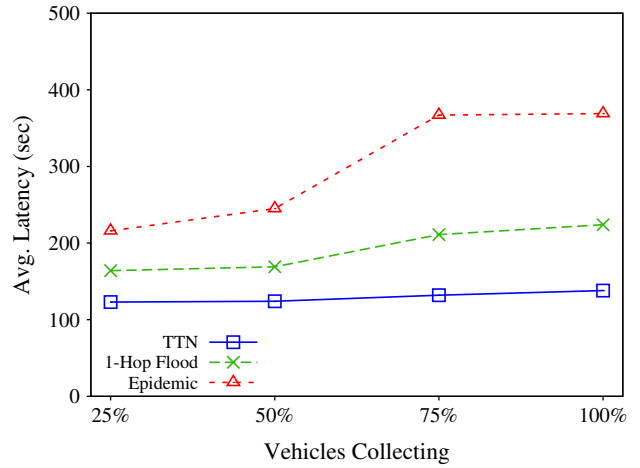


Fig. 3: Average message latency when varying the number of vehicles collecting data.

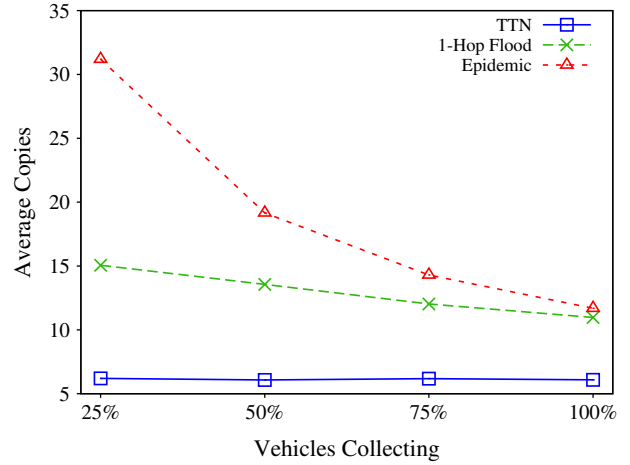


Fig. 4: Average number of message copies when varying the number of vehicles collecting data.

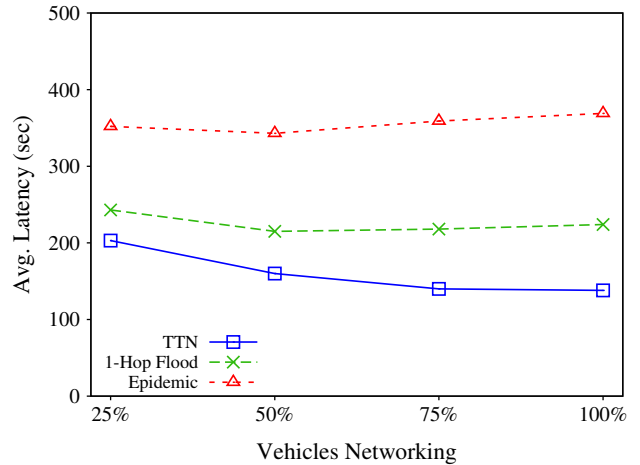
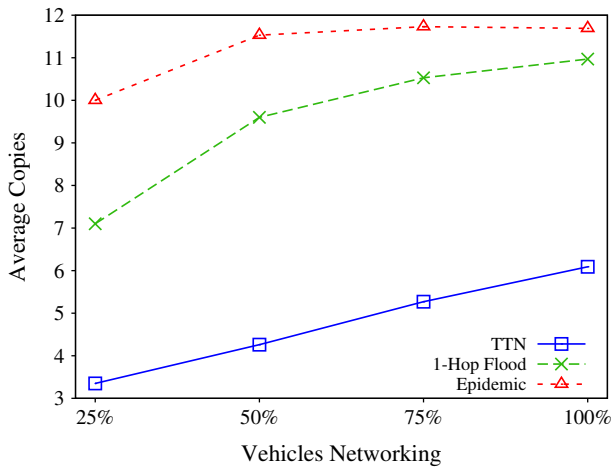
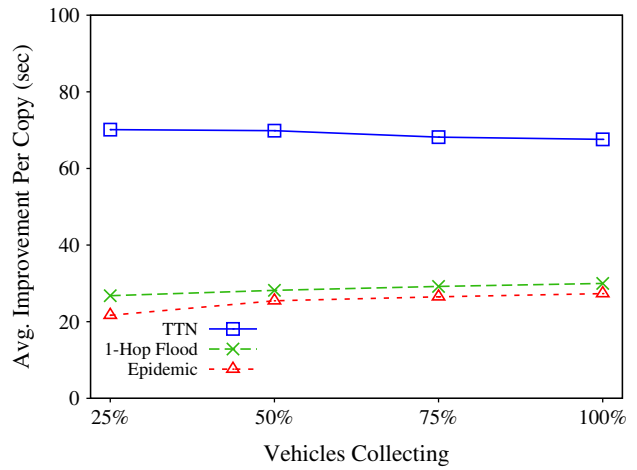


Fig. 5: Average message latency when varying the number of vehicles networking.

forwarding allows data to extend many more hops away from its source, but such behavior does not contribute to



**Fig. 6:** Average number of message copies when varying the number of vehicles networking.



**Fig. 7:** Average efficiency when varying the number of vehicles collecting data.

lowering latencies; essentially, many of the multihop copies of the data are noise and eventually expire. Both graphs show better scalability for TTN, and it appears to scale better as the number of vehicles collecting data approaches 100%. As more data collection vehicles are introduced, network congestion and higher data volumes cause 1-hop flooding and epidemic forwarding to produce less copies per message on average, while TTN evenly distributes messages with available bandwidth. The increase in average copies when varying the number of vehicles networking illustrates the increase in forwarding opportunities, and its slope is less steep for TTN due to filtering and choosing only some of the additional data paths.

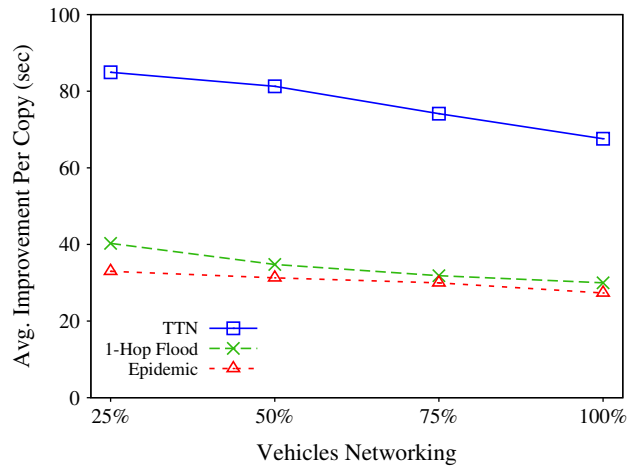
In the next subsection, we use the same varied densities and look at the efficiency of each.

### C. Efficiency and Overhead

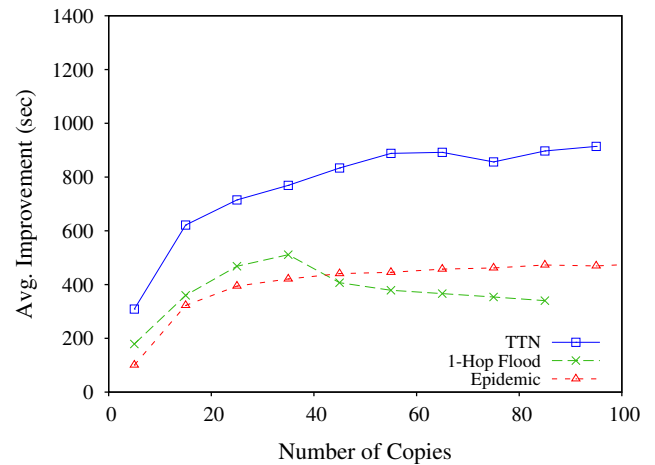
Single-hop flooding and epidemic forwarding exhibit similar efficiency. In the epidemic case, higher latencies are caused both by network congestion and the propagation of data that is not improved. Figures 7 and 8 show the average efficiency (improvement per message copy) different collection and networking densities.

Figures 7 and 8 indirectly show that TTN forwarding provides the lowest average latency but is also capable of providing a lower latency per message copy thus increasing overall efficiency. Efficiency appears scalable for 1-hop flooding and epidemic forwarding as well, but the reasons for similar efficiencies among the experiments is the relationship between latency and message copies; therefore, the most useful detail gleaned from these graphs is the higher efficiency of TTN.

For the simulations including all vehicles networking and collecting data, Figure 9 shows the improvement (reduction in latency) as it relates to the number of copies of each data item and Figure 10 shows the total number of duplicate messages for each forwarding method throughout the simulation.



**Fig. 8:** Average efficiency when varying the number of vehicles networking.



**Fig. 9:** Average improvement (reduction in latency) for a given number of copies with all vehicles collecting data and networking.

Figure 9 shows that TTN provides a clear benefit for each data copy generated. This is partially true for 1-hop flooding



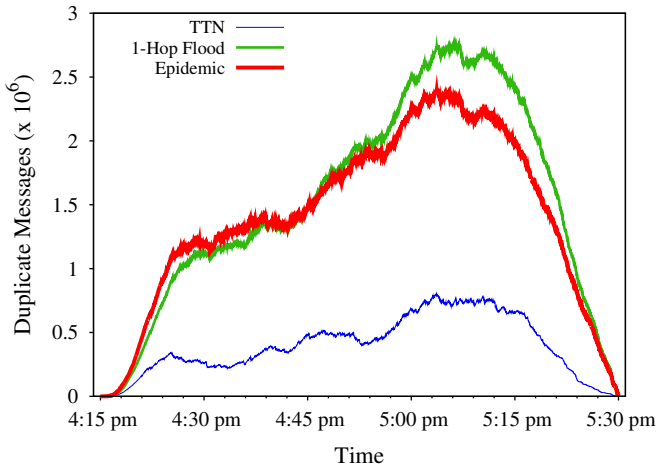


Fig. 10: Number of duplicate messages throughout the simulation.

and epidemic forwarding as well, but the benefit is lower and falls off as the number of copies increases. Epidemic forwarding, specifically, produces messages with many more than the 100 copies shown, but the improvement value tapers off in a way that indicates a weaker relationship between improvement and message copies.

Figure 10 shows the total amount of duplicate data in the network throughout the simulation. TTNs duplicate data is much lower primarily due to the fact that data transfer between any two vehicles is primarily one-way (depending on the difference between their  $T_{est}$  parameters), while single-hop and epidemic send data in both directions where possible. During the highest-volume simulation times, duplicate data in the single-hop experiment surpasses that of epidemic due to the requirement that epidemic forwarding method drop data items older than 10 minutes to ensure that data leaves the network at some point rather than being continuously recirculated.

#### D. Optimum Performance

Here we investigate the performance of TTN compared to the optimal routing solution for our mobility model, discussed in Section III-E. We consider two optimums: first, we investigate the optimum path representing the absolute minimum latency achievable given the mobility data used in all of our experiments; second, we evaluate the optimum performance of the TTN algorithm that consists of the additional requirement that each hop reduces the latency of data sent across it (according to the  $T_{est}$  parameter, keeping results in line with our other simulations).

In order to understand the effect of our network simulation on these optimal paths, we also run the network simulation in such a way that vehicles only communicate with others when it is optimal to do so. Any data not transmitted during an optimal encounter is held until the next, and otherwise no more forwarding is done. In order to avoid negative improvement due to path breaking, and to behave more like TTN, a copy of each data item is held at each hop along an optimal path,

therefore providing zero latency reduction in the worst case.

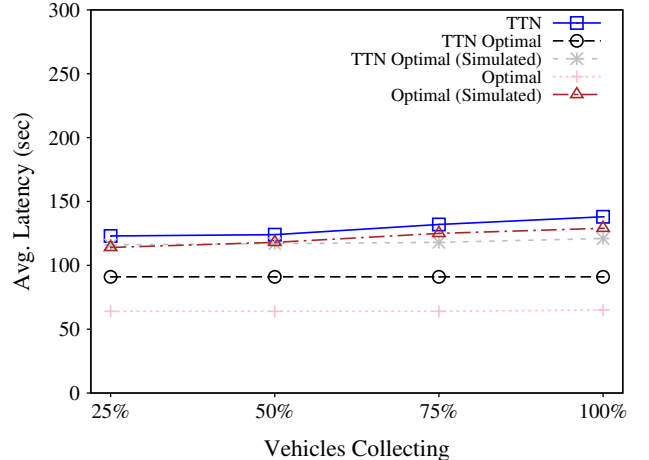


Fig. 11: Comparison of TTN to optimal routing schemes when varying the number of vehicles collecting data.

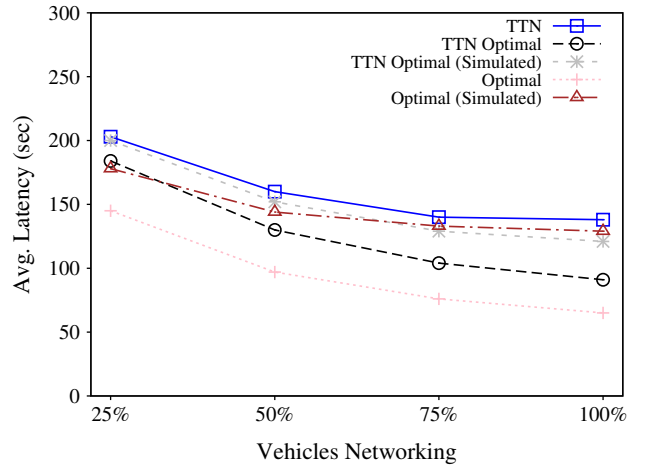


Fig. 12: Comparison of TTN to optimal routing schemes when varying the number of vehicles networking.

Simulating optimal routing provides a performance baseline for TTN. Figures 11 and 12 show that, while the optimal routes provide a much lower latency, the difference between TTN and the optimal routes is relatively small in practice when network performance is taken into consideration. When varying the number of vehicles collecting data in Figure 11, it is clear that any change in the average reporting latency is only due to the limited time nodes are able to exchange data; in the optimal case, the larger amount of data can be reported along the same optimal paths. Figure 12 shows that introducing more networking vehicles into the simulation allows for more opportunities to decrease latency, both for the optimal and networked cases.

## V. CONCLUSION & FUTURE WORK

In this paper, we proposed a new method of forwarding data in mobile ad hoc networks utilizing a *time-to-network* (TTN) algorithm. We described how we used mobility data from



the TRANSIMS simulator in order to evaluate our proposed algorithm. Our results showed that TTN forwarding with a T parameter corresponding to estimated destination arrival times provided lower reporting latency (30% on average, or up to 5 minutes improvement for a 15 minute trip) and higher network efficiency in terms of goodput and data duplication. Additionally, comparing TTN to its optimal counterpart and the overall optimal results show that TTN performs exceedingly well in terms of the best possible latency reduction. Although this comes at the cost of data duplication, we have demonstrated that TTN is able to scale to higher data densities without too much loss in performance.

For future work, we would like to tune TTN in order to further increase networking efficiency. One possibility we have considered is to determine if sometimes forwarding in the opposite direction (to vehicles with later end times, for example) could possibly decrease the average latency. We would also like to test scalability further—we are currently using less than 5% of the available mobility data from our TRANSIMS database. Finally, we would like to test different MAC-layer protocols in order to determine their impact on TTN.

#### REFERENCES

- [1] K. Fall, "A delay-tolerant network architecture for challenged internets," in *SIGCOMM '03: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*. Karlsruhe, Germany: ACM, Feb. 2003, pp. 27–34.
- [2] Los Alamos National Laboratory, "TRANSIMS: Transportation analysis and simulation system," 2009, <http://transims-opensource.net/>.
- [3] S. Jain, K. Fall, and R. Patra, "Routing in a delay tolerant network," in *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*. Portland, Oregon, USA: ACM, Sep. 2004, pp. 145–158.
- [4] S. Merugu, M. Ammar, and E. Zegura, "Routing in space and time in networks with predictable mobility," Georgia Institute of Technology, Tech. Rep., 2004.
- [5] H. Dubois-Ferriere, M. Grossglauser, and M. Vetterli, "Age matters: efficient route discovery in mobile ad hoc networks using encounter ages," in *MobiHoc '03: Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*. Annapolis, Maryland, USA: ACM, 2003, pp. 257–266.
- [6] C. Liu and J. Wu, "Scalable routing in delay tolerant networks," in *MobiHoc '07: Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*. Montreal, Quebec, Canada: ACM, Sept. 2007, pp. 51–60.
- [7] A. Vahdat and D. Becker, "Epidemic routing for partially-connected ad hoc networks," Duke University, Tech. Rep., 2000.
- [8] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *In Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, 1999, pp. 90–100.
- [9] B. Karp and H. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in *MobiCom '00: Proceedings of the 6th ACM annual international conference on Mobile computing and networking*. Boston, Massachusetts, USA: ACM, Aug. 2000, pp. 243–254.
- [10] I. Leontiadis and C. Mascolo, "Opportunistic spatio-temporal dissemination system for vehicular networks," in *MobiOpp '07: Proceedings of the 1st international MobiSys workshop on Mobile opportunistic networking*. San Juan, Puerto Rico: ACM, Jun. 2007, pp. 39–46.
- [11] J. Zhao and G. Cao, "VADD: Vehicle-assisted data delivery in vehicular ad hoc networks," in *INFOCOM 2006: 25th IEEE International Conference on Computer Communications*, Barcelona, Spain, April 2006, pp. 1–12.
- [12] D. Engelhart, A. Sivasubramaniam, C. Barrett, M. Marathe, J. Smith, and M. Morin, "A spatial analysis of mobility models: application to wireless ad hoc network simulation," in *ANSS-37 2004: Proceedings 37th Annual Simulation Symposium*. Arlington, Virginia, USA: IEEE, Apr. 2004, pp. 35–42.
- [13] G. Marfia, G. Pau, E. D. Sena, E. Giordano, and M. Gerla, "Evaluating vehicle network strategies for downtown Portland: opportunistic infrastructure and the importance of realistic mobility models," in *MobiOpp '07: Proceedings of the 1st international MobiSys workshop on Mobile opportunistic networking*. San Juan, Puerto Rico: ACM, Jun. 2007, pp. 47–51.
- [14] G. F. Riley, "The Georgia Tech network simulator," in *MoMeTools '03: Proceedings of the ACM SIGCOMM workshop on Models, methods and tools for reproducible network research*. Karlsruhe, Germany: ACM, Aug. 2003, pp. 5–12.
- [15] "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications: Higher-speed physical layer extension in the 2.4 GHz band." IEEE Std 802.11b-1999 (R2003), IEEE, Tech. Rep., 2003.
- [16] M. Nekovee and B. Bogason, "Reliable and efficient information dissemination in intermittently connected vehicular adhoc networks," in *VTC 2007: IEEE 65th Vehicular Technology Conference*. Dublin, Ireland: IEEE, Apr. 2007, pp. 2486–2490.
- [17] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "MaxProp: Routing for vehicle-based disruption-tolerant networks," in *INFOCOM 2006: 25th IEEE International Conference on Computer Communications*. Barcelona, Spain: IEEE, April 2006.
- [18] B. Burns, O. Brock, and B. Levine, "Mv routing and capacity building in disruption tolerant networks," in *INFOCOM 2005: 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 1, Miami, FL, USA, March 2005, pp. 398–408.
- [19] M. Grossglauser and D. Tse, "Mobility increases the capacity of ad hoc wireless networks," *IEEE/ACM Transactions on Networking*, vol. 10, no. 4, pp. 477–486, Aug 2002.
- [20] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," *SIGMOBILE Mobile Computer Communication Review*, vol. 7, no. 3, pp. 19–20, 2003.
- [21] V. Naumov, R. Baumann, and T. Gross, "An evaluation of inter-vehicle ad hoc networks based on realistic vehicular traces," in *MobiHoc '06: Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing*. Florence, Italy: ACM, May 2006, pp. 108–119.