TRAFFICINFO: AN ALGORITHM FOR VANET DISSEMINATION OF REAL-TIME TRAFFIC INFORMATION[1]

Ting Zhong[2]
School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, Sichuan, 610054 China

Bo Xu, Piotr Szczurek, Ouri Wolfson
Department of Computer Science, University of Illinois at Chicago
851 S. Morgan St., Chicago, IL 60608, USA

Abstract: In this paper we propose an algorithm for disseminating reports about real-time traffic conditions in vehicular ad-hoc networks (VANET's). Using this method, each vehicle makes local decision on when to disseminate reports, how many to disseminate, and which reports to disseminate. In order to deal with the bandwidth and memory constraints, reports are prioritized in terms of their value, as reflected by supply and demand. We compare by simulations the proposed algorithm with Grassroots, an existing VANET dissemination algorithm. The results show that the proposed algorithm significantly outperforms Grassroots.

Keywords: Floating car data, vehicular ad-hoc networks, real-time traffic information

I. INTRODUCTION

Existing commercial systems for collecting and disseminating traffic information (e.g., Traffic.com [1], GCM Travel [3]) are implemented by a centralized architecture. These systems usually use roadside sensors to get the traffic information and send the information to central servers. Thus, users are able to select alternative routes in order to avoid congested streets. However, these existing systems tend to cover selected highways where speed sensors are deployed, while leaving out a major fraction of roadways. The main factor that prevents these systems from covering the entire road network is the cost involved to deploy sensors to cover the entire road network.

With a GPS receiver and a digital map installed, a vehicle is able to keep track of its location and to determine its travel time for each road segment recently visited. By sharing the travel time information in a peer-to-peer fashion, vehicles can be aware of the traffic situation of the road network. To share the traffic information, wireless communication between vehicles is required. A vehicular ad-hoc network (VANET) is a set of vehicles that communicate via short-range wireless technologies such as IEEE 802.11 and DSRC (Dedicated Short Range Communications). With such communication mechanisms, a vehicle receives traffic information from its neighbors or from remote vehicles by multi-hop transmission relayed by intermediate vehicles. Each vehicle *m* participating in the VANET periodically produces reports regarding the traffic condition it is experiencing. The reports are disseminated by the VANET. If *m* receives a report that indicates a severe congestion on a road segment on *m*'s currently planned route, then *m* may change its route to avoid the congestion. Compared to centralized, the VANET approach has the following advantages:

1. Due to the fact that short-range wireless networks utilize the unlicensed wireless spectrum, communication in a VANET is free. In addition, there is no cost involved in setting up and maintaining the fixed infrastructure server.

[1] Research supported by US NSF Grants DGE-0549489, 0513736, 0326284, and China NSFC 60573129, 60473090.
[2] Ting Zhong is presently doing research in the Department of Computer Science, University of Illinois at Chicago.

2. The VANET approach enables vehicles to collect traffic conditions of both highways and other roadways.

There are two paradigms to conduct reports dissemination in a VANET, namely structured and structureless. In structured dissemination, a routing structure such as a tree is imposed and maintained among the vehicles (e.g., [11]). Structured dissemination may be very ineffective in a highly mobile environment, as the routing structure quickly becomes obsolete (see [10]). For example, in the tree structure proposed in [11], if a parent node moves away, then all of its children nodes are isolated from dissemination.

In structureless dissemination, no routing structures are maintained; the intermediate vehicles save reports to their local databases and later transfer these reports. In the literature this paradigm is also called structureless gossiping, epidemic, or store-and-forward dissemination. The problem with store-and-forward is that the reports that need to be stored and forwarded by a vehicle may exceed its storage and bandwidth capacities. The algorithm we propose in this paper addresses this problem by ranking of reports, so that the most relevant reports are transmitted and saved to the local database.

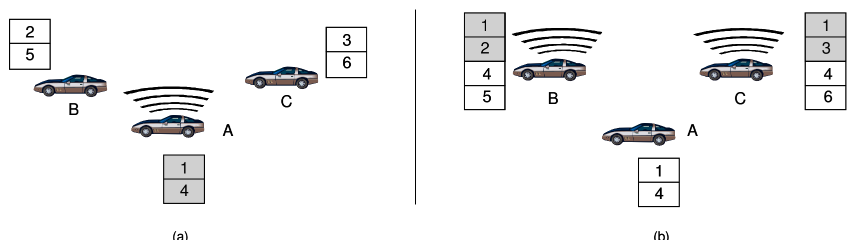

Figure 1.1. Rank-based store-and-forward. (a) Vehicle *A* broadcasts its top two reports which are reports 1 and 4. (b) After receiving from *A*, vehicle *B* incorporates the received reports, re-ranks, and broadcasts the top two (shadowed). The same for *C*.

At a high level, our rank-based store-and-forward method works as follows (Figure 1.1). Each vehicle *O* periodically selects the *k* most relevant reports in its local database, and broadcasts (i.e. forwards) them to its neighbors (i.e. the vehicles that are within the transmission range of *O*). Upon receiving the broadcast from *O*, each neighboring vehicle incorporates the received reports into its own local database, and subsequently broadcasts the top *k* reports. Thus reports transitively spread across vehicles. Similarly, when the local database size is insufficient to store all the received reports, only the most relevant reports are saved.

The fundamental components for the rank-based store-and-forward are the following:

1. How many reports to communicate so that the bandwidth is best utilized? In other words, how should the value of *k* be determined? Observe that if vehicles transmit too much, then many collisions would reduce the number of successfully received reports; and if they transmit too little, report dissemination would suffer. In [10] a formula is developed, with which a vehicle dynamically adjusts the transmission size based on the length of the period of time between subsequent transmissions, such that overall bandwidth/energy efficiency is maximized. The formula is applicable to 802.11 and DSRC.

2. Demand, as introduced in [12], indicates the relevance of the report to vehicles' decision making such as travel-route planning. The higher the demand, the higher the relevance.

3. Supply indicates how many vehicles already have the report. The higher the supply of a report, the lower its relevance. We have developed a machine learning algorithm called MALENA (MAchine LEarning based Novelty rAnking) for the estimation of supply.

In this paper we integrate the above components into the TraffInfo algorithm that ranks reports based on supply and demand, and each transmission communicates the "right" amount of information.

In order to evaluate the proposed TrafficInfo algorithm, we experimentally compared it with Grassroots [5], an existing structureless algorithm for reports dissemination in VANET's. In Grassroots, a report is disseminated by flooding. Specifically, the report is broadcasted by its producer. Each vehicle, on receiving the report for the first time, rebroadcasts it exactly once. The comparison uses the STRAW/SWANS simulation test-bed ([14, 15]). The SWANS (Scalable Wireless Ad hoc Network Simulator) system simulates the detailed procedure and factors of 802.11 communication. The STRAW (STreetRAndom Waypoint) system simulates the vehicle traffic mobility on a region of downtown Chicago.

We propose a novel metric, called the difference in knowledge (DIK), as the performance measure. The DIK measures the difference between the actual traffic condition on a road segment and that known to a vehicle. Intuitively, the nearer a road segment is to a vehicle, the more important its traffic information is to that vehicle. So in our proposed metric, the DIK for a road segment is weighted by the distance of the road segment from the vehicle. In the paper we argue that the DIK combines the two metrics commonly used for data dissemination, namely throughput and response time.

In summary, the main contributions of this paper are as follows. 1) We propose a rank-based store-and-forward algorithm (TrafficInfo) for disseminating the real-time traffic information in a VANET. 2) We compare TrafficInfo with an existing VANET dissemination algorithm. 3) The comparison uses a novel performance metric which combines the throughput and response time metrics.

The rest of the paper is organized as follows. Section II introduces the model. Section III describes the TrafficInfo algorithm. Section IV compares TrafficInfo with Grassroots. Section V discusses relevant work, and section VI concludes the paper.

II. The Model

A. Digital map

The system consists of a set of vehicles moving on a road network. For the purpose of route planning, each vehicle *O* has a *digital map* of the road network, denoted by DM_O. The digital map is organized by road segments, where a road segment is a stretch of a road between two successive exit points (junction, exits, etc). Each road segment has a unique identifier. For each road segment *s*, the digital map stores three attributes:

1. The identifier of *s*;
2. The coordinates of the endpoints of *s*;
3. The estimated travel time of *s*. Initially, i.e., at the time when *O* enters into the system, the estimated travel time of *s* is set to be the free-flow travel time of *s*. The free-flow travel time is the travel time when *s* is traveled with its speed limit.

B. Travel-time report and reports database

Each vehicle *O* is equipped with a GPS receiver. Every time *O* travels through a road segment *s* and reaches the end of it, *O* produces a *travel-time report*, or *report*, regarding the travel time experienced by *O* on *s*. The report contains the following attributes:

1. The identifier of s;
2. The experienced travel time, i.e., the travel time experienced by *O* on *s*;
3. The timestamp of s, i.e., the time when the report is produced.

O disseminates the produced report to other vehicles to share its travel time experience. Each vehicle stores its produced and received reports in the *reports database*. The reports database of vehicle *O* is denoted DB_O. DB_O can hold at most F_O reports. In the rest of this paper we use *R.S* to denote the road segment reported by a report *R* and *R.T* to denote the travel time reported by *R*.

C. Updating digital map upon receiving a travel-time report

Each vehicle *O* keeps a record of all reports received within the last 3 minutes. Upon receiving a new report *R* from another vehicle, the report is included in the list of latest reports and *O* updates the estimated travel time of the corresponding road segment in its digital map DM_O. *O* updates the estimated travel time maintained at DM_O according to the following equation:

$$T(O,R.S) = \alpha \cdot \sum_i (\beta_i \cdot T_i) \qquad (2.1)$$

where, $\beta_i = (\frac{1}{1+age_i})$ and $\alpha = \frac{1}{\sum_i \beta_i}$. *T*(*O*,*R.S*) is the travel time of road segment *R.S* maintained at *O*'s digital map. T_i represents the travel-time in a report received within the last 3 minutes for road segment R.S by vehicle O, where *i* is the index of the report. age_i is the age of report *i*, i.e., the length of the period of time in seconds since report *i* is produced. $\alpha \cdot \beta_i$ is the weight of report *i*, and the weights of all report *i*'s sum up to 1. The new estimated travel time is thus the weighted average of the most recent reported travel times in which more weight is given for the more recent reports. The reasoning behind using this formula for estimating the current travel time is that although the most recent report gives the best approximation of the traffic conditions, it may not be representative of the expected travel time on that road segment. For example, the most recent report may be generated by a vehicle that traveled much slower than most cars on that segment. Therefore, the formula allows for having a more representative travel time estimate, while at the same time making sure it reflects the recent traffic conditions.

III. THE TRAFFICINFO ALGORITHM

This section is organized as follows. §III.A presents the principles that are integrated into TravelInfo. §III.B discusses reports ranking. §III.C describes the TrafficInfo algorithm.

A. Principles of TrafficInfo Dissemination

Intuitively, the TrafficInfo algorithm is an integration of three mechanisms that enable each vehicle to keep its digital map as up-to-date as possible, under the bandwidth and storage constraints (see Figure 2.1). These mechanisms include:

1. How much to transmit in a broadcast. In TrafficInfo, every time a vehicle *O* travels through a road segment *s* and reaches the end of it, *O* produces a travel-time report for *s* and triggers a broadcast. The broadcast includes the produced report and the reports in *O*'s reports database. Observe that a vehicle may have a lot of reports to transmit in a broadcast but it may not be able to transmit all of them due to bandwidth constraints. How many reports a vehicle can transmit in a broadcast is determined to optimize the utilization of bandwidth. Intuitively, if the transmission size is too small, then the bandwidth is underutilized and the report dissemination suffers. On the other hand, if the transmission size is too big, then many collisions would reduce the number of successfully received reports. Thus there is an *optimal transmission size* that achieves the best tradeoff between the bandwidth utilization and transmission reliability.

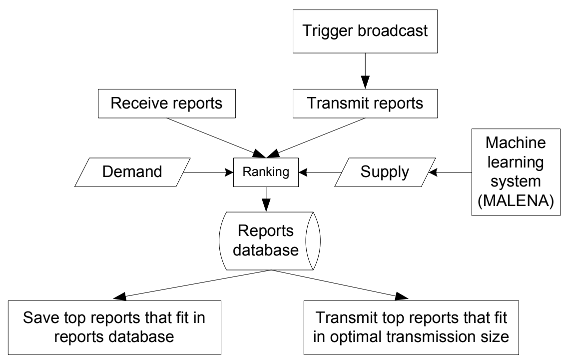

Figure 2.1. Principles of the TrafficInfo algorithm (execution at an individual vehicle)

In this paper the optimal transmission size of each vehicle for each interaction is determined based on the formula developed by [10]. We refer to this formula as the Good Citizen (GC) formula. The GC-formula is applicable to carrier-sense multiple access (CSMA) protocols, such as 802.11 and DSRC. Using this formula a vehicle dynamically adjusts the transmission size *k* based on the length of the time between subsequent broadcasts, such that the overall effective bandwidth is maximized. More precisely, our scheme is as follows. First, we developed the GC-formula that, for a given set of environmental parameters (e.g., available bandwidth, transmission range, vehicle density, transmission frequency), gives the effective throughput, i.e. the number of reports that are correctly received (namely without incurring collisions). By finding the maximum of the GC-formula, we obtain an answer to the following question: assuming that all vehicles broadcast with the same given frequency, what should be the number of reports broadcasted, such that the effective throughput is maximized.

Now clearly, not all vehicles broadcast with the same frequency (e.g., if no new reports are received by a vehicle, and its set of neighbors has not changed, there is no point in it continuously broadcasting the top *k* reports to the same neighbors). Thus each vehicle uses the GC-formula as follows. When it is ready to broadcast, it computes the length *x* of the time period since its last broadcast, and determines based on the GC-formula what is the optimal transmission size *k* for the frequency *1/x*; and it broadcasts *k* reports. In other words, the GC-formula enables each vehicle to maximize the transmission size, while avoiding thrashing (i.e. excessive collisions). Observe that thrashing is caused by the excessive broadcast by all vehicles in the system, a factor out of control of a single vehicle. But the formula enables a vehicle to be a "good citizen", i.e. broadcast only the amount that "is allowed" by the period of time from its last broadcast. A reader is referred to [10] for details.

2. What to transmit during an interaction. Observe that since the amount of transmission is limited, not all the reports in the reports database can be transmitted. Ranking is done to determine which reports to transmit. Intuitively, the rank of a report depends on its demand (how relevant it is to travel-route planning), and supply (how many vehicles already have it). The demand is computed using a spatio-temporal function that decays as the age of the report and the distance of the reported road segment to the vehicle. For the estimation of supply, we use the MALENA algorithm introduced described in §III.B.2.

3. When to broadcast the reports. In TrafficInfo, a broadcast is triggered at a vehicle *O* when any of the following three events occurs. 1) *O* reaches the end of a road segment and produces a travel-time report; 2) *O* receives new travel-time reports from another vehicle; and 3) the time length since the last broadcast at *O* exceeds a certain threshold, called the *broadcast time threshold* (*BTT*). The justification for the third event is that vehicles travelling through longer segments would have otherwise not transmitted any reports for an extended period of time. By incorporating the BTT, vehicles are forced to broadcast reports and thus speeding up the dissemination process.

There are two ways of setting up the BTT. The first is to use a fixed threshold, for example, 5 seconds. This variant is referred to as the *static BTT*. The problem with the static BTT is that whatever time is used for the threshold, it may be either too short or too long. For example, when vehicles are traveling at slow speeds, a short BTT might mean that the vehicles are in communication range for more than one broadcast period. This would possibly waste valuable bandwidth since most of the reports within that period probably remain the same. On the other hand, when vehicles travel very fast and a long BTT is used, vehicles might be in communication range for too short of a time for a broadcast to be triggered. Due to these concerns, the second way of setting up the BTT is by adjusting its value based on vehicle speeds and the transmission range. This variant is referred to as the *dynamic BTT*. In the dynamic BTT, the BTT is calculated according to the formula (r/s), where r is the transmission range and s is the current speed of the vehicle. This formula represents the duration of time two vehicles, travelling on the same road segment, but in opposite directions, will be within transmission range. It will thus guarantee that under these conditions, the vehicles will initiate broadcasts at least once.

B. Reports Ranking by Demand and Supply

In §III.B.1 we define and justify the ranking method. In §III.B.2 we discuss how to estimate the overall supply locally at a vehicle.

1) The Ranking Method

The rank of a report *R* at location *p* at time *t* is determined by the following two factors.

1. The *demand of R at location p at time t*, denoted *demand*(*R*,*p*,*t*), represents the relevance of *R* to a vehicle's route planning if the vehicle were at location *p* at time *t*. This relevance depends on two spatial-temporal factors, i.e., the distance from *p* to the road segment reported by *R* (i.e., *R.S*); and on the time that elapsed since *R* was produced. Intuitively, the demand of *R* decays with distance and time. Formally, the demand of *R* is computed as follows.

$$demand(R,p,t) = \frac{1}{c+g} \quad (3.1)$$

c is the age of *R* (i.e., the time that elapsed since *R* was produced until *t*). *g* is the free-flow travel time along the shortest-distance path from *p* to the middle point of *s*. The purpose of using the travel time along the shortest-distance path rather than the shortest-distance is to make the spatial factor and the temporal factor addable.

2. The *supply of R at time t*, denoted *supply(R,t)*, is the fraction of vehicles in the system at time *t* that have received *R* before time *t*. This number is also a global parameter that is normally unknown by each individual vehicle, but it can be evaluated by the vehicle based on metadata about *R* such as the number of times *R* has been received from other vehicles. The computation of the supply is described in §III.B.2. Formally, the *rank* of *R at location p at time t* is

$$rank(R,p,t) = demand(R,p,t)\cdot(1-supply(R,t)) \quad (3.2)$$

Now we justify Equation 3.2. Based on its definition, *demand*(*R*,*p*,*t*) indicates the relevance of *R* to a vehicle *m* at location *p* at time *t*, under the condition that *R* has not been received by *m* by time *t*. Based on the definition of *supply*(*R*), (1−*supply*(*R*)) indicates the probability that *R* has not been received by *m* by time *t*. Thus $demand(R)\cdot(1-supply(R))$ indicates expectation of the relevance of *R*.

2) Computing Supply by Machine Learning

In this subsection we outline an algorithm, called MALENA (MAchine LEarning based Novelty rAnking), for the computation of *supply*(*R*) by each vehicle. To introduce the MALENA algorithm, observe that the supply of *R* depends on attributes of *R* (e.g. its age), as well as global system parameters such as the turnover rate (i.e. the rate at which vehicles enter and exit the system). The attributes of *R* that can affect its supply are called *supply indicators*. It can be shown that, unfortunately, no single indicator is a good predictor of supply in all environments. For example, in some environments the intuition that the age of the report is a good predictor of supply is correct whereas in some other environments (e.g. when the turnover is very high) it is not.

MALENA combines various supply indicators in order to estimate the supply. The combination uses machine learning to infer from previously received reports what the indicators of a new report "look like". In other words, it learns the supply based on the supply indicators of reports that it receives. For this purpose, the set of supply indicators of a report *R*, called the *supply indicator vector* (SIV), are transmitted by each vehicle together with *R*. Upon receiving *R*, a vehicle *O* determines whether or not *R* is new, and the respective SIV becomes a training example. In other words, *O* treats itself as an arbitrary vehicle. In this fashion, *O* progressively collects a training set which improves its learning system. When *O* ranks reports, the learning system is used to calculate the supply. Furthermore, using a sliding window of examples MALENA can adapt to new environments.

C. Description of the TrafficInfo Algorithm

The execution of TrafficInfo is triggered at a vehicle *O* by any of the following three events: 1) *O* reaches the end of a road segment and produces a travel-time report; 2) *O* receives a transmission of travel-time reports from another vehicle; and 3) The time length since *O*'s last broadcast exceeds the BTT. The description of the TrafficInfo algorithm is given below.

Algorithm: TrafficInfo algorithm executed at vehicle O	
Input:	W: The set of travel-time reports produced or received by O[3]; DM_O: The digital map of O; DB_O: The reports database of O; F_O: Size limit of DB_O.
1. For each report in W, update DM_O using Equation 2.1 (see §II.C).	
2. Rank W together with DB_O, using Equation 3.2 (see §III.B); save the top F_O reports in DB_O.	
3. Compute k, the number of reports in the current broadcast, using the GC formula.	
4. Broadcast the top k reports in DB_O.	

IV. EVALUATION OF TAFFICINFO

In this section we compare TrafficInfo with Grassroots. §IV.A describes the Grassroots algorithm. §IV.B introduces the simulation method. §IV.C presents the simulation results.

A. The Grassroots Algorithm

The Grassroots algorithm (see [5]) is executed by m periodically at the so-called *maximum dissemination rate*. Upon execution, m examines the travel-time reports it has produced since the last execution of the trigger procedure. Among these, m selects the one for which the difference between the estimated travel time and the travel time actually experienced by m is the maximum. Then a flooding is initiated to disseminate the selected travel-time report (denoted by R). The flooding starts with m broadcasting R to all the neighbors. Each neighbor that receives R in turn immediately rebroadcasts R exactly once. This process is repeated by each vehicle that receives R. We conducted simulations to fine-tune the maximum dissemination rate parameter. It turned out that the performance (defined in IV.B.3) of Grassroots is optimized if m immediately initiates the flooding of a travel-time report after the report is produced by m. In this paper we do so for Grassroots.

B. Simulation Method

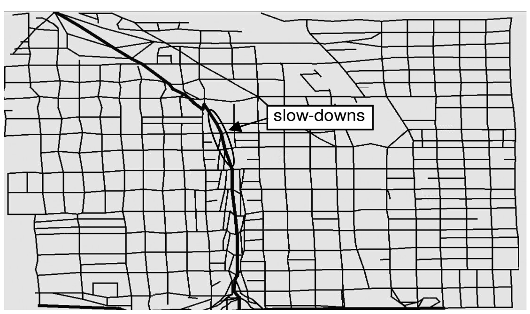

Figure 4.1. Simulated road network: portion of downtown Chicago. The thick curve is highway I-90.

We used STRAW (STreetRAndom Waypoint) [14] for the simulation of mobility. STRAW provides a microscopic vehicular traffic model on a road network. In the STRAW model, vehicle movement is constrained to roadways defined by real maps. The vehicle mobility is limited according to the speed limit of each road segment, car-following rules, and traffic control mechanisms (e.g., stop signs and timed stoplights).

The road network is a 3.2km×2.2km region of downtown Chicago taken from the digital map published by the Geographic Data Technology Inc. (see Figure 4.1). We deployed n vehicles in the region. All the vehicles participate in VANET dissemination. By varying n we varied the density of the VANET network.

Initially, each vehicle m is placed at a random location on the road network, and another random location on the road network is selected to be the destination. m then moves from the origin to the destination. When the destination is reached, another destination is randomly selected. This procedure is repeated until the end of the simulation.

[3] In the case that the execution of TrafficInfo is triggered by O reaching the end of a road segment, W includes a single member which is the travel-time report produced for that road segment. In case the execution is initiated by exceeding the BTT, W is empty.

In order to evaluate the effectiveness of each dissemination algorithm, we introduced slow-downs on a section in the highway I-90 (the circled section in Figure 4.1). Specifically, when STRAW simulates the vehicle mobility on each of these high-way segments, the speed limit is set to be 8 kilometers/hour (instead of 128 kilometers/hour which is specified by the digital map). All the slow-downs are created at the beginning of the simulation and they last throughout the simulation. In reality, such slow-downs may be caused by crashes, disabled vehicles, adverse weather, work zones, special events, and other temporary disruptions to the road network. The length of each simulation run is 1000 simulated seconds.

2) Simulation of Communication

The STRAW system uses SWANS (Scalable Wireless Ad hoc Network Simulator) [15] for the simulation of inter-vehicle communication. SWANS implements the IEEE 802.11b Medium Access Control (MAC) protocol. While simulating, SWANS considers detailed communication factors such as the decay of radio signals with increasing distance, signal collisions, and the delay for channel capturing. Using these factors it determines whether each reception succeeds, and how long it takes.

All of the nodes use the 802.11b protocol operating at 2Mbps. They share common radio properties typical of commodity wireless network cards and operate in an environment with a free-space path loss model[4]. The size of each travel-time report is set to be 100 bytes. For each vehicle, the size limit of the reports database is 200 reports. The simulation parameters and their values are summarized in Table 4.1.

TABLE 4.1 SIMULATION PARAMETERS AND THEIR VALUES

Parameter	Unit	Value	Parameter	Unit	Value
Total number of vehicles		100, 300, 500	802.11 bandwidth	bits/sec	2M
Transmission range	meter	250	Size of each report	byte	100
Grassroots maximum dissemination rate	second	60	Size limit of reports database	report	200
Slow-down speed	km/hour	8	Length of each simulation run	second	1000

3) Performance Measure

At the intuitive level, the performance measure is the difference between the travel time maintained by each vehicle for each road segment and the actual travel time of that road segment. In the following we first formally define the actual travel time of a road segment and then we define the performance measure.

For computing the actual travel time, the simulation system maintains a single special digital map which is referred to as the *true map.* The organization of the true map is the same as the digital maps maintained at each simulated vehicle. However, whenever a report is produced, the true map is updated instantaneously, using Equation 2.1 (see §II.C). At any point t in time, the travel times maintained at the true map are considered the actual travel times at t. Intuitively, the true map represents the knowledge that would have been maintained by a perfect dissemination algorithm, i.e., the algorithm that disseminates every report to every vehicle reliably and with no delay. The difference between the true map and the local digital maps indicates how close a VANET dissemination algorithm is to the perfect case.

[4] Free-space path loss is the loss in signal strength of an electromagnetic wave that would result from a line-of-sight path through free space, with no obstacles nearby to cause reflection or diffraction.

Let s_1, s_2, …, s_n be all the road segments in the road network. Let p be the location of a vehicle O at t. Let $T(s_k)$ be the actual travel time of a road segment s_k. Let $T(O,s_k)$ be the travel time of s_k maintained at O's digital map. The difference in knowledge of O at t, denoted $DIK(O,t)$ is

$$DIK(O,t) = \sum_{k=1}^{n} \left(\frac{1}{g_k} \mid T(s_k) - T(O,s_k) \mid\right) \quad (4.1)$$

where g_k is the free-flow travel time along the shortest-distance path from p to the middle point of s_k. Intuitively, the difference in knowledge of O is the weighted sum of the difference in knowledge of O between the true map and its local digital map for each road segment. The farther away the road segment is from O, the lower its weight. Specifically, the weight of a road segment s_k is $1/g_k$.

We compute the difference in knowledge for each vehicle every 10 seconds. The average difference in knowledge among all the vehicles throughout the simulation run is taken to be the performance measure.

C. Simulation Results

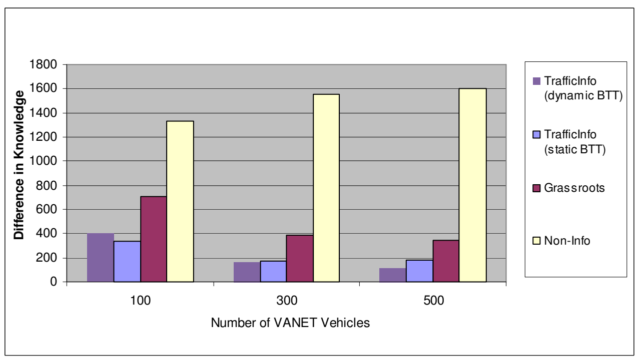

Figure 4.2. Comparison between TrafficInfo and Grassroots

Figure 4.2 shows the comparison among TrafficInfo (static BTT, dynamic BTT), Grassroots, and Non-Info. In the following we discuss this figure from three perspectives, namely the comparison between VANET (TrafficInfo and Grassroots) and Non-Info, the comparison between TrafficInfo and Grassroots, and the comparison between the static BTT and dynamic BTT variants of TrafficInfo.

VANET versus Non-Info. From Figure 4.2 it can be seen that both TrafficInfo and Grassroots significantly outperforms NonInfo, for all the VANET densities. When the VANET density is high (300 and 500 vehicles), TrafficInfo and Grassroots are over four times better than NonInfo. This quantifies the benefit of VANET dissemination. It is interesting that the DIK of NonInfo increases as the VANET density increases, whereas that of TrafficInfo and Grassroots decreases as the VANET density increases. This can be explained as follows. Recall that in the NonInfo case, the local digital maps stored at each vehicle are never updated. Thus for NonInfo, DIK measures the difference between the actual traffic condition and the free-flow traffic condition. As the VANET density increases, the actual traffic condition deviates more from the free-flow traffic condition, and thus the DIK value of NonInfo increases. In the TrafficInfo and Grassroots cases, the local digital maps are updated by the VANET dissemination. When the VANET density increases, the network is more connected and the VANET dissemination keeps the local digital maps more up-to-date. Thus, the DIK value of TrafficInfo and Grassroots increases as the VANET density increases.

TrafficInfo versus Grassroots. Furthermore, TrafficInfo significantly outperforms Grassroots. When the VANET density is 500 vehicles, TrafficInfo (static BTT) outperforms Grassroots by 66%. TrafficInfo is better than Grassroots for two reasons. First, Grassroots does not store reports and retransmit. In other words, in Grassroots a report can be disseminated only to the vehicles that have a contemporaneous path to the producer when the report is produced. Thus in the area where the network is disconnected, the reports dissemination suffers. In TrafficInfo, via store-and-forward, a report may reach vehicles that are not connected to the producer. Second, Grassroots does not have strategies to control bandwidth usage, whereas TrafficInfo does so by adaptively adjusting the broadcast size and prioritizing reports to optimize the bandwidth usage.

Static BTT versus Dynamic BTT. It is interesting to look at how the two version of TrafficInfo differ in performance. When the VANET density is high (500 vehicles), the figure shows clearly that using a dynamic BTT allowed for a more efficient use of bandwidth. For the case with lower VANET densities, the dynamic BTT does not help and on average, it is similar to the static BTT version.

V. RELEVANT WORK

Message delivery in mobile/vehicular ad-hoc networks. The work in this area is mainly concerned with sending a message to a specific destination given by the network address (see [6] for a survey). In our case the network addresses of the destinations (all the vehicles in the network) are not known a priori. There is a body of work that deals with geographic routing (e.g., [9]) in which a message is routed from a source node to a geographic location or area. However, in most of the existing literature in this area message delivery is possible only if the source and destination are connected, namely there exists a path from the source to the destination.

Disseminating traffic information in VANET's. TrafficView [4] is a store-and-forward approach to disseminating traffic information in VANET's. In TrafficView [4], multiple traffic reports are aggregated into a single report for saving and transmission. Via aggregation the bandwidth consumption is reduced. In our approach, the bandwidth constraint is dealt with via ranking and transmission size control. Thus our approach is orthogonal to and can be combined with TrafficView. The combination of TrafficInfo and TrafficView is a subject of our future work. In our prior work [8] we study the discovery of spatio-temporal resources in vehicular networks. However, in [8] the VANET is used to disseminate availability of spatio-temporal resources and the performance measure is the resource discovery time, whereas in this paper the VANET is used to disseminate traffic conditions and the performance measure is difference in knowledge.

Real-time travel time estimation. Work has been done on estimating link travel times by integrating the observations of probe vehicles (see e.g., [13]). Our approach is orthogonal to this work in the sense that we deal with travel-time dissemination whereas this work deals with travel-time updating.

VI. CONCLUSION

In this paper we proposed the TrafficInfo algorithm for disseminating real-time traffic information in VANET's. TrafficInfo includes a strategy for a vehicle to prioritize the reports based on their relevance. The relevance of a report depends on its demand (how relevant it is to travel-route planning), and supply (how many vehicles already have it). A machine learning algorithm, called MALENA, is used to enable the estimation of the supply. Furthermore, TrafficInfo adaptively adjusts the number of reports included in a transmission. With such adaptive control of transmission size, the number of collisions is minimized and the available bandwidth is optimally utilized. Two versions of TrafficInfo were presented: a static BTT version, where the broadcasting happens at fixed periods, and a dynamic BTT version, where broadcasting depends on the speed of the vehicle and the transmission range. We have shown that the dynamic BTT version is more efficient for cases where the VANET density is high.

We compared the TrafficInfo algorithm with Grassroots, which disseminates reports by flooding. The comparison is based on the simulation of vehicle mobility in a real road network and of the 802.11 protocol. TrafficInfo outperforms Grassroots by as much as 66% and maintains an advantage under different vehicle densities.

REFERENCES

[1] Traffic Information for US Cities. http://traffic.com

[2] Greaves, S.P. (2000), Simulating Household Travel Survey Data in Metropolitan Areas, Ph.D. Dissertation, Louisiana State University, 2000.

[3] GCM Travel. http://www.gcmtravel.com

[4] S. Dashtinezhad, T. Nadeem, C. Liao, and L. Iftode. Trafficview: A scalable traffic monitoring system. *IEEE Int. Conf. on Mobile Data Management*, January 2004.

[5] S. Goel, T. Imielinski, K. Ozbay. Ascertaining viability of WiFi based vehicle-to-vehicle network for traffic information dissemination. *ITS* 2004.

[6] E. Royer and C. Toh, A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks, *IEEE Personal Communications*, pages 46-55, Apr. 1999.

[7] H. Wu, et al. MDDV: a mobility-centric data dissemination algorithm for vehicular networks. *VANET'04*.

[8] O. Wolfson, B. Xu, and H. Yin. Reducing Resource Discovery Time by Spatio-Temporal Information in Vehicular Ad-Hoc Networks. *VANET*, 2005.

[9] T. Imielinski, J. Navas. GPS-Based Geographic Addressing, Routing, and Resource Discovery. *Communications of the ACM*, Vol. 42, No. 4, pp 86-92, 1999.

[10] O. Wolfson, B. Xu, H. Yin, H. Cao. Search-and-Discover in Mobile P2P Network Databases. *Int. Conf. on Distributed Computing Systems,* 2006, pp. 1-9.

[11] Y. Huang and H. Garcia-Molina. Publish/Subscribe Tree Construction in Wireless Ad-Hoc Networks. In *4th Int. Conf. on Mobile Data Management*, 2003, pages 122-140.

[12] O. Wolfson and B. Xu. Mobile Peer-to-peer Data Dissemination with Resource Constraints. *MDM'07*.

[13] B. Hellinga and L. Fu. Assessing Expected Accuracy of Probe Vehicle Travel Time Reports. *Journal of Transportation Engineering*, vol. 125, No.6, pp524-530, 1998.

[14] David R. Choffnes and Fabián E. Bustamante. An Integrated Mobility and Traffic Model for Vehicular Wireless Networks, *VANET*, 2005.

[15] R. Barr. An efficient, unifying approach to simulation using virtual machines. PhD thesis, Cornell University, 2004.

[16] H. Wu, R. Fujimoto, R. Guensler, and M. Hunter. MDDV: Mobility-Centric Data Dissemination Algorithm for Vehicular Networks. *VANET*, 2004.

[17] S. Dashtinezhad, T. Nadeem, C. Liao and L. Iftode. TrafficView: A Scalable Traffic Monitoring System. *MDM*, 2004.