VITP: An Information Transfer Protocol for Vehicular Computing
by
Dikaiakos et al.

Presented by
William Scott
October 27, 2009

- **Note:** all material taken from VITP: An Information Transfer Protocol for Vehicular Computing by Dikaiakos et al.

Table of Contents

- What is this study?
- A Service Model For VANETs
  - Context
  - Motivation and Problem Statement
  - Key Design Concepts
    - *Location-aware requests*
    - *Virtual Ad-Hoc Servers (VAHS)*
    - *VITP transactions*
    - *Return Conditions*
    - *Protocol layering*
    - *Other Issues*
- VITP Specification
- Simulation Study of VITP
  - Simulation Setup
  - Metrics and Results
    - *Effects of query distance D*
    - *Effects of vehicle density*
- Conclusion
- References

What is This Study?

- **History:**
- Proposed by Dikaiakos et al. in VANET '05
- Analysis continues..

- **What is the study about?**
  - The authors focus on the problem of providing *location based* services to moving vehicles by taking advantage of short-range, inter-vehicle wireless communication and vehicular ad-hoc networks (VANETs).
  - The paper concentrates on services that distribute on-demand information describing road conditions and available facilities in some geographic area; in particular: *traffic conditions* (e.g., congestion, traffic flow), *traffic alerts* that result from on-road emergencies (e.g., a traffic accident or a broken vehicle obstructing traffic on a road), and *roadside service directories* (e.g., location and price-lists of gas stations, location and menus of restaurants).
  - The introduction of the Vehicular Information Transfer Protocol (VITP), an application-layer communication protocol, which is designed to support the establishment of a distributed, ad-hoc service infrastructure over VANET.
  - The authors discuss
    - Key design concepts of the protocol
    - The infrastructure
    - The protocol specification
    - Simple examples of protocol interactions that support driver inquiries
    - A simulation study of VITP performance properties.

- **What are the study findings:**
  - The viability of VITP and proved the feasibility of their approach in VANETs.
  - The choice and tuning of Return Conditions affect the accuracy of VITP results, the dropping rate of VITP transactions, and response time.
  - There is a sampling-size value (*cnt*) that results to optimal efficiency with adequate accuracy: the choice of *cnt* should be done with care in realistic scenarios

A Service Model For VANETS

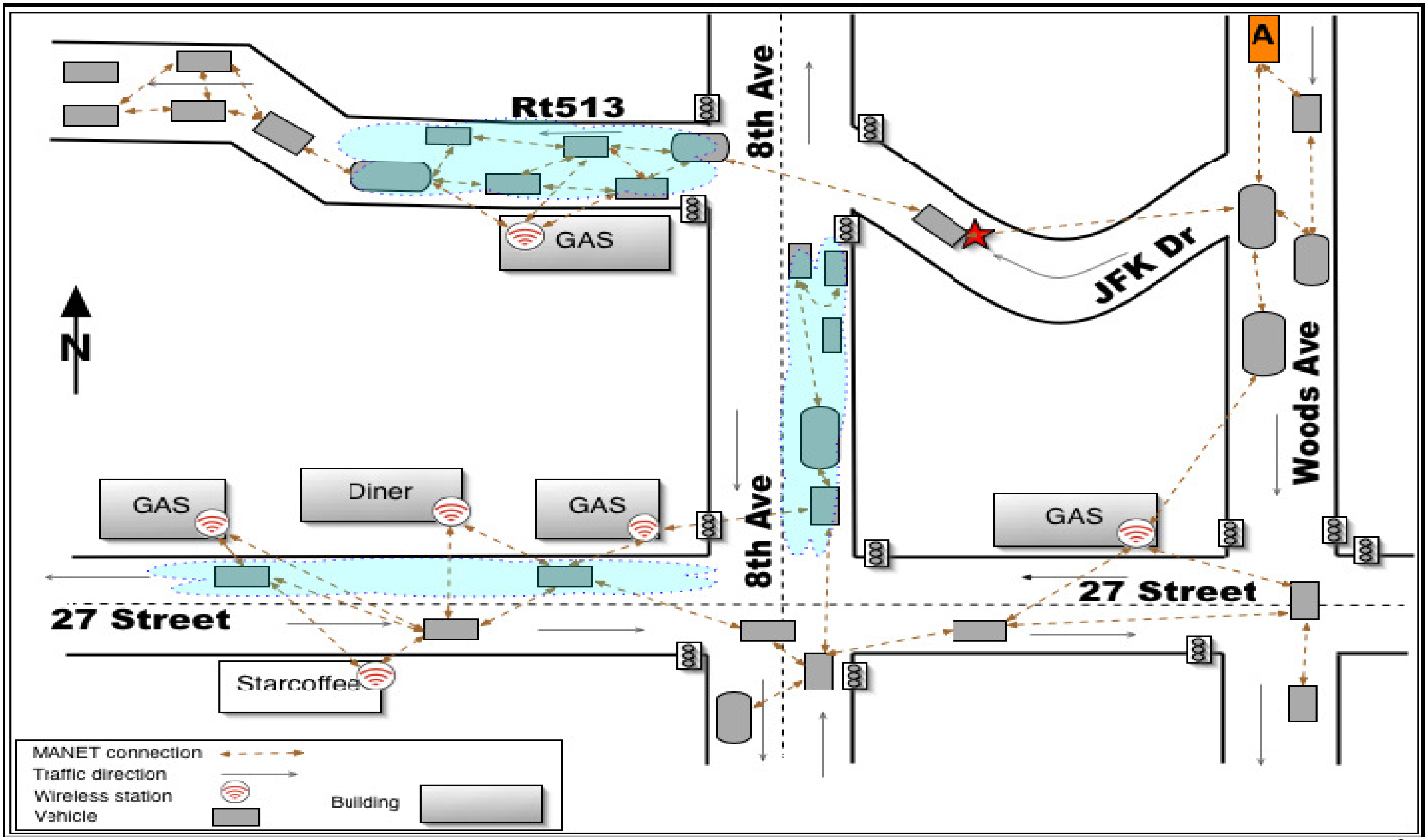
- **Context:**
  - The infrastructure needed to provide vehicular services and the VITP design, is shown in the city setting of Figure 1.
  - The scenario in the figure represents the plan of a small city-district, which is traversed by 5 streets; the direction of traffic is shown with arrows placed near the street names. A snapshot of traffic conditions shows a number of vehicles (appearing as grey boxes) of numerous sizes on the streets.
  - Most vehicles are equipped with an embedded computer with a display interface, a GPS receiver, a wireless network interface for inter-vehicle communication, and an on-board diagnostics (OBD) interface. It is possible for other vehicles to have alternative wireless network connectivity support from an on-board cellular GSM/GPRS device.
  - The OBD is used to get a set of data from mechanical and electronic sensors mounted on the vehicle (e.g., current speed and acceleration, direction of motion, average speed during the last few minutes).
  - All subsystems (GPS, OBD, wireless networking) are connected and provide data to the embedded computer.

A Service Model For VANETS

- **Context:**
  - A navigation software system is installed on the computer that relates the vehicle's geographic position to an internal data-structure representing the road networks of a large geographic area around the vehicle.
  - This data structure can be made from publicly available geographic referencing systems.
  - These vehicles form a vehicular ad-hoc network (VANET) infrastructure through their wireless connections; Figure 1 shows VANET connections between vehicles as dashed, double-headed arrows.
  - Many gas stations, coffee shops, restaurants, etc.. are equipped with short-range wireless interfaces to participate in the VANET infrastructure.
  - Vehicles and roadside stations use this infrastructure to exchange multi-hop messages that are supported by geographic routing protocols, which push messages toward their geographic destination.
  - Even without a VANET Infrastructure, messages can be sent in the area using other wireless (cellular/GPRS, Internet) & passed onto the vehicles via roadside wireless gateway stations.

A Service Model For VANETS

- **Figure 1: A vehicular service provision scenario.**



A Service Model For VANETS

- ## Motivation and Problem Statement
  - **An example of the types services to support:**
    - Imagine that vehicle A, located at the top right of Figure 1 is moving southward on Woods Ave, and is going to the West side of the city.
    - The driver wants to go either through Rt 513 or through 27th St.
    - He also wants gas along the way, but is not willing to pay over 1.8 dollars per gallon for gasoline.
    - He asks the on-board navigation system for the traffic conditions on alternative routes that lead to Rt513 West and 27th St West, and for the location of drive-in coffee shops and gas stations along those routes.
    - A possible way to Rt 513 goes through JFK Dr, which is only a few meters down the road from the present location of A.
    - So, the service infrastructure should attempt to come up with a solution to the driver's queries, before the driver decides whether or not to take the JFK Dr exit.
    - The on-board navigation system has either voice or a touch-screen interface.

A Service Model For VANETS

- **Motivation and Problem Statement**
  - **An example of the types services to support:**
    - Vehicle A ‘s drivers information request can be computed from data available on surrounding vehicles and roadside facilities located in the road segments specified by A’s inquiries.
    - For example, the traffic-flow on the segment of Rt. 513 in Figure 1 can be calculated by estimating the average speed of vehicles moving on that roadway segment; congestion in that segment can be recognized from a combination of low average speeds and a lot of vehicles on that road.
    - The gas stations operations on 27th St is known by the data sent by the gas-station’s wireless access point, which explains the services offered, the business address, and gas prices.
    - To get the data, the on-board system of vehicle A translates end-user questions into a series of location-sensitive queries which are routed to a designated location of interest by the vehicular ad-hoc network or some other network.
    - When arriving at its destination the local vehicular service infrastructure must get the query and the nodes work on-the-fly to compute an answer, which is sent back to the location where the query originated.
    - The goal here is to leverage VANETs established among vehicles equipped with these systems, in order to design a vehicular service infrastructure that carries out transactions like the ones described earlier.
    - Weakness: all vehicles and stations on the road must have sensor systems

A Service Model For VANETS

- ## Motivation and Problem Statement
  - Key infrastructure building blocks:
    - Vehicular Information Transfer Protocol (VITP)
      - An application layer,stateless communication protocol, that specifies the syntax and the semantics of messages carrying location sensitive queries and replies between the nodes of a vehicular ad-hoc network.
      - Independent of underlying VANET protocols that undertake the transmission and routing of VITP messages.
    - VITP peer:
      - A lightweight software component to be deployed on the embedded computer of vehicles.
      - Implements the VITP protocol and operate as clients, intermediaries, or servers in a VITP-protocol interaction.
    - Location encoding scheme:
      - Organizes and represents symbolically road segments and directions.
      - Used by VITP for the specification of location-aware queries and for supporting underlying geographic routing protocols, which make use of on-board navigation services to transform symbolic locations into GPS coordinates.
    - Protocol features:
      - Performance optimizations (message caching, VITP traffic reduction)
      - Quality assurance for VITP results (termination conditions of VITP queries)
      - Privacy protection of drivers.
    - VITP proposes the pull-based retrieval of traffic information, triggered on-demand by location-sensitive queries issued from VITP-enabled vehicles and also supports the push-based circulation of messages for various emergency alerts or serious traffic conditions.

Key Design Concepts

- It is believed that service provision over vehicular ad-hoc networks will be based on an extended client-server model where a driver makes inquiries about traffic conditions or available facilities on some road segment.
- These inquiries translate into query messages sent to that road segment by the underlying VANET.
- Vehicles in the destination area work together to found a server to resolve the incoming queries and to return messages.
- VITP specifies the format and the semantics of query and reply messages exchanged between vehicular clients and servers. For the design of the VITP architecture and message specification we must take into account the following key observations that will be discussed:
  - Location-aware requests
  - Virtual Ad-Hoc Servers (VAHS)
  - VITP transactions
  - Return Conditions
  - Protocol layering
  - Other issues
    - Caching
    - Message identifiers
    - Driver privacy

.

Key Design Concepts

- **Location-aware requests**
  - In the vehicular-service provision model, users are interested in traffic conditions and service facilities available to drivers in some geographic area, so vehicular-service queries are location-sensitive, specifying explicitly the target location of their inquiry.
  - Since the vehicles travel on road systems it is assumed the drivers interest is in roads, road segments, directions of traffic, and adjacent roadside areas.
  - In VITP locations are represented as 2 value tuples:
    - **Road id** - is a unique key representing a road.
    - **Segment id** - is a number representing a segment of that road; opposite traffic directions on the same part of a road are represented as different road segments.
  - The road id and segment id representation of a vehicle's location is used in GPS-based, driver-friendly navigation support systems.
  - This representation can be derived from a vehicle's GPS position with a transformation calculation that uses data from public databases that store the correspondence of (longitude, latitude) positions to road id and segment id tuples

.

Key Design Concepts

- **Virtual Ad-Hoc Servers (VAHS)**

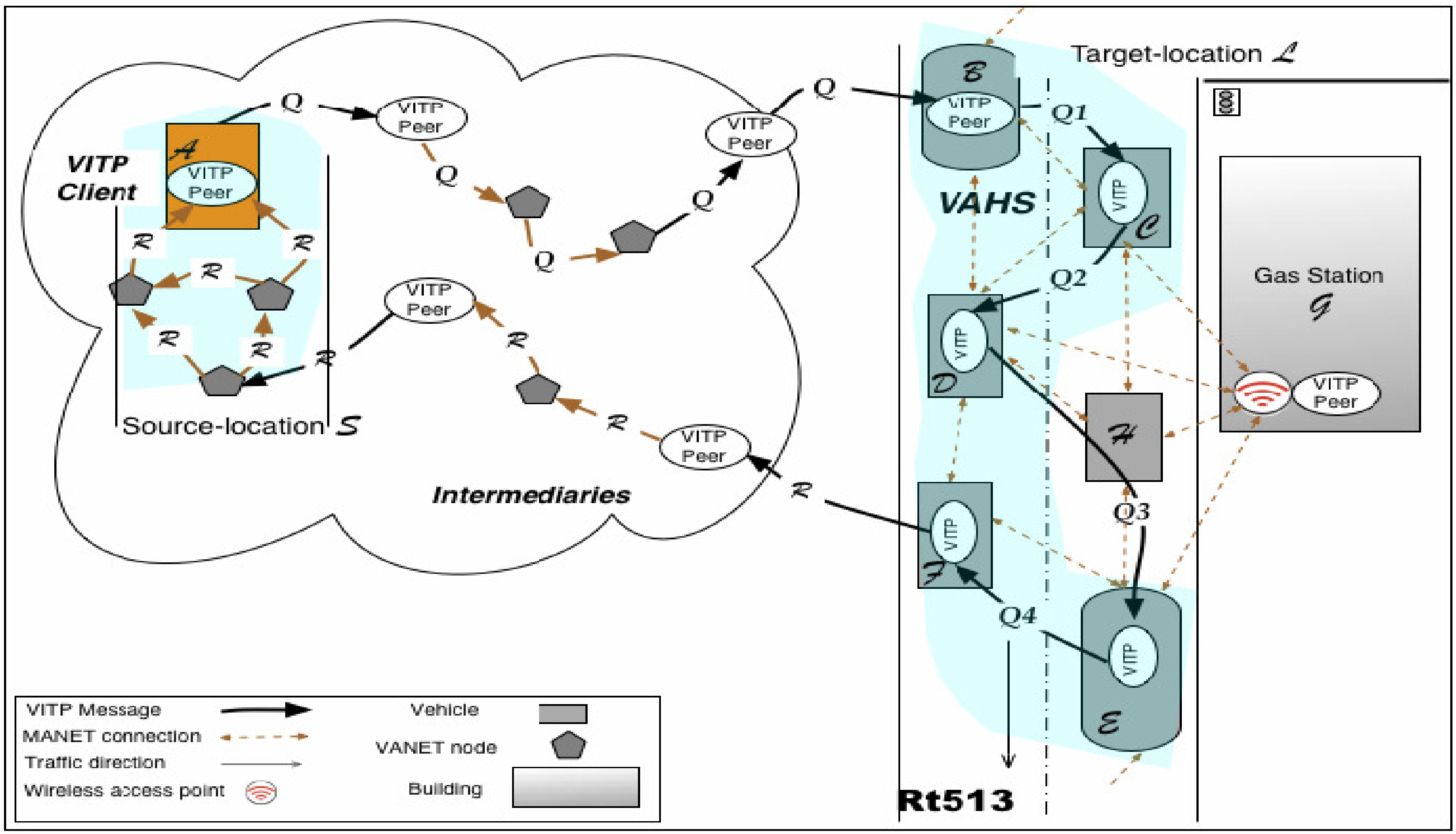
  - Because of the dynamic nature of traffic information and large size of road networks, centralized approaches for vehicular-service information provision can be expensive and difficult to implement.
  - The collection, indexing, continuous update, and timely publication of traffic-related information through centralized servers requires:
    - An extensive infrastructure of sensors that continuously monitor traffic and service conditions, collecting information even in the absence of information consumers and queries.
    - The continuous management of a high volume of messages carrying updates to central servers.
    - A highly efficient server infrastructure that can process many simultaneous updates throughout large geographic areas and can provide timely and accurate replies to queries.
    - A communication infrastructure that can carry large volumes of service queries and replies between moving vehicles and central servers.
    - Maintaining the monitoring and communication infrastructure could be too costly.

  - It is questionable whether a centralized approach would scale to a wide geographic area and to a large number of vehicles.

.

Key Design Concepts

- **Virtual Ad-Hoc Servers (VAHS)**
  - **Figure 2: Clients and Servers in a VITP transaction.**



Key Design Concepts

- **Virtual Ad-Hoc Servers (VAHS)**
  - VITP gets around the scalability problems of centralized approaches by not having additional infrastructure. so, the server that computes the reply to a VITP query is essentially a dynamic collection of VITP peers, each of which:
    - Runs on a vehicle that moves inside the query's target-location area.
    - Is willing to participate in the query's resolution by contributing in formation from its on-board diagnostics sensors or local memory.
  - The founding of this collection of peers is done by ad-hoc, and relies on the vehicular ad-hoc network established by vehicles moving inside the target-location area.
  - The dynamic and adhoc establishment of VITP servers is shown by the dynamic collection of VITP peers that are inside the target-location of a VITP query and take part in the query's resolution, as a Virtual Ad-Hoc Server *(*VAHS*)*.

Key Design Concepts

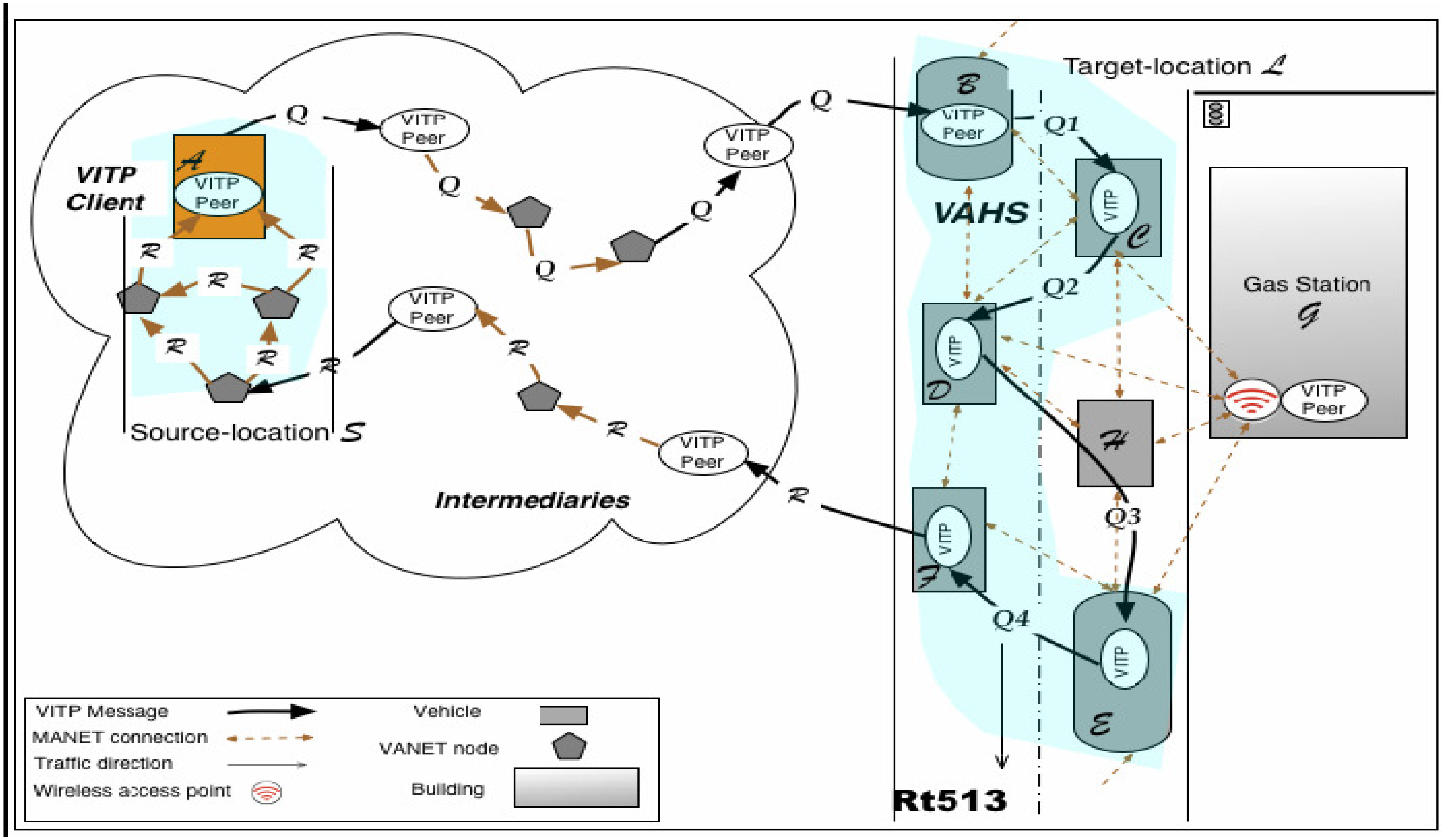
- **Virtual Ad-Hoc Servers (VAHS)**
  - Note:
    - The collection of peers that comprise a VAHS, and the VITP peers that manage the VITP communication are a best-effort operation. They provide no guarantees or special features for the recovery of lost or dropped messages.
    - A VITP peer, which has joined a Virtual Ad-Hoc Server, does not have information about other members of the group.
    - It's possible that a VITP peer joins a VAHS, does its computation, and leaves the target-location area before the completion of the query's resolution.
    - The Virtual Ad-Hoc Server does not maintain explicit knowledge of its members.
    - The VAHS is established on-the-fly; its constituents can be derived only by the choice that VITP peers make individually about serving or simply forwarding VITP requests, and by the semantics of the VITP messages they exchange. So, the Virtual Ad-Hoc Server is identified with a query and its target-location area, rather than with the VITP peers that participate in it.
  - This was done purposely in order to make the VITP protocol stateless, lightweight, to keep the VITP state-machines simple, and was necessary in order to design simple and efficient VITP peers that fit into on-board embedded processors.
  - Strength: stateless and lightweight

Key Design Concepts

- **VITP Transactions**
  - In Figure 2 a typical VITP-transaction takes place in the context of the service-provision scenario presented earlier in Figure 1.
  - The transaction is started by vehicle *A*, which is located in road segment *S* (Woods Ave in Fig. 1) and inquires about the average speed of at least 4 vehicles inside road segment *L* (1st segment of Rt513), as an estimate of traffic-flow conditions in *L*. To this end, the VITP peer of *A* submits a VITP request *Q* with *A*'s inquiry.
  - The source and the target-location areas, *S* and *L*, are connected through a vehicular ad-hoc network.
  - **4 phases of the VITP transaction:**
    1. **Dispatch-query phase-** *Q* is transported through the underlying VANET toward target area *L*. *Q* goes through a number of intermediary VANET nodes, which push the message toward its destination using geographic routing. Intermediary nodes may not be VITP enabled (these are depicted as grey pentagons in Figure 2); these nodes simply pass the message on toward *L*.
    2. **VAHS-computation phase** - VITP transaction enters this phase when *Q* is received by a peer *B* that is inside target area *L* and is willing to join a Virtual Ad-Hoc Server to resolve *Q* , the VITP request is routed between the VITP peers of the VAHS. These peers modify the VITP request to:
       - Indicate that the request is part of an ongoing VAHS computation (this modification takes place only at the first peer that joins the VAHS)
       - Piggyback partial query results to the VITP message's payload. An example, is Figure 2, when peer *B* receives the VITP request *Q*, it parses the request, extracts the requested information from its on-board diagnostics system, *rewrites* the query in order to store the partial result into the query's body and indicate that the query is now part of a VAHS computation, and passes the message on to its neighbor.
       - The semantics of such a query indicate how the underlying network protocol will treat the rewritten VITP query (unicasting or broadcasting it to neighboring peers).
    3. **Dispatch-reply phase** - When a VITP request is transported between VAHS peers until some return condition is satisfied. The VAHS peer that detects the upholding of the return condition, creates the VITP reply and posts it toward source-region *S* through the VANET
    4. **Reply delivery phase** - When the VITP reply reaches area S. During this phase, the underlying network protocol broadcasts the VITP reply to the VANET nodes of S, so that the reply can be received by the VITP peer of A. The case that a vehicle has moved outside its initial road segment by the time a VITP reply reaches that road segment, can be dealt by the VAHS by specifying an extended region over which the reply should be broadcast.

Key Design Concepts

- **Virtual Ad-Hoc Servers (VAHS)**
  - **Figure 2: Clients and Servers in a VITP transaction.**



Key Design Concepts

- **Return Conditions**
  - How to define the Return Condition for a VITP request.
    - A return condition determines at which point the resolution of a VITP request can be considered done. So, the return condition indicates if a VITP reply can be created and dispatched back to the originator of the request.
    - The decision on what constitutes success in the resolution of a VITP query depends on the semantics of the query itself, so, the return condition must be defined explicitly as part of the query's specification.
  - An example would be in Figure 2, where vehicle *A* is looking for a gas station on road segment *L*; when the related VITP request reaches the VITP peer of gas station *G*, the peer switches to the VAHS computation phase, parses the incoming query, detects that the query requests information about at least 1 gas station in *L*, and decides that it can fully resolve the query and that the return condition is satisfied.
  - So, it creates a VITP-reply message with *G*'s coordinates and prices, and sends the reply to *S.*
  - In comparison, if *A* is looking for the prices of many gas stations in road segment *L*, *G*'s peer will start the VAHS-computation phase, re-write the incoming query, and try to pass the query on through the VANET, in search of other gas stations: so, the query's return condition is not satisfied yet.
  - If there are no other gas stations in *L*, then, this return condition will never be met, the original VITP query will not be resolved, and *A*'s peer will not receive any VITP reply.
  - In these cases, VITP supports an alternative return condition, which is constrained on the total time the infrastructure can spend in a VAHS-computation phase.

Key Design Concepts

- **Protocol Layering**
  - VITP protocol presumes an underlying networking infrastructure, which transports and routes VITP messages between peers installed in vehicles and roadside services.
  - Networking support is provided by vehicular ad-hoc networks, although VITP messages could also be transported by other networks.
  - The way VANET nodes handle VITP messages is influenced by VITP semantics.
  - A VITP message that is part of a VITP transaction in either dispatch-query or dispatch-reply phase, must be routed geographically toward its target location.
  - Alternatively, a VITP reply that is part of a VITP transaction in the reply-delivery phase, should be broadcast inside its target-location area (nearby areas too), so that it reaches the VITP peer that originated the VITP transaction.
  - The routing of a VITP message in the VAHS-computation phase, depends on the semantics of the method specified in the related VITP request.
  - An example is in Figure 2, where the resolution of VITP query *Q* can be satisfied with a simple tour through the VAHS peers of road segment *L*.
  - It is done by unicasting *Q* inside *L*. another request method, or a different resolution method for the same request, may dictate the broadcasting of *Q* inside *L*

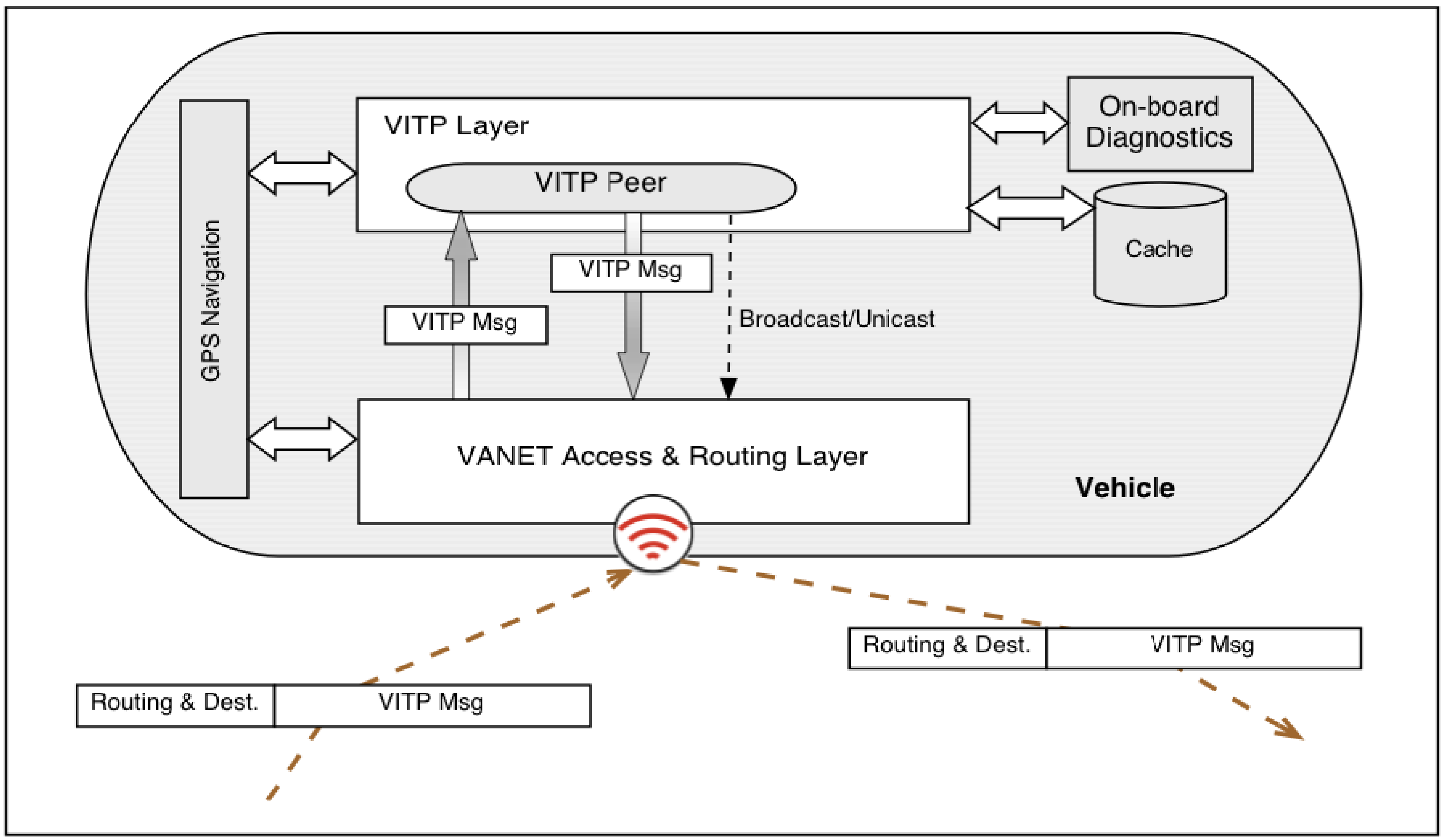
  Strength: the use of simple semantics

Key Design Concepts

- **Protocol Layering**
  - Interaction between VITP and the routing protocol of the underlying VANET is :
    - When a VITP message arrives at a VANET node, the network layer always makes a call to the local VITP peer (see Figure 3).
    - The call is made even if the peer is an intermediary (it’s not placed inside the target location of the message).
    - If the node is not VITP-enabled or if its VITP peer is busy or down, the call will fail; in that case, the network layer will retransmit the message to a neighboring node. Else, the peer will receive and parse the message.
    - Depending on the active VITP phase and the semantics of the message, the peer may re-write the message before retransmission.
    - The peer signals the VANET routing module about the routing method to be used when transmitting the outgoing message (unicast or broadcast).

Key Design Concepts

- **Protocol Layering**
  - **Figure 3: Protocol layering.**



Key Design Concepts

- **Other Issues**
  - **Caching:**
    - In Figure 2, when the VITP peer is an intermediary and the incoming message is a request, the peer searches its local cache for a matching reply.
    - The matching test takes into account both the semantics of the VITP query (as described by the query's uri) and the specification of the target region.
    - If there is a match, the peer sends the cached reply back to the VITP client and either complete the VITP transaction or retransmit the incoming message to its target.
    - This decision affects the return condition of the VITP request and must be based on the semantics of the incoming VIT P message, so, VITP provides cache-control headers that can be included in VITP messages and act as directives to VITP peer caching decisions.

Key Design Concepts

- **Other Issues**
  - **Message Identifiers:**
    - To get a VITP reply to the requesting peer and to preserve the correctness of a VAHS computation, you must ensure:
      - VITP peers can match incoming replies against its pending requests and can detect replies belonging to other peers.
      - VITP peers will not act again on the same VITP request even if it receives this request multiple times because of a random identifier (msgID) attached to every new request.
    - The msgID is also put on to messages derived from the original VITP request, that is, to modified requests exchanged during the VAHS-computation phase and to the resulting VITP replies.
    - VITP peers keep a cache of recently received msgID's. Each time a peer gets a VITP message, it compares its msgID to the cached msgID's to decide how to handle the incoming message. New identifiers are cached for a default time period.

Key Design Concepts

- **Other Issues**
  - **Driver Privacy:**
    - VITP message identifiers can be produced by random-number generators minimizing the possibility of clashes (where 2 different VITP messages have the same identifier).
    - Driver privacy is protected by 2 things:
      - The use of a randomly produced unique identifier for every new VITP request
      - Messages exchanged in a VITP transaction do not carry any information identifying the driver or the vehicle that started the transaction.

      - Note: This means that you cannot be tracked by the local police and given a ticket for speeding

Key Design Concepts

- **Other Issues**
  - **Dissemination of traffic alerts:**
    - VITP features explained to this point support the pull-based model of vehicular service provision extracted from Figure 1.
    - In this setting driver inquiries trigger all VITP transactions.
    - Vehicular applications can equally benefit from a push-based model of information provision.
      - An example from Figure 1 is a vehicle moving in JFK Dr may detect a slippery road.
      - Information about this condition (shown with an asterisk in Figure 1) will be distributed to other vehicles in the area.
      - The vehicle that detects this condition must produce an alert message and send it by the underlying VANET.
      - Supporting transmission of traffic-alert information, VITP provides a special message-type dedicated to information dissemination ("push").
      - A "push" message carries a special VITP method, the VITP representation of a target-location area, a description of the alert, a unique identifier, and other VITP-specific attributes (expiration time, caching directive, etc).
    - The protocol treats VITP alerts similarly to VITP replies:
      - An alert message is transported to its target-location by geographic routing.
      - Upon arrival, the message is broadcast to all vehicles in that area.
    - Other implementations can combine geographic routing with broadcast inside areas along the way between the source and the target location.

VITP Specification

- The syntax of a generic VITP message is given in Table 1. VITP provides 2 types of messages, distinguished by the METHOD entity placed at the beginning of each message (Line 1 in Table 1).
- METHOD takes 2 values:
  - GET represents a VITP request that queries the attributes of some geographic area (pull model)
  - POST is used for VITP replies and for messages that disseminate an attribute of some particular location toward some other geographic area (push model).
- The information requested by or transported through a VITP message is specified further by the <uri> attribute (see Table 1, Line 1).
- The last part of Line 1 declares the protocol used and its version number, and the syntax of the <uri> attribute is shown in the second part of Table 1.
- The <type> field of the uri specifies the classes of VITP-enabled physical-world entities involved in the resolution of a GET message or in the creation of a POST message.

VITP message syntax	
Line 1:	METHOD <uri> VITP/<version_number>
Line 2:	Target: [rd_id-dest,seg_id-dest]
Line 3:	From: [rd_id-src,seg_id-src] with <speed>
Line 4:	Time: <current_time>
Line 5:	Expires: <expiration_time>
Line 6:	Cache-Control: <directive>
Line 7:	TTL: <time_to_live>
Line 8:	msgID: <unique_key>
Line 9:	Content-Length: <number of bytes>
Line 10:	CRLF
Line 11:	<message body>
URI syntax	
/<type>/<tag>?[<rc_expr>,...]&<param_expr>&...	

Table 1: VITP Syntax.

VITP Specification

- **2 types of entities:**
  - Vehicle (car, truck etc..)
  - Service entity corresponds to roadside facilities that offer services to vehicles (gas stations etc).
- Other entities could be supported (traffic light).
- The <type> field can take the value all that indicates VITP peers running on any kind of physical entity may take part in the computation of a VITP request.
- The <tag> field shows the information wanted or disseminated by a VITP request.
- Example: a tag value *traffic* shows that the request queries for information on road-traffic conditions dealing with the average speed of vehicles in the area.
- A tag value alert shows the request to be either trying to get pending alerts (if used with a GET) or to post a new alert (if used with a POST).
- A special type of VITP query tag-value *index*, returns the types of queries supported by VITP peers at the query's destination area.

```
GET /vehicle/traffic?[cnt=10&tout=3000msec]&tframe=3min
GET /vehicle/traffic?[cnt=*&tout=1800msec]&tframe=0min
GET /service/gas?[cnt=4&tout=1800msec]&price<2USD
GET /vehicle/alert?[cnt=20&tout=500msec]&type=accident
POST /vehicle/alert?[cnt=*&tout=*]&type=slippery_road
```

Table 2: Type and query tag combinations.

VITP Specification

- “?” character following the tag field in the URI syntax of Table 1 separates the request specification from its parameter list.
- The list is a series of name, value expressions separated by the “&” character.
- **VITP distinguishes between 2 types of parameter expressions:**

1. Expressions used to define the return conditions of VITP requests (rc_ expr). These expressions are placed inside a pair of brackets immediately after the “?” character.
   - **2 default return-condition parameters:**
     - tout (time-out) specifies the maximum lifetime of a GET-request resolution (VAHS computation phase).
     - cnt (count) specifies the number of peers that should contribute to the resolution of a request.
   - Every VITP peer that gets a request and participates in its resolution reduces the cnt value by one., the peer checks whether the value of cnt equals zero or if the timeout period has been exceeded.
   - In this case, the return condition of the request is fulfilled and a reply can be generated and sent back to the VITP client.

2. Expressions placed after the Return Condition brackets, specify the values that are passed to the actual query that is to be executed on the VITP peer (<param expr> field in Table 1).

VITP Specification

- Examples of VITP requests are shown in Table 2. The first 2 queries seek to get traffic-flow conditions (Traffic-flow information is represented in terms of the average speed of cars or the density of cars in some area).

  1. The first query requests the average speed of 10 vehicles moving in the area of interest and specifies that this computation should be completed within 3000 msec.
     - When the query reaches a VITP peer it will retrieve from the OBD the speed of the vehicle, averaged over the last 3 mins; the 3min averaging is specified with the tframe=3min request parameter, which is specific to the traffic query.
  2. The second query is designed to estimate the density of vehicles in some area;
     - The cnt parameter is assigned a value of infinity ("*") indicating that the query should go through all the reachable vehicles and retrieve their speed
     - The VAHS-computation phase must not take longer than 1800msec.

  - The following **3** requests specify respectively:
    1. A query for any gas-station facilities with a price of not more than 2 dollars per gallon
    2. A query for any posted alerts regarding traffic accidents
    3. The generation of an alert regarding a slippery road condition.
  - The sender is not interested in getting back any information from the "slippery road" alert (a POST).
  - The purpose is to disseminate the information, so there is no return condition specified and it is up to the recipients of the alert to cache and/or keep posting it for as long as they want.
  - If the tout field of a query is set to "*" VITP can impose a default timeout return condition of many hours.

VITP Specification

- The Target and From headers of Lines 2 and 3 in Table 1 show the target and source-location areas of a VITP message.
- Locations are formatted in a standard scheme that specifies the road and segment identifiers that were retrieved by an on-board navigation and positioning system.
- With GET messages, the source-location area can be followed by an optional entity specifying the speed of the vehicle at the time that the request was created.
- This information is useful when the VAHS needs to estimate the location of the source vehicle when routing a reply back to it.
- The Time header (Line 4 in Table 1) carries a time-stamp showing the time when the VITP message was produced by its originating peer.
- The Expires header of a VITP message (Line 5 in Table 1) shows the time when the corresponding VITP transaction has to be terminated.
- If a GET request is used then the expiration time indicates that the originating peer wants to receive a reply before the specified expiration time.
- So, any peer that receives a VITP message (request or reply) as part of the transaction after this time, can drop the message and not transmit it further.
- With a POST request, the expiration-time header indicates when the VITP peers should stop sending the corresponding alert.

VITP Specification

- Cache-control header of Line 6 defines the caching directives that should apply to a VITP request.
- With GET messages, the possible values that can be assigned to this header are:
  - “no,” which indicates that the originator peer does not accept as reply a message generated during a previous transaction and cached in the infrastructure.
  - “yes,” which indicates that the originator peer is willing to accept a reply previously cached during some other transaction.
  - “fwd,” which indicates that the peer wishes that the transaction goes forward but is also willing to accept cached replies.
- Caching (Cache-control:yes) can result in VITP peers getting replies quicker and to an overall reduction of VITP messages exchanged to the detriment of the accuracy and freshness of query results.
- With POST messages, the Cache-Control header specifies whether the POST message is cacheable (value set to “yes”) or not (value set to “no”).
- When caching is allowed, the TTL header can determine how long a reply can be kept in intermediary caches.

VITP Specification

- msgID header is used to carry the unique numerical message identifier assigned to every VITP query.
- In order to protect driver privacy, instead of using as msgID the Vehicle Identification Number (VIN), they generate a random msgID by hashing a combination of the VIN, current time, and the vehicle location.
- Why the message identifier is important:
  - A copy of the msgID is cached by VAHS participants so as to avoid counting the same information twice during a query resolution.
  - This is also true for the identifier of POST requests, so that the same alert is not registered twice in the same peer.
  - A corresponding reply is identified by the same message identifier as the original query, so that the source peer will know exactly what it is looking for when receiving replies through its VANET interface.
- Lines 9-10 in Table 1 are used only in VITP requests that carry intermediate results during VAHS computation phase or in POST messages.
- The Content-Length header declares the size of the data carried by the request and the actual data follow after the CRLF character.

Simulation Study of VITP

- **Simulation Setup**
  - VITP simulation used:
    - *ns-2* simulator
    - There own traffic generator tool that accepts these parameters:
      - Simulation time
      - Road length in meters
      - Number of lanes per road
      - Average speed of the vehicles in meters/sec
      - Average gap distance between vehicles on same lane
      - Number of service nodes on the road
      - Number of users on the road.
    - The tool uses a simplified traffic model:
      - Vehicles may enter or leave the road through evenly distributed entries and exits located along the road every 1000 meters
      - Vehicles can change their speeds and lanes independently of other vehicles
      - Vehicles are evenly distributed on the road; once a vehicle leaves the road a new vehicle enters the road randomly.
    - During the simulation, they generated traffic for a 25km-long highway with 3 lanes.
    - Average vehicle speed was 20m/s while simulation time is set to 500 seconds.

I would like to see a more complex traffic (real world) model

Simulation Study of VITP

- **Simulation Setup**
  - Used wireless 802.11-compliant network with a data transmission rate of 11Mb and a transmission range of 250 meters 1.
  - Each vehicle broadcasts a Hello packet every period selected randomly from the range of 0.75 to 1.25 seconds allowing vehicles to maintain neighbor connectivity
  - Received signal strength threshold for maintaining information about neighbors is set to distances below 200 meters in order to accommodate with the fast dynamics of the network and to maintain neighborhood information consistency.
  - Upon road entry, the vehicle initiates a query at a random time chosen uniformly over its remaining simulation time.
  - Vehicle re-sends the query if no answer is received within 10 seconds of timeout. seconds.

Simulation Study of VITP

- **Simulation Setup**
  - VITP messages are forwarded to their destination region using geographic routing.
  - Because of this, you select the next vehicle in the route to be the one closest to the target destination.
  - If a vehicle has 3 consecutive message transmission failures to its next hop, it selects another neighboring vehicle.
  - If it still cannot connect with 3 different neighbor vehicles, then the query message is dropped indicating a failure of the query-dispatch phase and a failure of the VITP transaction.
  - In the next sections they experiment with the traffic query that requests the average vehicular speed within a road segment of *S* meters long, in order to get an estimate of traffic-flow at that segment.
  - They assume that the road segment is *D (query distance)* meters away from the query sender vehicle (*QS*)
  - They use as return condition *cnt* (Total number of vehicles of the target road-segment that are sampled for their speed).
  - A vehicle in the target segment can get the same query message multiple times, but it participates in updating the query results only when it gets the query message for the first time.

Simulation Study of VITP

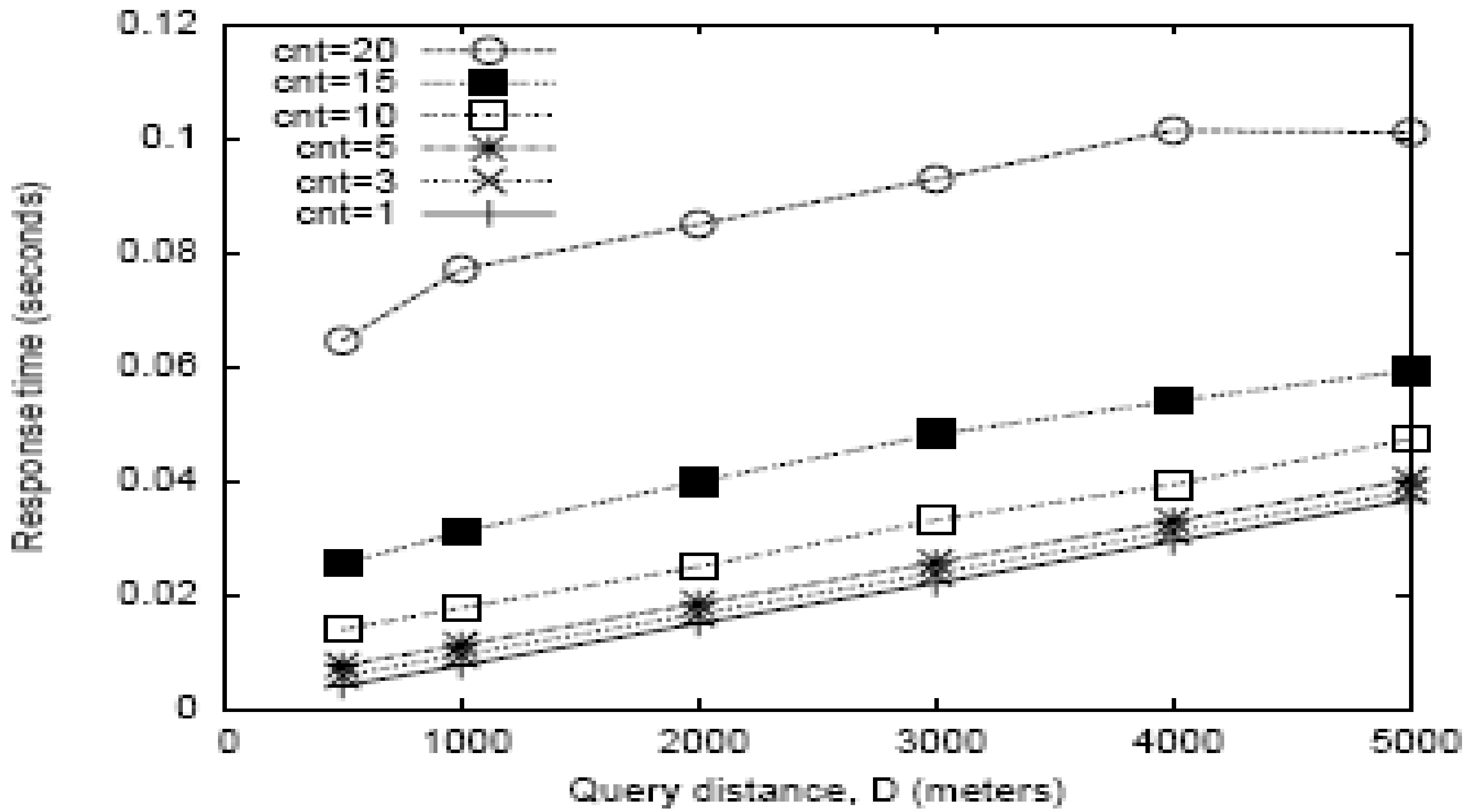
- ## Metrics and Results
  - The following metrics were used to evaluate the performance of VITP:
    - **Response time**: average time of a successful VITP transaction. It measures the elapsed time between the time a query is initiated and the time a corresponding reply is received.
    - **Dropping rate:** the percentage of unsuccessful queries (queries for which a vehicle times-out before getting a valid reply).
    - **Accuracy:** measures how close the estimated average speed (calculated by a subset of the available vehicles in the target segment) is to the actual average speed in the region of interest (calculated by considering all present vehicles in the same area).
    - **Efficiency:** measures the percentage of the number of exchanged query messages that were actually employed in calculating a result over the total number of query messages exchanged both in geographic routing and inside the target location.
  - A message is thought to have participated in the result computation when a vehicle in the target segment gets it for the first time.

Simulation Study of VITP

- ## Metrics and Results
  - **Effects Of Query Distance D:**
    - They set the average gap between consecutive vehicles on the same lane to 100m.
    - They vary the query distance *D* from 500 to 5000 meters, with the query's target segment length *S* fixed to 800 meters (This is an average of 30 vehicles moving inside the target segment).
    - The value of *cnt* ranges from 1 to 20 vehicles.
    - They assume a negligible computation time within a vehicle.
    - Figure 4 plots the response time versus the query distance *D* for different *cnt* values.
    - The response time increases almost linearly with *D* but, as the value of *cnt* increases it becomes comparable to the total number of vehicles in the target road-segment, a VITP request would have to cover a large percentage of the target vehicles to satisfy the return condition, so, the query message would have to re-visit a lot of vehicles before succeeding in finding an unvisited vehicles.
    - This means you have longer VAHS-computation and response times.
    - Using 30 vehicles in the target road-segments, they observed that response time increases substantially for values of *cnt* greater or equal to 15 (see Figure 4).

Simulation Study of VITP

- **Metrics and Results**
  - **Effects Of Query Distance D:**
    - **Figure 4: Response time vs. query distance (*D*).**



Simulation Study of VITP

- ## Metrics and Results
  - **Effects Of Query Distance D:**
    - Table 3 shows the dropping rates for different query distances.
    - The 2 dropping rates correspond to the query-dispatch and reply-delivery phases:
      - **Forward dropping rate** is measured as the percentage of the failed queries due to the failure of the query-dispatch phase, over the total number of generated queries.
      - **Backward dropping rate** is the percentage of the failed queries, due to failure of the reply-delivery phase, over the number of queries that successfully reach the region of interest.
    - Table 3 shows both dropping rates increase with query distance D.
    - In very distant queries, the forward dropping rate becomes high (i.e., 65% for *D*=5000m), making it harder for queries to complete successfully.
    - Once a query message gets to its target location, it is very possible that the reply message will be routed successfully to the *QS* vehicle, because the connectivity between vehicles during the VAHS computation phase is stable.
    - The data reported here corresponds to a *cnt* = 10 (this value results to greater efficiency and a high accuracy in the simulation scenario).
    - *cnt* had no effect on both dropping rates when using different values.

Query distance (D)	Forward dropping rate (%)	Backward dropping rate(%)
500	11.84	0.47
1000	18.41	0.64
2000	36.06	1.52
3000	50.70	2.72
4000	60.69	3.65
5000	65.95	4.24

Table 3: Dropping rates vs. query distance (D).

Simulation Study of VITP

- ## Metrics and Results
  - **Effects Of Query Distance D:**
    - Figure 5 plots the accuracy of the query results versus *cnt* for different query distances *D*.
    - Note: they achieve a maximum accuracy of approximately 90% for *cnt* values greater than 5 and the query distance has negligible effect on the accuracy.

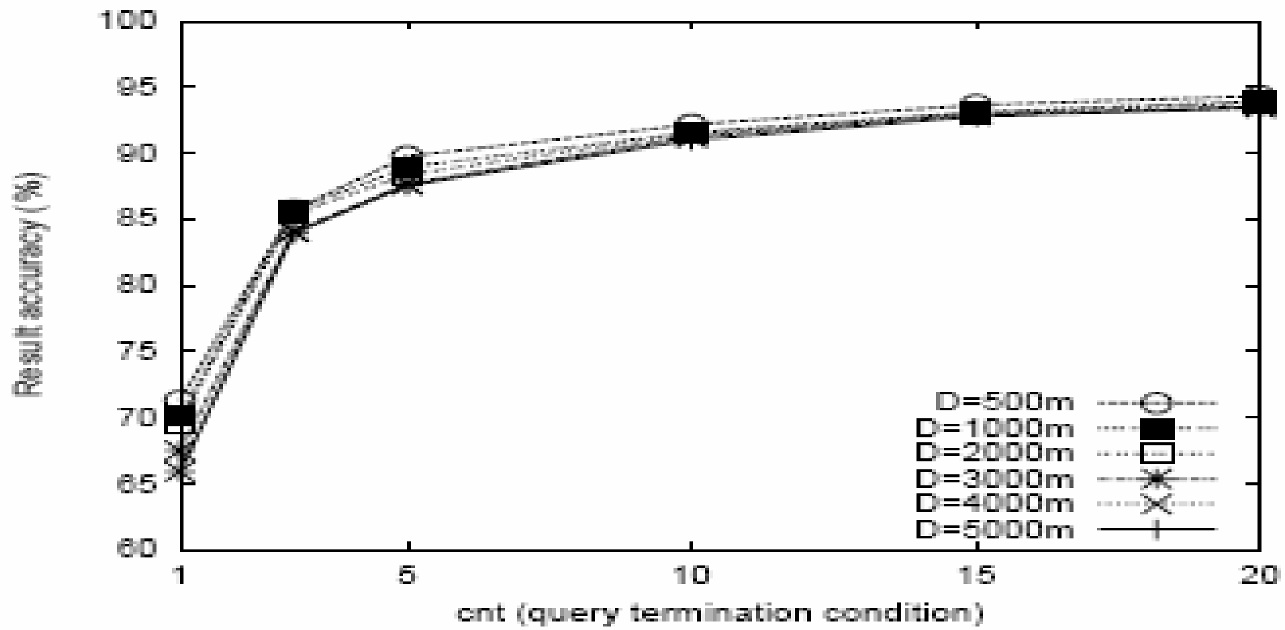


Figure 5: Result accuracy vs. cnt for different query distances (D).

Simulation Study of VITP

- ## Metrics and Results
  - **Effects Of Query Distance D:**
    - Figure 6 shows the query efficiency versus *cnt* for different query distances.
    - The efficiency is higher for smaller *D*'s because, as *D* decreases, the number of forwarded query messages in the query-dispatch and reply delivery phases becomes smaller.
    - For each query distance examined, there is an optimal value of *cnt* for which the efficiency is maximized.
    - When using smaller *cnt* values, the overhead of forwarding the query messages during the query-dispatch and reply-delivery phases dominates.
    - Adopting *cnt* values greater than the optimal value causes visits to a lot of vehicles in the target segment and results in a large number of query messages being forwarded to previously visited vehicles.
    - In the parameters and for most query distances examined, the optimum value of *cnt* is around 10.

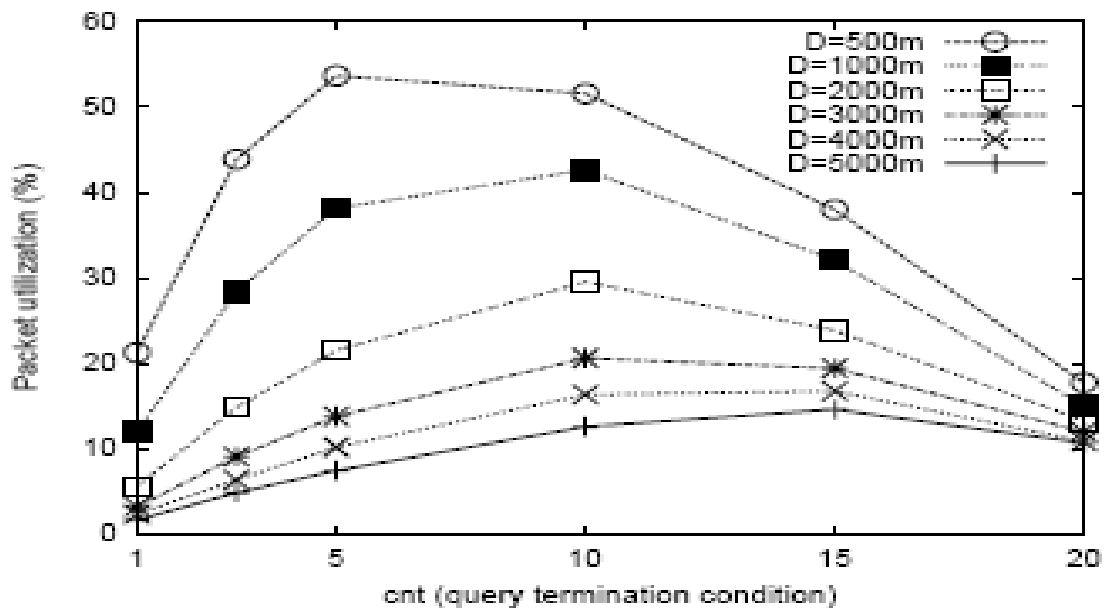


Figure 6: Query efficiency vs. cnt for different query distances (D).

Simulation Study of VITP

- ## Metrics and Results
  - **Effects Of Vehicle Density:**
    - They study the effects that vehicle density has on VITP performance using a simulation similar to the one used in evaluating the effects of query distance *D*.
    - They fix *D* to 2000 meters and change the vehicle density by changing the gap between consecutive vehicles on the same lane from 50 to 200 meters.
    - In Figure 7 the response time increases with the gap, but this effect becomes pronounced for larger values of *cnt*.

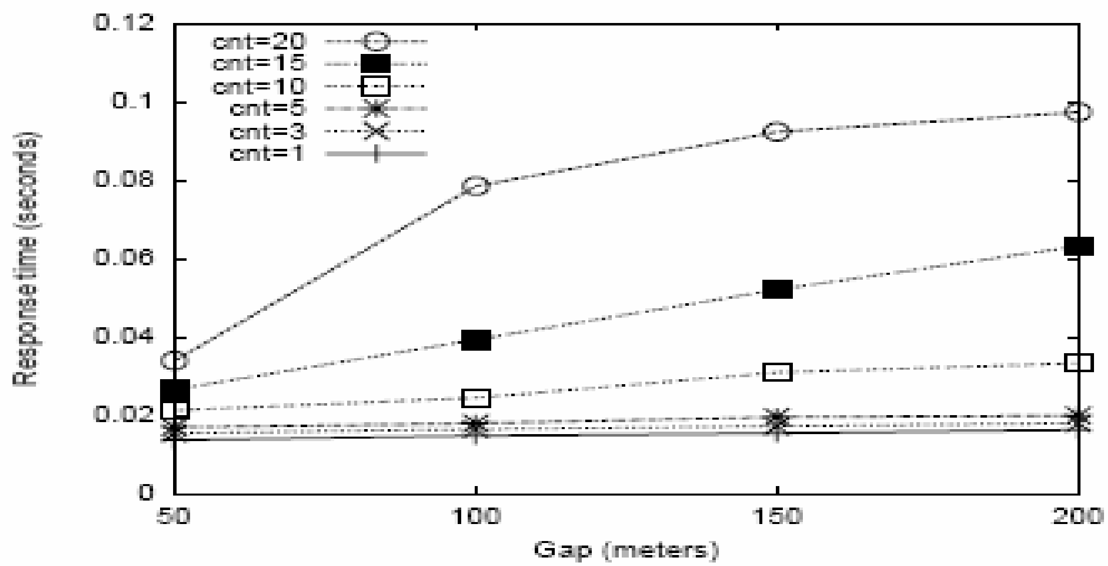


Figure 7: Response time vs. gap between consecutive vehicles for different *cnt* values.

Simulation Study of VITP

- **Metrics and Results**
  - **Effects Of Vehicle Density:**
    - For all values of *cnt*, they find that the forward and backward rates increase significantly with the gap.
    - Example: forward dropping rate for *cnt* = 10 increases from 0.81% to 89.14% when the gap increases from 50*m* to 200*m*; for the same increase in the gap, the backward dropping rate increases from 0.1% to 7.89%.
    - Figure 8 an increase in inter-vehicle gap reduces significantly the measured efficiency for large *cnt* values.
    - Why the decrease?: as the vehicle density decreases when increasing the gap, it becomes more difficult to reach the required number of vehicles in the target region. The difficulty increases more when *cnt* is large.

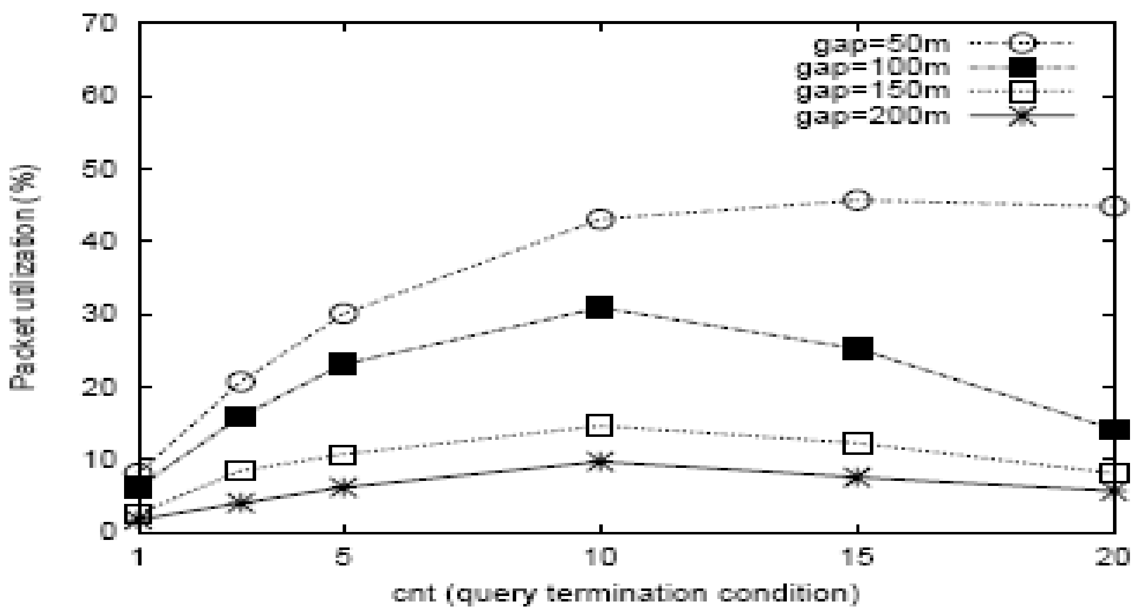


Figure 8: Query efficiency vs. *cnt* for different gap values.

Simulation Study of VITP

- ## Metrics and Results
  - **Effects Of Vehicle Density:**
    - They studied the effects that the query request rate has on VITP performance.
    - They measured the response time for successful queries and found that changing the request rate didn't have a proportional effect on response time.
    - The request rate had an effect on dropping rates.
    - An increase in the request rate resulted in a significant increase in the forward and backward dropping rates.

Conclusions

Significance & Goals:

- This paper introduced the *Vehicular Information Transfer Protocol* (VITP), an application-layer communication protocol designed to support the institution of distributed, ad-hoc, best-effort service infrastructures over vehicular ad-hoc networks (VANET).
- The paper discussed the functionality of the *VITP peer*, the software component that should be installed on VAHS-enabled vehicles to support VITP interactions.
- The authors' described the semantics of VITP transactions and provided the specification of VITP messages.
- They introduced the concept of the *Vehicular Ad-Hoc Server*, which is established on-demand as an ad-hoc collection of VITP peers that collaborate to resolve an incoming VITP request.

Conclusions

- **VITP:**
  - Application-layer, stateless protocol.
  - Supports the deployment of vehicular services on top of VANET infrastructures.
  - Specifies the syntax and semantics of VITP messages.
  - VITP messages carry location-oriented requests and replies between VITP peers of a VANET.
  - It's agnostic to underlying protocols (for routing and/or MAC-layer).

Conclusions

- **Simulation Conclusions**
  - The choice and tuning of Return Conditions affect:
    - The accuracy of VITP results
    - The dropping rate of VITP transactions
    - Response time.
  - There is a sampling-size value (*cnt*) that results to optimal efficiency with adequate accuracy: the choice of *cnt* should be done with care in realistic scenarios.

Conclusions

- **Simulation Conclusions**
  - The dropping rate in the query-dispatch phase is high and increases substantially as query distances and query request rates increase, and with decreasing vehicle densities.
  - The dropping rate parameter determines the percentage of VITP transactions that complete successfully and the end-to-end performance of VITP based applications.
  - Observed high dropping rates suggest that various optimization techniques be adopted to enhance VITP performance. Examples:
    - Use of alternative routing mechanisms in the absence of accessible routes in low-density VANETs (routing through cellular GSM/GPRS networks).
    - Caching of VITP replies inside the VANET infrastructure.

Conclusions

- **What are the Strengths of VITP?**
  - VITP has simple semantics.
  - VITP is lightweight, stateless.
  - VITP offers the ability to find optimal cnt values
  - One of the key features of VITP is that it allows nodes to aggregate (or summarize) location sensitive information and to report the summarized results to the requester.

Conclusions

- **What are the Weaknesses of VITP?**
  - Only pull-based solutions are explored.
  - No algorithm is given for predicting location of moving source on reply.
  - All simulation tests (except Accuracy) are dependent on the routing protocol used rather than VITP.
  - Return Condition value doesn't indicate timeout or count success in determining reply.
  - All vehicles and stations on the road must have sensor systems installed for VITP to work.

References

– M. D. Dikaiakos, S. Iqbal, T. Nadeem, and L. Iftode. VITP: An Information Transfer Protocol for Vehicular Computing . In *ACM VANET*, Cologne, Germany, Sept. 2005.