IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 2, Issue 5, Oct-Nov, 2014
ISSN: 2320 – 8791 (Impact Factor: 1.479)
www.ijreat.org

High Range Inquiry Dispensation and Blunder Restraint with Load Balancing in Intelligent VANET

Aswiga.R.V[1], Vijayaraj.A[2], Raja Ramya.M[3]

[1,2,3]**Department of Information Technology, Saveetha Engineering College**
Thandalam, Chennai

Abstract

Vehicles participating in the mobile network may host different number of virtual servers and they are enable to balance their load and query the RSU in the reallocation of different vehicles. Most decentralized load blancing algorithms designed for adhoc networks based on their virtual servers requires the vehicle to be asymmetric, where as some serve as the rendezvous vehicle to pair virtual servers and participating peers, there by introducing another RSU to provide response to the queried vehicle. While symmetric load balancing algorithms, introduce significant algorithmic overheads and guarantee no throughput and performance metrics. In this paper, A Novel Symmetric fault tolerance algorithm along with query processing and dispensation for adhoc network is presented by having the participating vehicles to approximate the system with global index. Each vehicle independently reallocates in our proposal its locally hosted virtual servers by advertising and inquiring the query based on their current location. Unlike competitive algorithms, our proposal exhibits performance guarantees in terms of throughput, security and load balance factor and the algorithmic convergence rate and introduces no load imbalance problem and no fault tolerance problem due to the algorithmic workload. Through NS2 Simulation we show that our proposed work outperforms existing algorithms in terms of blunder restraint and high range query processing in vehicular adhoc network with a comparable movement cost.

Index terms: *Virtual Server, RSU, SRSU, Load Balancing, query dispensation, blunder restraint, Advertisements in Ad-hoc networks.*

1. Introduction

Balancing the load in VANET is the key building blocks in the design and implementation of successful applications . Essentially load balancing along with query processing and fault tolerance in vehicular Adhoc provides the Get(x) operation to retrieve the published vehicle which is considered as object and the key is x as well as put(y,x) operation to store the value of vehicle object with hash key y. Here the load is balanced and the queries are stored in vehicular Adhoc network with the help of distributed hash table. As vehicles participating in a network are often heterogeneous, our work introduces the notion of vehicular servers to cope up with vehicles heterogeneity. In our proposed work, the RSU act as central and virtual server to cope up with vehicles. In the RSU, We have DHT with servers, different virtual servers to manage the disjoint hash subspace in S. Let N be the set of participating vehicles in the distributed hash space and v be the set of virtual servers hosted by the RSU in N. More specifically, let Sv be the hash subspace managed by the vehicle. Thus, vehicles heterogeneity can be exploited because the participating vehicles can host different number of virtual servers in RSU. Designing a query processing and fault tolerance in load balancing problem, heterogeneity aware hash space with virtual servers in RSU is technically challenging. In particular, blunder restraint and inquiry dispensation algorithms designed for Distributed hash space based on virtual servers need to take the following constraints into considerations.

2. Range inquiry in Multiple RSU

By query dispensation, we mean that each RSU manages the query proportional to its exposure capability. Previous studies suggest migrating virtual servers in RSU among the participating vehicles in order to balance the number of queries among the vehicle load. However, this is at the expense of introducing query dispensable cost due to the migration of vehicles from 1 RSU to another RSU. How to balance the queries in RSU while reducing dispensable cost as much as possible is a critical issue.

2.1 Efficiency of High Range Inquiry Dispensation

In the underlying query processing algorithms each RSU is responsible for a different partition of the domain . Since domain is uniformaly distributed among vehicles , all RSU need to bear the vehicle dynamic in mind because vehicles may dynamically join and leave the distributes hash space in RSU. In addition, the load of virtual server in RSU may change from time to time, solving the fault tolerance problem in RSU.

2.2 Algorithm Robustness and Workload

Fault tolerance algorithms need to be robust without introducing the performance bottleneck and multiple point of failure. In addition to this, a query processing algorithms also incur workloads, such workloads shall not induce another load imbalance problem. On the other hand, a well-designed query dispensation algorithm will not generate considerable overheads.

2.3 Concert Assurance

Load balancing algorithms along with fault tolerance shall work well with performance guarantee given at any traffic conditions like heavy, medium and low. Specifically distributed hash space networks may operate in dynamic and large scale environments, thus presenting a large number of traffic instances for performance investigation. To tackle the load balancing problem for query processing in RSU, prior proposals have presented centralizes algorithms for providing fault tolerance that depends on a few rendezvous nodes to balance the load of vehicles in distributed hash space in the RSU[7,8]. However considering the large scale traffic and dynamic distributed hash space networks, the centralized algorithms may introduce the performance bottleneck and the single/ multiple point of failure for distributing the query among the multiple RSU to provide the appropriate response. In contrast to the centralized behavior, some research topics,[9][10] suggest organizing rendezvous vehicles in a hierarchial manner.

Virtual servers are first matched with vehicles through RSU in the lower layer of the hierarchy for providing the response to the matched query; for unpaired virtual servers, the rendezvous vehicles relay them to the rendezvous in the upper layer to seek reallocation of virtual servers in any other Road Side Units to provide the proper response. This process repeats recursively until an unpaired virtual server reaches a rendezvous in the highest layer. For Example in [9], the rendezvous vehicles self-bind and self maintain auxiliary tree-shaped networks in [10]. The rendezvous vehicles are formatted in a 2-tier fashion. Consequently, the rendezvous vehicles may experience heavy workloads when more vehicles are taking part in query dispensation process, introducing another load imbalance problem[11]. They may also become the performance bottleneck and single point of failure. Moreover, the hierarchial networks facilitating the load balancing algorithms are prove to vehicles communication failure, thus demanding sophisticated maintenance for the vehicular Ad-hoc networks.

3. REVS Design with Fault Tolerance

We now describe REVS, our approach to enhance hash space, so that nodes can utilize the success and failures of individual lookups in a redundant search to infer malicious nodes. Here nodes refer to vehicles. REVS then directs lookups to neglect these nodes to participate in two steps: 1) the originator of a lookup picks the best possible start nodes (“local improvement”) to provide the best possible response to the originating vehicle and 2) each vehicle involved in the lookup selects high performance identity (“joint improvement”) thereby avoiding malicious identity at every step. We begin with a brief overview of the REVS idea and then explain how we apply this approach to Virtual server in RSU and the other ongoing vehicles on the road side to balance the load and query processing in VANET’s.

4. Overview

REVS can be applied to the distributed hash space for fault tolerance that meets these requirements :

1. Redundant lookups can be performed with diverse lookup paths
2. Every querying peer has a choice of peers for each entry (i.e vehicle) in its routing table. We call the set of available vehicles at each routing table entry as a targets.
3. Given a set of targets based on the redundant search, it is possible to select the correct target vehicle within that set if it exists.

The success or failure of a particular redundant search can be linked to the virtual server in that lookup. To achieve fault tolerance during range inquiry processing, we add a slight modification to the previously presented technique. ie.,when a vehicle is failed to respond or query a particular RSU, a new ring is selected to resume the processing of range inquiry, until there are no more vehicles to select from for the missing value or the query processing algorithm terminates.

Obviously, in case that all vehicles responsible for a given query at some ring have failed, all instances of this specific query value have been lost. The first requirement is the basis for the systems like RSU to provide robust lookups against moderately strong attackers. Without redundancy and path diversity in vehicles, the system’s lookup success rate will be unacceptably low[5],[6]. and REVS may not be able to distinguish between honest and malicious vehicles. For the second requirement, it is important that the choices of each vehicle in the RSU meet the basic routing requirements of the Distributed hash space. In particular any vehicle in the virtual server will

cut the remaining distance to a target by half for load balancing in vehicular adhoc networks. This requirement allows for preferential selection of vehicles within the virtual server in RSU by avoiding the malicious attackers on the road side while maintaining the path guarantees of redundant searches on the road side. The third requirement allows a querying vehicle to access the set of targets returned by the various sub lookups and pick an honest target vehicle if it exists in the return set. In most Distributed hash space, where ownership of a resource is by vehicles who are closest to the RSU, closest target to the virtual server that can be considered to be a success. If the closest vehicle in the target set is honest, then it is guaranteed to be selected. REVS treats all targets that are not closest to the target as a "unsuccessful" search. In all distributed hash space that we know of, the owner of the key being looked up will be the closest vehicle to that key in the ID space. There may be multiple vehicles, But the closest vehicle to the virtual server in RSU will be one of them. The fourth requirement allows for a mechanism to attribute success or failures of a lookup to particular vehicles. In some Distributed hash space, this mapping of vehicles to RSU is obvious because each sub lookup proceeds independently, but in other Distributed hashes space, redundancy is built into the search, and greater care is needed to keep away from negative acknowledgment. A system that meets these requirements or can be modified to meet them can apply REVS as follows. First each vehicle should track the success rate of lookups through each virtual server in the RSU. The success rate is used to calculate the reputation score for that vehicle. Second, the requesting vehicle should use these reputations scores to pick the target with the highest reputation from each virtual server situated in the RSU. With enough reputation information, the failure rate at each RSU in the lookup is expected to drop significantly because all vehicles in the virtual server must be malicious to subvert a lookup. REVS thus lookup success rate for that vehicle. To better illustrate the REVS design , we show its generality in RSU and virtual server that contains a deterministic mapping of vehicles to routing tables, where as the results of redundant searches in the road side units are combined iteratively.

To ease our discussion, we define the following terminologies and notations:

Definition 1: the load per unit capacity, ν, which is a vehicle that hosts in a unbiased shipment distributed hash space, is defined as follows for providing query processing and fault tolerance

$$\nu \triangleq \frac{\sum \nu \varepsilon V\ Lv}{\sum RSU \varepsilon N \quad \sum Vmax} \times \forall Xi$$

Definition 2: the ideal load, denoted by L_I , which vehicle νεN manages in a load balanced distributed hash space, is

$$X_i \triangleq \nu N_i^{max} * \sum \nu \varepsilon V\ Lv$$

Definition 3 : the remaining capacity of virtual server in RSU for blunder restraint iεN is

$$V_i \triangleq \chi_i - \sum L_v$$

Our load balancing algorithm intends to balance the load of participating vehicles by minimizing the load per unit capacity. It also aims to reduce the movement cost of RSU and Virtual servers as much as possible.

5.Conclusion

In a typical distributed hash space, participating vehicles can join and leave arbitrarily. Thus, reallocation of a virtual server in RSU from source vehicle to a destination vehicle can be simply done by simulating the leave and join operations offered by a typical distributed hash space to response to the query. The load of any virtual server v at a particular time for query processing and fault tolerance is the sum of load of vehicles hosted by RSU at that time; the load of a vehicle is the aggregate of load of virtual servers maintained by RSU. The potential metrics for measuring the load include vehicle utilization, travelling time, etc. We planned to implement the above mathematically proven technique to implement in real time simulators by taking possible scenarios which intends to be our future work. we assume that there is only one bottleneck resource in the vehicle, leaving the multiple-resource load balancing in the future.

References

[1] Hung-chang, "A symmetric load balancing algorithm with performance guarantees for distributed hash tables"in IEEE Transaction On computers ,Vol 62 ,NO 4 , APRIL 2013.

[2] Changqiao ,Hongke,"QoE-Driven User Centric VoD Services Multihomed P2P-Based Vehicular Networks",in IEEE Transactions on vehicular technology,vol 62,no 5,JUNE 2013

[3] Saif , Ali ,"Context Aware Driver Behaviour Detection System in Intelligent Transportation Systems(ITS)",in IEEE Transaction On Vehicular Technology,

[4] Ramin ,"DTN Routing Protocols for VANETs:Issues and Approaches",in IJCSI ,Vol 8,Issue 6,No 1,November 2011

[5] E. Strom, H. Hartenstein, P. Santa, and W. Wiesbeck, "Vehicular communications: Ubiquitous networks for sustainable mobility," *Proc. IEEE*,vol. 98, no. 7, pp. 1111–1112, Jul. 2010.

[6] D. Luckham, *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*. Reading, MA: Addison- Wesley, 2002.

[7] O. Etzion, "Towards an event-driven architecture: An infrastructure for event processing position paper," in *nRules and Rule Markup Languages for the Semantic Web*. Berlin, Germany: Springer-Verlag, 2005, pp. 1–7.

[8] M. P. Gardner, "Highway traffic monitoring," Committee on Highway Traffic Monitoring, Washington, DC, Tech. Rep., 2000.

[9] H.-Y. Cheng and S.-H. Hsu, "Intelligent highway traffic surveillance with self-diagnosis abilities," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1462–1472, Dec. 2011.

[10] R. Wang, L. Zhang, R. Sun, J. Gong, and L. Cui, "Easitia: A pervasive traffic information acquisition system based on wireless sensor networks,"*IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 2, pp. 615–621, Jun. 2011.

[11] S. Vaqar and O. Basir, "Traffic pattern detection in a partially deployed vehicular ad hoc network of vehicles," *IEEE Trans. Wireless Commun.*,vol. 16, no. 6, pp. 40–46, Dec. 2009.

[12] R. Bauza, J. Gozalvez, and J. Sanchez-Soriano, "Road traffic congestion detection through cooperative vehicle-to-vehicle communications," in *Proc. 4th IEEE LCN Workshop User Mob. Veh. Netw.*, 2010, pp. 606–612.

[13] D. F. Llorca, M. A. Sotelo, S. Sánchez, M. Ocaña, J. M. Rodríguez- Ascariz, and M. A. García-Garrido, "Traffic data collection for floating car data enhancement in V2I networks," *EURASIP J. Adv. Signal Process.*,

vol. 2010, pp. 5:1–5:13, Mar. 2010.

[14] I. Leontiadis, G. Marfia, D. Mack, G. Pau, C. Mascolo, and M. Gerla, "On the effectiveness of an opportunistic traffic management system for

vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1537–1548, Dec. 2011.

[15] A. Skordylis and N. Trigoni, "Efficient data propagation in trafficmonitoring vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 3, pp. 680–694, Sep. 2011.

[16] M. Saito, J. Tsukamoto, T. Umedu, and T. Higashino, "Design and evaluation of intervehicle dissemination protocol for propagation of preceding

traffic information," *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 3,pp. 379–390, Sep. 2007.

[17] A. Adi, D. Botzer, G. Nechushtai, and G. Sharon, "Complex event processing for financial services," in *Proc. IEEE Serv. Comput. Workshops*, 2006, pp. 7–12.

[18] N. Museux, J. Mattioli, C. Laudy, and H. Soubaras, "Complex event processing approach for strategic intelligence," in *Proc. 9th Int. Conf. Inf. Fusion*, 2006, pp. 1–8.

[19] J. Dunkel, A. Fernández, R. Ortiz, and S. Ossowski, "Event-driven architecture for decision support in traffic management systems," *Expert Syst. App.*, vol. 38, no. 6, pp. 6530–6539, Jun. 2011.