Scalable VANET Content Routing Using Hierarchical Bloom Filters

Yu-Ting Yu, Xiao Li, Mario Gerla and M. Y. Sanadidi  
Computer Science Dept.,  
University of California, Los Angeles, CA 90095, USA  
{yutingyu, pololee, gerla, medy}@cs.ucla.edu

Abstract—In this paper, we discuss scalable content-oriented routing that enables storing, sharing and searching data totally within the VANET. We introduce a proactive content discovery scheme, Hierarchical Bloom-Filter Routing (HBFR), to tackle mobility, large population and rich content challenges of VANETs. HBFR is compared to the popular ICN reactive content discovery scheme in practical VANET scenarios. The results show that HBFR suits non-sharable data services, while reactive ICN inspired content discovery works well with popular sharable data. We suggest a hybrid approach that adaptively utilizes proactive and reactive schemes for time-sensitive data in ICN VANET.

Keywords—VANET, Content Routing, Hierarchical Routing, Bloom Filters

I. INTRODUCTION

Recently, Information Centric Networking (ICN), also referred to as named data networking, has attracted much attention. In ICN, a data object is retrieved based on its content identity instead of the IP address of the node on which it resides. While several ICN designs [1-3] exist, all designs have the following common attributes: (1) receiver-oriented, chunk-based transport, (2) in-network per-chunk caching, (3) name-based forwarding, and (4) uniquely identifiable content naming.

ICN is beneficial to Vehicular Ad-Hoc Networks (VANETs), which is widely used in tactical and emergency networks and expected to be an important component of future urban networks. In the future, VANET will provide various services such as communication, storage, and computing for a range of applications from safe navigation to on-road entertainment, live video streaming, Internet access, urban surveillance and massive net games.

Intuitively, the multi-source nature and in-network caching ability of ICN is helpful for conquering the mobility and intermittent connectivity challenges that were difficult to solve in traditional IP network. For example, a data retrieval failure due to intermittent connectivity can be recovered more quickly by leveraging distributed caches. However, with so much content around, the major challenge in VANET ICN design is the scalability of content searching. That is, the ability to efficiently find the content that was requested by the driver, or that was matching the mission requirements.

Latency is critical to the retrieval of time-sensitive data, for example, orders and terrain images in tactical networks. Two common approaches to mitigate latency are frequent routing updates (in a proactive approach) and content replication/caching (in a reactive approach). However, frequent routing updates cause excessive overhead in high mobility and large population; content caches become quickly obsolete in real-time applications. In this paper, we advocate a hybrid forwarding framework that adaptively performs proactive or reactive content discovery based on content characteristics. We propose to utilize the content identifiability and to use the content name prefixes to classify the time-sensitive data and apply the most suitable strategy to each category. For proactive content dissemination and discovery we propose a Hierarchical Bloom-Filter based Routing algorithm (HBFR). HBFR is structured in a self-organized geographical hierarchy, making the approach scalable to large metropolitan VANETs.

The organization of this paper is as follows. In section II, we review and evaluate existing ICN routing approaches. In section III, we propose the hybrid design intended to provide effective support to the various categories. We introduce HBFR in section IV, and report on its implementation and preliminary results in section V. Section VI concludes the paper.

II. INFORMATION CENTRIC ROUTING

ICN differs from IP-based routing in two aspects. First, the number of content names to consider in ICN is significantly larger than the number of IP addresses. Second, in host-based networks, each IP address is associated with one host interface. In contrast, copies of data object chunks may exist in multiple locations. The resultant multi-source nature must be taken into account in ICN routing design.

We discuss below three ICN routing approaches for VANET: reactive routing, proactive routing, and opportunistic forwarding.

A. Reactive name-based routing

The reactive name-based routing consists of two phases: content discovery and content forwarding. Reactive content discovery is accomplished via flooding. Consumers initiate content retrieval by flooding their requests. Once the data object is found, it is forwarded back to the consumer by an appropriate forwarding scheme. In vehicular networks, popular forwarding schemes uses similar concepts as the ones in IP
routing such as AODV[6] and DSR[7], and GPSR [8].

CCN [2] is an example of reactive ICN routing design. In CCN, the requests are flooded if the routing information is unknown. Once the data is found it is forwarded in reverse on the path(s) from which the query arrived (breadcrumbs routing).

Flooding-based content discovery can quickly find the nearest cached data objects. However, forwarding algorithms such as AODV are vulnerable in a vehicular setting due to path intermittence [9]. This approach also suffers from the cached fragment phenomenon. That is, path discovery may lead to a cache that holds only a few data chunks and thus gets quickly exhausted. One must frequently re-flood to discover the remaining chunks. Frequent large-scale flooding may cause network congestion and nullify the cache benefits.

B. Proactive name-based routing

Proactive name-based routing is based on data object advertising. Flooding is not required since the object locations are pre-announced. The in-network storage is still useful, but only if there are cached copies along the constructed path. This way, the foreground search overhead is much lighter than that in reactive routing. On the negative side, there is significant background advertising overhead.

DONA [1] uses an example of proactive ICN routing. DONA maintains a name-resolution hierarchy. Nodes must register all data objects and their locations to Resolution Handlers (RHs), which form a hierarchical structure, and content searches are done by following a recursive resolution process similar to DNS.

Using proactive approaches in vehicular network is challenging due to several reasons. An ICN would require storing location information of at least $10^{12}$ data objects [10]; maintaining location information for all content names means significant cost in vehicular network. If names are hierarchical, as postulated in [2], one can perform prefix routing (as currently done in the Internet for DNS Domain Names and for IP addresses) rendering the problem more manageable. However, only a selected set of prefixes can be advertised.

C. Opportunistic forwarding

The two previous approaches require an active (i.e., reactive or proactive) content discovery phase. To avoid the associated overhead, an approach called opportunistic forwarding, or carry-and-forward scheme, has also been pursued. Requests and advertisements are not flooded or propagated proactive to the entire network. Rather they are disseminated hop by hop through “opportunistic” node encounters. This scheme allows total control on propagation overhead. The Haggle [12] architecture is built on an opportunistic approach.

The opportunistic scheme significantly reduces the request flood and location announcement overhead. However, it also introduces significant latency and was originally developed strictly for delay tolerant applications [10][11]. It is obviously not suitable for time-sensitive applications.

III. HYBRID ICN ROUTING FRAMEWORK

Observing that existing routing approaches have complementary advantages and disadvantages, we propose a hybrid routing framework for time-sensitive VANET data services. The hybrid design uses different routing methods for different content. In the following, we classify data services and pair them with suitable routing approaches.

A. Data service categorization

Data Services classification follows closely the content time-sensitivity. We can classify Data Services in three categories:

1) Popular sharable data services (Type A): these services retrieve time-sensitive sharable data. These data are generally accessible by most users at least in a local geographic scope and are small enough to be cached at relay nodes. General orders, emergent announcements, map services are examples of this category. Given the small file size and sharability, these data can be cached by most vehicles. Content re-discovery can quickly retrieve data without large-scale flooding. Reactive routing is the most suitable routing scheme for this category.

2) Popular non-sharable/non-cacheable data services (Type B): This type indicates popular data services that produce non-sharable or non-cacheable data. For example, contents such as access-restricted orders and highly-credidential information pose privacy and security threats and are non-sharable. Large-size, popular recorded video files with time sensitive delivery may be too large to be completely cached at relay nodes in high mobility, else they would trigger the cached fragment problem. Yet, these data services are “popular” since there are many requesters, while there are only a limited number of providers that can supply these files. For popular, delay-sensitive data that cannot be cached because of with privacy restrictions or large size, it is worth to proactively advertise the path to mobile providers in order to shorten latency and save search traffic overhead.

3) Unpopular data services (Type C): one example of unpopular data is private messaging. For unpopular contents, the individual chunks are not expected to stay cached in the network for very long since the in-network storage size may be limited. The in-network caching is beneficial mainly for recovery in very intermittent connectivity. Large-scale blind request floods for unpopular contents should be avoided if at all possible. In delay tolerant applications (e.g., non time critical messaging), opportunistic forwarding can be used to avoid route construction/maintenance costs. However, if the data is time-sensitive, it may be necessary to accept some content discovery overhead for better quality of service, depending on time criticality. In the latter case, the reactive routing is the only option since it can achieve low response time. In this paper, we will not pursue unpopular contents. Rather, we will focus on popular data services.

B. Content naming

To adapt the routing to content types, we design the following naming convention for each data chunk: /Category/Service_name/Additional_info/. Category may
represent one of the three above mentioned categories. It is then easy for the system to decide which routing approach to apply at runtime. Service_name is an identifier of a service. Note that a service may be provided by multiple provider nodes. Additional_info includes the content identifier. For example, a request for a map of Westwood to a local sharable map service can be named as /type_a/map/los_angeles/westwood/, and an access-restricted tactical video clip in a peace keeping operation in Kabul may be named as /type_b/video/location/.

For popular data services, we assume the existence of a changing-over-time prefix list that is retrievable with a known identifier. Moreover, we assume the availability of a Network Edge Service that allows the customer to precisely and non-ambiguously formulate the query with the above naming convention and thus determine, from the prefix list, the query type, A or B.

IV. HIERARCHICAL BLOOM-FILTER ROUTING (HBFR)

In urban networks, we expect that popular non-sharable data services will dominate the mobile ICN traffic. Therefore, the hybrid framework requires an efficient proactive ICN routing algorithm for popular non-sharable/non-cacheable data. However, due to the high mobility and limited bandwidth in vehicular networks, previous proactive designs that require all contents to be announced create too large overhead. To handle the large size of typical urban VANETs, we propose Hierarchical Bloom-Filter Routing (HBFR) for reactive content routing.

We speculate that this content will be offered by a few fixed or mobile providers. We expect the popularity of non-sharable data services follows Zipf’s distribution as the web traffic does [13-14]. With these premises, it is reasonable to use bloom-filters [15] to announce only the popular prefixes with a well defined, consistent convention followed by producers and consumers. Bloom-filters are widely used in applications such as Internet caching and P2P content discovery [16-19]. The two key advantages of Bloom-Filters for content advertising are: (1) Bloom-Filter size adjustable to the max number of prefixes and; (2) Bloom-Filter aggregation.

A. HBFR Overview

In HBFR, the urban map is hierarchically organized into geographical partitions and the vehicles are likewise partitioned in corresponding clusters. Vehicles are equipped with GPS. Thus the vehicles are time-synchronized and know which partition they are in. In Fig. 1, a three-level rectangular partitioning that fits a Manhattan grid topology is shown. We denote the partition \( j \) of level \( i \) as \( \{i,j\} \). In practice, leaf partitions can be designed to correspond to road segments. Segments joined by road intersections form the next level partitions and so on. Each vehicle reads the number of levels and the cluster maps when it enters the local VANET.

We use Bloom-Filters (BF) to advertise the presence of name prefixes in the corresponding partitions. At each level, each node stores BFs of its own partition and the sibling partitions. In other words, each node has \( n \) BFs per level \( (n=4 \text{ in Fig. 1}) \), corresponding to \( n \) different partitions at that level; namely, the partition to which it belongs at that level, and the three sibling partitions. These BFs are disseminated in the four partitions. During the dissemination process, each BF, while in its partition, picks up the updates from its lower level BFs by XORing itself with them. In the sibling partitions, the BF is not updated.

For the leaf partition to which it belongs, a node constantly aggregates (XORs) its node BF, which summarizes content service identifiers residing in this node, with the leaf-partition BF and propagates BF throughout the leaf partition. The node also receives and forward, but does not aggregate, the BFs of the sibling partitions or higher-level partitions. Note that only providers and full caches should include the prefixes in their node BFs. Nodes holding partially cached chunks must not claim themselves as providers. Nodes that cache partial contents do not include content in the BFs.

Considering the example in Fig. 1, it follows that at the top level, at steady state there are four different BFs in circulation, one corresponding to each partition and reflecting the content of that partition.

Note that, different from the previous approach [22], there is no cluster head in each partition to supervise the BF aggregation. The aggregation is completely distributed. The major advantage of the distributed BF aggregation is robustness to mobility. In a highly mobile network the cluster head may be short lived and must be reelected when it drifts out of its region. Moreover, in a sparse VANET, the cluster head may become isolated from other nodes thus jeopardizing the proper transmission of BFs to upper layers. HBFR is robust to node isolation and intermittent connectivity, since it percolates content information via BFs in parallel among all nodes.

The content search mimics a DNS query that is refined level by level, starting from the top. Each node has the full view of all the BFs in all the partitions to which it belongs. We illustrate the procedure by example. Since the laptop shown in Fig. 1 is in partition \( \{1\} \), it has BF\{0,0\} and BF\{1,0\}, BF\{1,1\}, BF\{1,2\}, BF\{1,3\}. Suppose it receives a request for content that happens to be in \{2,10\}. The laptop starts from BF\{0,0\}. It finds a match. Next, it checks the lower level filters and finds a match in BF\{1,3\}. It forwards the query to partition \{1,3\} using geo-assisted forwarding. If geo-assisted forwarding fails because of a dead end, it forwards the query on the path from which the BF\{1,3\} came, i.e. breadcrumb forwarding. Finally, the leaf BF with the content is found. The request is then flooded in \{2,10\}. The reader may note that a similarity between the above...
hierarchical structure and the GHT (Geographic Hash Table) structure in [4]. If most queries are locally satisfied both schemes scale with $O(\log N)$, where $N$ is the name space size. BFR offers the advantage of aggregation, with the drawback of false positives. In all, BFR was judged more practical than GHT for the non-sharable data services. The latter requires posting of the content in nodes at specific positions. The nodes may or may not be there when the query arrives. Moreover, the BF structure is more robust to mobility and to sparse connectivity. In BFR, the requests are directed toward the service providers via a structured request propagation network. The path load balancing problem is addressed by the in-network caching.

B. Implementation consideration

We assume the service providers can judge if the content they provide is sharable. The popularity of services is decided based on prior knowledge such as the statistics collected from previous missions. If the provider determines that the content is popular and sharable – based on number and diversity of request it received, it stops advertising it in the BF.

To balance the tradeoff between overhead and accuracy, the BF dissemination frequency is set proportional to the degree of change of current BF advertisements. The degree of change is defined as the number of bits newly set/removed compared to previously received BFs.

The BFs must be periodically refreshed to account for content withdrawals due to mobility across cluster boundaries and to be promoted from non-popular to popular (thus relying on abundant caches). A possible way to refresh the BFs is to circulate both the old version, which is frozen and is used for routing, and the new version, which is being updated, simultaneously. Once the new version stabilizes, a new cycle is started. The above can be achieved by defining a waiting period which is long enough for the new version to become stable. Nodes can switch to new version and start the next BF circulation when the waiting period expires.

C. Scalability of the HBFR approach

The HBFR approach includes three sources of potentially non scalable overhead: BF size storage overhead; BF dissemination overhead and query search overhead. The BF size can be contained by limiting the entries to “non-sharable” prefixes and by excluding prefixes that turn out to be sharable by a large population. In the worst case, false positives will occur at the top level causing multiple queries in parallel. However, false positives are resolved near the bottom of the hierarchy. BFA dissemination can potentially explode with millions of vehicles posting data. However, Fisheye routing [20] can be used thus reducing the dissemination to $\log(N)$ where $N$ is the number of levels, proportional to system size. Query routing overhead is proportional to $\sqrt{N}$ if the content is uniformly distributed over the urban grid. However, in most practical cases the content is location relevant (e.g. traffic jam information) and thus the query is satisfied in the local partition, reducing query routing overhead to $\log(N)$.

V. IMPLEMENTATION AND EXPERIMENTS

We implemented HBFR, a proactive ICN routing approach, and CCN [2], a reactive ICN routing approach in Qualnet 6.1. We simulate three VANET scenarios in which 50 vehicles with average speeds range from 5m/s to 30m/s (11.18 to 67.1 mph) are moving in 2km by 2km maps. The three maps and all mobility traces are generated by VanetMobiSim[21] using different seeds with the same settings. It simulates vehicle behaviors considering traffic lights, congestion, lane changing, acceleration, car following model, etc and the maps include three clusters: residential, suburban, and urban areas, approximately occupy one-third of the 2km² area in each scenario. One of the experiment scenario is shown in Fig.2 as an example. The other two scenarios have similar but different layouts.

The simulation time is 1000 seconds. The applications are CBR applications which initiate a new request every second. Since all packets are broadcast, intermediate nodes are allowed to retransmit requests for up to two times if the request is not forwarded to the next hop successfully. Data consumers may reinitiate the request to prevent data loss for up to four times. The retransmission interval is 100ms and 500ms for intermediate nodes and data consumers, respectively. The Bloomfilter size used in the simulation are 72 bytes. We use a two-level partition map similar to that in Fig. 1 but the leaf partition size is 1000m by 1000m.

We measure (1) the response time, which is defined as the interval from each request sent to its corresponding data chunk received, and (2) the completion rate, which is defined as the ratio of the number of corresponding data received at the receiver side over the number of requests sent. We also measure request, data, and control traffic rates of the network, which includes the request, data, and control traffic forwarded by all nodes. All experiments are repeated 20 times for all scenarios using different seeds. Confidence intervals (CI) are reported.

A. Popular sharable data service

Our popular sharable data service scenario includes ten data consumers downloading the same 900MB file from one mobile data provider. All downloads start at the same time at 7 second after the BFs are stabilized.

In Fig. 3, we show the average response time of each
scenario. Both HBFR and CCN achieve short response time less than 15 ms. Interestingly, in some cases HBFR performs slightly better than CCN. The reason is that HBFR pursues a single-path approach using georouting, and hence has less channel contention than CCN, which uses a flooding approach, does. Since the data is sharable, the flooding-based content discovery leverages the cache and does not create too much traffic or congestion.

To take a closer look, we report the request and data traffic of HBFR and CCN. The request traffics of CCN are almost identical in all three scenarios despite the different mobility patterns and geographical map since redundant requests for the same data segments are reduced because of the data identifiability of ICN. Note that HBFR actually reported less overall request transmissions in our experiments. However, HBFR requires more overhead in the packet header to accomplish georouting, and hence introduces larger traffic. The results in Fig. 5 prove the above claim since HBFR pulls less data traffic than CCN does. Data traffics are proportional to the number of request packets forwarded.

Due to the limited space, we do not show the results of control traffics in figure. The average control traffic of HBFR is 4.32KBps. The standard deviation is 0.32KBps.

B. Popular non-sharable data service

We use the same scenarios for this experiment again. However, ten data consumers request access-restricted ten different real-time generated files from one single data provider, which means the file is encrypted and thus cached chunks cannot be used by others. The file size remains 900MB. All other settings are the same as before.

Fig. 6 shows the average response time of this set of experiments. Since the data is non-sharable, flooding does not benefit from caching. Therefore, frequent flooding causes congestion and triggers the hidden-terminals problem even in the simple ten-flow scenario. BFR improves the response time by approximately 45% compared to CCN. As all requests are
forwarded towards the data provider, BFR eliminates most of the traffic and hence is able to maintain a response time at about 170ms for the three different scenarios. In Fig. 7 and Fig. 8, the request and data traffic are presented. We confirm based on the results that CCN introduces a much higher request volume and consequently much more data traffic than HBFR due to multipath flood.

The average control traffic of HBFR is 4.1KBps. The standard deviation is 0.33KBps. The results are similar to that of the sharable data experiments since the control packets of HBFR are periodically sent. However, note that the total traffic of HBFR is much less than that of CCN.

VI. CONCLUSIONS

In this paper, we discussed the tradeoffs between proactive and reactive content-oriented VANET routing for different content types. To enable scalable proactive VANET ICN routing, we proposed HBFR, a scalable routing method using Bloom-Filter for content advertisement. The procedure utilizes hierarchical geographical partitioning. We evaluate the proposed methods by simulation. Our results show that the CCN and HBFR achieve comparable response time for popular sharable data. However, HBFR requires more overhead than the reactive CCN approach. For popular unsharable data, our proactive routing approach, HBFR, achieves 45% shorter response time and about 70% less traffic than that of CCN. This confirms the reason to use a hybrid content-oriented routing policy in VANETs. Future work will evaluate the proposed framework under realistic content retrieval traces and will evaluate the performance and assess the validity of the hybrid approach.

REFERENCES