Modeling Epidemic Query Dissemination in AdTorrent Network

Alok Nandan, Saurabh Tewari, Leonard Kleinrock
Computer Science Department
University of California Los Angeles
Los Angeles, CA 90095-1596
{alok, stewari, lk}@cs.ucla.edu

***Abstract*— One of the most important sources of revenue for big Internet-based companies are advertisements. With vehicular networks poised to become part of the Internet, this new "edge" of the Internet represents the next frontier that advertising companies will be striving to reach. In this paper we investigate the design parameters for *AdTorrent*, an integrated system for search, ranking and content delivery in Digital Billboards, a scalable "push" model architecture for targeted dissemination of ad content in a car network. We present an analytical model to estimate the performance impact of key design parameters such as the scope of the query flooding on the query hit ratio in epidemic query dissemination.**

I. Introduction

As advertisers struggle to reach increasingly distracted and jaded American consumers, they have sought nontraditional media for their Ads, from elevators to cell phone screens.

Content-targeted advertising paradigm has proved to be a resounding success in advertising on the conventional Internet. As the Internet expands to mobile devices, even vehicular nodes are becoming a part of the "edge" of the Internet. In this paper, we discuss the design issues relevant to a system for targeted ad delivery mechanism for cars. This system is comprised of *Digital Billboards*, a push model architecture for disseminating advertisements to vehicles, and *AdTorrent*, an integrated system for searching, ranking and swarming-based content delivery of localized advertisements relevant to the user.

The vehicular environment presents interesting challenges and constraints not encountered in content delivery on the Internet. In particular, mobility and the limited/intermittent bandwidth cause sessions to be transitory and subject to frequent disconnections. Therefore, frequently, a push model system will be able to push only pieces of any content to individual vehicles making swarming an ideal choice for content delivery in these networks. In addition, unlike the wired Internet, in wireless scenarios only limited bandwidth is available and, hence, one must carefully evaluate the benefits of increasing communication costs when making any design decision.

In Section II, we give an overview of the operation of the ad service in a vehicular scenario. Section III describes the AdTorrent design. AdTorrent uses scoped query dissemination for discovering peers that have the relevant content. To evaluate the trade-off between the scope of the epidemic dissemination and the the query hit rate, we present a model for estimating the query hit rate given the scope of the query in Section **??**. We outline the related work in Section VI. Finally, Section VII concludes the paper.

II. System Overview

Vehicular Environment: While our targeted ad-delivery system is independent of specifics of the underlying vehicular ad-hoc network, we describe a typical environment for reference. The network consists of a set of nodes (vehicles) with similar computation and transmission capabilities, communicating through bidirectional wireless links between each other in a infrastructure-less ad-hoc mode of operation. There are wireless gateways at regular intervals providing access to the rest of the Internet using infrastructure support(either wired or multi-hop wireless). We do not assume any routing protocol running in the underlying network. While, in reality, nodes may or may not run the peer-peer application protocol, we can ignore the nodes not running the peer-to-peer application protocol and consider our network as only comprised of participating peers whose connectivity may depend on non-participating nodes in addition to node mobility, transmission medium characteristics etc. Nodes use TCP for reliable transfer of data and UDP for dissemination of index updates. As typical, CSMA/CA MAC layer protocol(IEEE 802.11a) providing RTS/CTS-Data/ACK handshake sequence for each transmission is used. Our vehicular wireless architecture is composed of two kinds of communications, namely, vehicle-vehicle and vehicle-gateway. Dedicated Short-Range Communication(DSRC) [4] is a short to medium range communication technology operating in the 5.9 GHz range can be used for vehicle-vehicle communication. For a more detailed description of the DSRC characteristics, we refer the reader to [5]. Communication between vehicles is over a low data rate connection. While this constraint depends on the radio technology used, currently, 802.11x devices will offer *goodput* of the order of a few hundred Kbps. Finally, we note that even though we expect the advertisements to be multimedia, the data delivery is not time-constrained.

Digital Billboards: In the digital billboard architecture, we envision the Access Points (APs) or gateways to continually disseminate advertisements relevant to the proximity of the gateway deployment to vehicles that traverse the coverage area. The advertisements can be simple text-based but can also be larger multimedia advertisements, for example, trailers of movies playing at the nearby theater, virtual tours of hotels in a 5 mile radius, or conventional television advertisements of local businesses. The economics of this digital billboard service are similar to its physical counterpart: the business owners in the vicinity will subscribe to the service for a fee. However, compared to the physical "static" billboards, the digital billboard offers a much richer media which is otherwise available only at a much coarser localization (e.g. local TV station) at a much higher cost. With the envisioned digital billboards, the physical wireless medium automatically induces locality characteristics into the advertisements. We note that a good application (e.g. driven by voice-commands) on the client (vehicle) side can ensure road safety (e.g. by making sure that the users see the Ad only if they actively search for it or indicate an interest in it).

AdTorrent: Leveraging the aforementioned Digital Billboard architecture, AdTorrent provides a location-aware distributed mechanism to search, rank and deliver content to the end-user (the vehicle). Every node that runs the application collects the advertisements and indexes the data based on certain meta-data which could be keywords, location and other information associated with the data. Upon a user indicating interest, a localized (hop-limited) query is broadcast. Each node upon receiving the query, goes through its local index and replies with the document identifier (TorrentID) and piece list relevant to that document. AdTorrent ranks the results and then obtains the content using swarming. In the next section we describe the details of the AdTorrent application design.

III. AdTorrent Design

We outline the primary design goals of *AdTorrent* as follows:

1) location aware torrent[1] ranking algorithm;
2) search should be simple, communication-efficient and robust, in presence of node failures and departures;
3) feasible commercial application on vehicular ad hoc networks that utilize unused channel;

There are three main tasks performed by our application apart from the continual aggregation of the ad data dissemination from the gateways. Namely, *search* for relevant ad-content, *query dissemination* and *content delivery*. We address each of these functions in the following sections.

A. Search

Search involves associating keywords with document identifiers and later retrieving document identifiers that match combinations of keywords. Each file is associated with a set of metadata: the file name, its format, genre(e.g. in advertisements). For some types of data, such as text documents, metadata can be extracted manually or algorithmically. Some types of file have metadata built-in; for example, ID tags on MP3 files.

[1]we use the terms : document and torrent interchangeably

Distributed Hash Tables (DHTs) have been proposed for distributed lookups. We do not use a DHT for distributed lookup, since it is well-known that DHT's are not very stable under high churn [2]. Our query dissemination mechanism aims to achieve robustness rather than communication efficiency.

AdTorrent uses a scoped query flooding mechanism. There is an inherent trade-off between the reliability/effectiveness of the search and the flooding overhead. Thus, the hop limit in the query flood is a key design issue. In the next section, we provide an analytical framework to estimate the hop limit based on the size of the cache, the file popularity distribution and the required performance.

Indexing: Vertical partitioning divides an index across keywords. Vertical partitioning minimizes the cost of searches. In our scenario, the number of queries will far outnumber the number of updates, since we assume the documents typically searched for, are not changing frequently. We add another optimization to the vertical partitioning. Our index is partitioned based on a *set of keywords*. This was first introduced in KSS [6]. The motivation of using a keyword set based indexing is the reduction of overhead in terms of query data information since the matched query results for multiple keywords can be remotely computed and the results returned, instead of the index hits of each individual keywords being returned to the query originator. The downside of this approach is higher cost of insert and storage.

However, every node tries to maintain information of the all the two-hop neighborhood of itself. Keys consists of SHA-1 hashes of the keywords sets. Along with the keys and the URL of the data, we also store additional meta-data associated with the data. The meta-data is stored in an index corresponding to each subset of at most K meta-data items. Each entry of the index contains: (1) the hash of the searchable sets of keywords as the index key, (2) a pointer to the data such as the URL of the data and, (3) meta-data associated with the data.

Placement: In a wireless scenario, it makes sense to colocate the index and the data corresponding to the index entry. This is to reduce the overhead of data discovery latency once the index for that data has been located.

B. Query Data Dissemination Optimization

Each node disseminates the content availability information in the form of a bloom filter. Bloom filter [1] is an efficient method to test for set membership. In our case, the bloom filter is constructed to test the keyword membership and consequently the document membership, for a particular node. A bloom filter is computed by each node based on the keywords related to the data, the node has stored. Since the data downloaded is only once every gateway encounter or if the node explicitly downloads some swarming torrent, hence the updates of bloom filter and dissemination is not very frequent. We now enumerate the basic steps of the algorithm.

The indexing scheme described above does not have a document ranking algorithm. The order of query results propagation and display is equally important for successful and timely

Algorithm 1 AdTorrent: Query Processing, Ranking and Content Delivery

```
user_input = search("A B C")

num_local_entries = lookup_local_index(hash(ALB),
hash(BC), hash(CA))
if (num_local_entries > k1)
    goto LookupDone
else
    /* Found < k1 local entries */
    /* not in the 2-hop neighborhood */
    num_remote_entries = scoped_flood( hash(XY), m )
    /* ∀ XY ∈ AB,BC,CA */
    After T1 seconds, if NO response, return NO
    If k1 entries are found then

LookupDone:
    /* now have k1 entries (local or remote in 1-4 hops) */
    send_udp_ctrl( Hash(XY))→METADATA( e.g. TorrentID)

    /* Collect meta_data after T2 */
    torrent_ranking(meta_data, paramos)

Step Final:
    swarm(TorrentID)
    /* returns a list of Peers & HopCounts*/
    /* ( this may be beyond the scope of the search) */
    decentralized_tracker()
    /* By allowing the list of Peers beyond the k-hop scope
of the search, we add some randomization */
```

dissemination in a VANET. This assumes further importance in VANET since the mobility of nodes might render some query results obsolete or irrelevant in short period of time. We incorporate a location metric in the document ranking scheme. One way to support the document ranking would be to score a document based on the following categories; (1) location, (2) max # of pieces, (3) stability of neighbors and (4) relevance of the torrentID to the Meta-Data queried. For brevity, we skip the description of this ranking.

C. Content Delivery

Once an accurate document ranking has been performed, the actual delivery of content can be done by swarming. One of the factors that determined the ranking of a document in the query results was the number of sub-pieces of the document that were available and the location of the pieces. Thus the torrent ranking guides the system to choose documents which are more amenable to swarming downloads. The vehicle now joins the existing BitTorrent-like stream to start getting pieces of the document from neighboring nodes. We propose to do this using our earlier work in [5]. Swarming allows us to be robust to node failures (cars going out of range or powering down) and efficient in terms of delivery (the cars form a sort of end-system-multicast tree).

IV. Hop limit selection

As discussed in the previous section, AdTorrent searches for relevant ad-content using a hop-limited query broadcast. Since setting a large hop-limit queries more nodes, a larger hop-limit improves the probability of finding the desired content and will likely increase the number of sources from which the content may be downloaded from. However, the gains in the quality of search results comes at the cost of significant increase in the messages sent per query in the network. Since only limited bandwidth is available in wireless medium, careful analysis of this trade-off between the quality of search results and the communication costs of the search is required. AdTorrent has many features but, for hop limit selection, we do not need to model all the details. A simplified model of the system that is sufficient for the required analysis is described next.

A. Model

We assume that a query flood of hop limit k reaches $M(k)$ peers. There are N unique files in the system (the term file represents any ad that would be downloaded), each with an associated request rate λ_i for file i per node (the request rates are *uniform* across nodes[2]). We assume that each file is of equal size[3]. Nodes have finite local storage space to store content files. We assume that the storage space at each node is equal[4] and has the capacity to store B files. When a node downloads a new content file and the local storage space is full, some older content must be deleted to make space for the new content file. We assume that the nodes use LRU cache replacement policy (i.e. delete the **L**east **R**ecently **U**sed file). A file may have multiple replicas in the search range and we assume that the download requests for file i are equally distributed over the replicas of file i available in the search range.

The notation for the various system parameters discussed is:

- k = Number of hops in the search query dissemination
- M = number of nodes in the search range
- N = number of unique files in the system
- B = per-node storage size in number of files
- i = File Id
- λ_i = request rate of file i per node
- $\lambda = \sum_{i=1}^{N} \lambda_i$

[2] We note that the search is localized over a small geographic area so it is not clear if node interests will be very different. Even when nodes have different interests, the uniformity of assumptions provides an adequate average case analysis i.e. a more accurate model that allowed for variations in request rate of a particular file across nodes would give similar results as our model with λ_i chosen to be the average per-node request rate per file.

[3] While file sizes can be different, cache replacement is typically implemented for fixed size data blocks so the only inaccuracy in the analytical model is on account of correlated requests for disk blocks which is not likely to alter the nature of our conclusions.

[4] We do not believe there will be a substantial difference in the storage capacity at each node. While small differences will not affect the result, larger differences can be accounted by rounding down the storage capacities and interpreting the results from the model as a lower bound on the hit rate.

- S = Swarming parameter
- j = Location in the local cache

To determine the probability of finding the desired content in the selected hop limit, we need to find the probability of finding the file at any one node (since the request rates are uniform across nodes, the probability of finding a file is the same across the network). Since each node is same in all respects, an analytical model of the network of LRU-managed caches can be constructed with a single cache that, in addition to serving the local requests, also serves requests from remote nodes. We model this cache from the perspective of a particular file, say, file i - all requests for file i move the file to the top-most position in the storage; a request for any other files moves file i down to one lower position. [7] presents an analytical framework for estimating the hit rate in stand-alone LRU-managed cache. By including the effect of remote requests, their model can be extended to model a network of LRU caches [8].

A critical component of this framework is $r(i, j, k)$, the rate at which file i is pushed down from position j to position $j + 1$ when the hop limit for scoped flooding is k hops. Let $p_{local}(i, j, k)$ be the probability of finding file i in top j positions in the cache when the hop limit is k hops. The probability of finding file i in local cache given a hop limit of k is then $p_{local}(i, B, k)$. $p_{local}(i, j, k)$ can be expressed in terms of $r(i, j, k)$ [7] by:

$$p_{local}(i, j, k) \approx \frac{r(i, j, k)}{\sum_{i=1}^{N} r(i, j, k)}$$

At steady-state, the push-down rate for file i from position j to $j+1$, $r(i, j, k)$, must equal the rate at which file i is brought into top j positions of the LRU stack (otherwise the probability of finding the file in these top j positions becomes unbounded). This conservation of flow principle helps us compute $r(i, j, k)$. File i is brought into top j positions under two conditions: (i) a local request for file i when file i is not in top j positions: the file may be brought to the top position from positions $j + 1 \cdots B$ of the local cache if it is available there or it may be brought from a remote node (if a node within the search range has the file), this is $r_{local}(i, j, k)$; (ii) a remote request for file i: since the file i is not in top j positions, it must be in the remaining $j + 1 \cdots B$ positions in the local cache for it to show up in top j positions on a remote request. Thus, we can write the following equations:

$$r_{local}(i, j, k) = \lambda_i[1 - p_{local}(i, j, k)]$$
$$[1 - (1 - p_{local}(i, B - j|j, k)(1 - p_{remote}(i, j, k)]$$

$$r_{remote}(i, j, k) = \lambda_i[1 - p_{local}(i, j, k)]$$
$$p_{local}(i, B - j|j, k)$$

where

$$p_{local}(i, B - j|j, k) = \frac{p_{local}(i, B, k) - p_{local}(i, j, k)}{1 - p_{local}(i, j, k)}$$

and

$$p_{remote}(i, j, k) = [1 - (1 - p_{local}(i, j, k))^{M(k)}]$$

A node sends out a file request only when it does not have the file. Thus, the rate at which the other $M(k) - 1$ nodes send a file request for this file to the peer-to-peer network is $\lambda_i[1 - p_{local}(i, B, k)]$, where $p_{local}(i, B, k)$ is the probability that the file i is available at a node. The nodes that have file i in their cache satisfy these requests for file i sent to the peer-to-peer network. Assuming that the requests are uniformly distributed over the nodes that have the file, the request rate for file i served by a node that has file i on account of requests from other nodes equals $\frac{(M(k)-1)r_{remote}(i,j,k)}{M(k)p_{local}(i,B,k)}$ [5]. Thus,

$$r(i, j, k) = r_{local}(i, j, k) +$$
$$\frac{(M(k) - 1)r_{remote}(i, j, k)S}{M(k)p_{local}(i, B, k)}$$

Starting with $p_{local}(i, 1, 1) = \lambda_i$ we can iteratively solve the above equations until the value of $p_{local}(i, B, k)$ converges. The complexity is O(KB) and, in our computations, the value of p converged in only a few iterations.

Given $p_{local}(i, B, k)$, we can compute the hit rate for file i in the k-hop neighborhood as $P(i, B, k) = [1 - (1 - p_{local}(i, B, k))^{M(k)}]$. Therefore, the overall hit rate (across all searches) is:

$$P(B, k) = \sum_{i=0}^{N} \frac{\lambda_i}{\lambda}[1 - (1 - p_{local}(i, B, k))^{M(k)}]$$

Among the inputs to our model, the cache size B and the hop limit k are the design choices while λ_i, the file request rate distribution, and $M(k)$, the number of nodes in the k-hop neighborhood, are inputs that the system designer must provide for the specific application scenario being investigated.

For a VANET scenario, our real-track mobility model [3] is an ideal choice. The model can be run with the expected user density and an empirical expression of $M(k)$ can be obtained from the collected statistics. In most web and multimedia applications, different content has been found to have widely varying popularities. We expect the same in our Ad-content distribution scenario. Skewed file popularity distribution is typically modeled by a Zipf-distribution and we also use the same model in our investigations.

V. Results

The results are shown in Figure 1 and Figure 2. In Figure 1 the growth rate of $M(k)$ with k is quadratic as may be possible in dense, urban scenarios of up to a certain distance (modeled as a uniformly distributed grid topology). In more sparse settings, the growth rate might be sub-quadratic. We ran our vehicular mobility model [3] and empirically derived a growth distribution of $M(k) \sim \alpha k^{1.4}$. Figure 2 shows the results for such a topology. As suggested in the previous section, the model used Zipf-distributed file request rates.

[5]This term can be multiplied by S, the swarming parameter, to include the effect of swarming

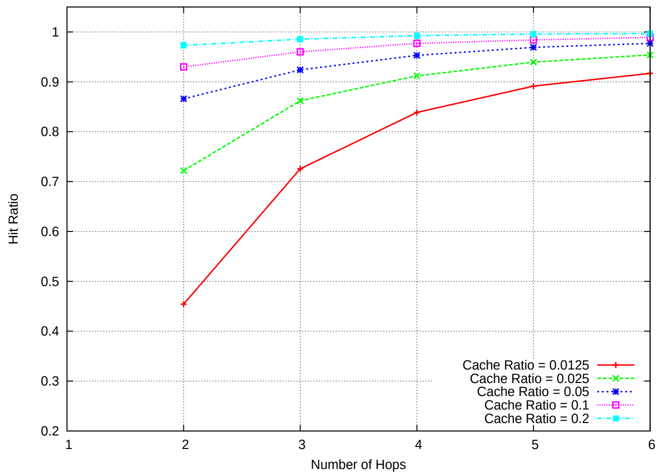

Fig. 1. **Distribution of Hit Rates with respect to Hop Count for varying Cache sizes with** $M(k) \sim \alpha k^2$

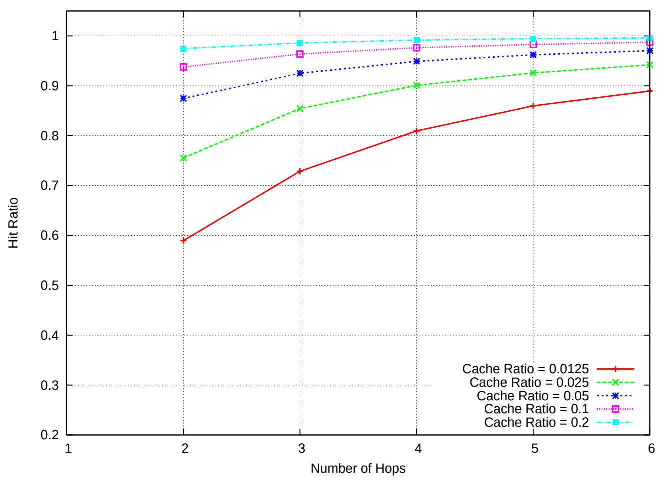

Fig. 2. **Distribution of Hit Rates with respect to Hop Count for varying Cache sizes for sparse node growth** $M(k) \sim \alpha k^{1.4}$

In Figures 1, 2, the cache ratio refers to the size of the individual node cache with respect to the total number of files in the network. So for example, a cache ratio of 0.1 means, an individual nodes' cache can store 10% of the total files in the network.

We find that with increasing hop count, the marginal gain in hit rates diminishes. This effect is even more pronounced as the cache ratio increases. Our analytical framework can be used to tune the query flood to achieve required levels of hit rates, and consequently the performance of AdTorrent by suitably adjusting the hop limit of the query flood. So, for example, if 80% hit rate was a satisfactory level of performance measure, our results suggest that a query hop limit of 4 will yield satisfactory performance irrespective of the cache size (as long it is above a certain threshold).

VI. Related Work

This section summarizes previous work related to co-operative data transfer protocols for the wired settings as well as vehicular environments. BitTorrent is a popular [**?**] file-sharing tool, accounting for a significant proportion of Internet traffic. There are two other peer-peer bulk transfer protocols namely, CarTorrent and Coopnet. CarTorrent [5] is a recent work that extends the BitTorrent protocol to the vehicular networks scenarios addressing issues such as intelligent peer and piece selection given the intermittent connectivity and limited bandwidth of the wireless medium.

Peer-to-peer networking in cooperative mobile environments has been proposed by several others. However, the constraint of limited buffers at the peers is discussed by very few others. [8] analyzed epidemic information dissemination to support web accesses with limited buffers per peer. Our analytical model to study the trade-off between the query hop limit and the overall hit rate is very similar to theirs. Since our problem setting is different from theirs, some of our assumptions are different. As discussed earlier, our model (as well as that of [8]) is an extension of the analytical model for a stand-alone LRU cache given in [7].

VII. Conclusion

In this paper we presented a novel application involving search and location aware content delivery(in our case advertisements/deals) to the nodes. We proposed a efficient keyword search on this content overlay. Our work is a first step in the design and eventually the implementation of AdTorrent application. We presented an analytical model of the epidemic query dissemination and evaluated the impact of the scope of the query dissemination on the hit rate. It was interesting to see that the incremental gain from increasing the scope of the query flood beyond 4 hops was minimal. System designers can use our analytical framework to estimate the required cache size and scope of the query dissemination based on user performance requirements.

References

[1] B. Bloom, Space/time tradeoffs in hash coding with allowable errors. CACM, 13(7):422-426, 1970.

[2] S. Rhea, D. Geels, T. Roscoe, and J. Kubiatowicz, Handling Churn in a DHT, USENIX 2005

[3] B. Zhou, K. Xu and M. Gerla, *Group and swarm mobility models for ad hoc network scenarios using virtual tracks,* In Proceedings of MILCOM 2004

[4] *Dedicated Short Range Communication Architecture,* www.astm.org/SNEWS/MAY_2004/dsrc_may04.html

[5] A. Nandan, S. Das, G. Pau, M.Y. Sanadidi and M. Gerla, *Cooperative Downloading in Vehicular Wireless Ad Hoc Networks,* In Proceedings of Wireless On-Demand Networks and Services, St. Moritz, Switzerland, Jan 2005.

[6] Omprakash Gnawali, *A Keyword-Set Search System for Peer-to-Peer Networks,* Masters Thesis, Massachusetts Institute of Technology, 2002.

[7] A. Dan and D. Towsley, *An Approximate Analysis of the LRU and FIFO Buffer Replacement Schemes,* in Proceedings of ACM SIGMETRICS 1990.

[8] C. Lindemann and O. P. Waldhorst, *Modeling Epidemic Information Dissemination on Mobile Devices with Finite Buffers,* in Proceedings of ACM SIGMETRICS 2005.