PAVAN: A Policy Framework for Content Availabilty in Vehicular Ad-hoc Networks

Shahram Ghandeharizadeh[1], Shyam Kapadia[1], Bhaskar Krishnamachari[1,2]
[1]Department of Computer Science
[2]Department of Electrical Engineering
University of Southern California
Los Angeles, CA 90089, USA
{shahram,kapadia,bkrishna}@usc.edu

ABSTRACT

Advances in wireless communication, storage and processing are realizing next-generation in-vehicle entertainment systems. Even if hundreds of different video or audio titles are stored among several vehicles in an area, only a subset of these titles might be available to a given vehicle depending on its current location, intended path, and the dynamics of its ad-hoc network connectivity. The vehicle's entertainment system must somehow predictively determine which titles are *available* either immediately or within the future δ time units, so that the user can select a title to view. The available title list must seek to satisfy the user by striking a delicate balance between showing far fewer titles than can actually be accessed and showing too many titles that cannot be accessed. In addition to defining this availability problem, we make two key contributions. First, a two-tier system architecture which leverages the low-rate cellular infrastructure to exchange control messages and facilitate delivery of large data transfers using the ad-hoc network of vehicular devices. Second, PAVAN as a policy framework for predicting the availability of a title. We describe several variants of PAVAN which incorporate information based on a Markov mobility model, spatio-temporal look-ahead, and title replications. Our results demonstrate that the quality of PAVAN's predictions is critically dependent on degree of title replication, as well as its display time relative to the trip duration. When degree of replication is below a certain threshold, PAVAN with content density information and the predictive mobility model is shown to provide the best overall performance.

Categories and Subject Descriptors

C.2.1 [Network Architecture and Design]: Wireless Communication

General Terms

Design, Performance, Verification

Keywords

VANETs, C2P2, PAVAN, Content Availability, Evaluation, Utility Models

1. INTRODUCTION

Rapid advances in wireless technology are offering network bandwidths in the order of tens of Megabits per second (Mbps). These bandwidths enable continuous media applications that stream audio and video clips among an ad-hoc network of mobile devices. One example is entertainment systems for vehicles that complement the existing wired multimedia network within each vehicle. These are named Car-to-Car Peer-to-Peer (C2P2) devices [6]. In addition to a wireless interface, a C2P2 may consist of gigabytes of storage and a fast processor. It may use its local storage to cache different titles. C2P2 devices form an ad-hoc network to exchange audio and video clips in support of applications such as video-on-demand. A C2P2 device may participate in three different roles simultaneously. First, it may display a title. Second, it may stream a clip from its local storage for display at another C2P2 device. Third, it may route a stream from a data producing C2P2 device to another C2P2 device that is displaying this data.

Typical components of a C2P2 system provide the following functionalities. First, a discovery component determines those titles available in δ time units and vehicles containing these titles. We term δ as the availability latency of a clip. A title is available immediately when its δ value is zero. Second, an interface showing this list of titles and their availability latency to a user, facilitating title selection. Once a user selects to display a title, an admission control component [5] (third component) ensures availability of both resources and the referenced data to deliver this data in a manner that supports a display free from disruptions and delays termed hiccups. Fourth, a data delivery scheduling technique utilizes resources as a function of time to deliver data to a displaying C2P2 device. This component, may switch between several candidate servers containing the referenced clip based on their proximity, current availability of resources, and network conditions. Fifth, a mobile ad-hoc network routing protocol facilitates delivery of data between C2P2 devices. Example protocols are DSR [10], ZRP [7], CEDAR [13]. Finally, a QoS component may monitor whether a target C2P2 is achieving its desired QoS level. This component may initiate construction of additional replicas of a title to enhance QoS metrics such as startup latency, i.e., the delay observed from when a request is issued to the onset of its display.

A policy that determines availability of titles is the focus of this study. It is important to predict the list of available titles to empower a user to make an informed selection. This prediction is termed the 'availability' problem and must strike a delicate balance between showing far fewer titles than can actually be accessed and showing too many titles that cannot be accessed. If it is too conservative and advertises a few titles than actually available then the user may not pick a title because none are interesting. On the other hand, if a C2P2 shows many titles only to reject a user's request because the referenced title is not available then the user might become frustrated and stop using the system. Moreover, this may result in system inefficiencies by issuing requests for titles that are not available.

The primary contributions of this paper are two-fold. First, we present a hybrid system architecture that leverages the existing cellular infrastructure to facilitate exchange of continuous media using the ad-hoc network of C2P2 devices. Second, we introduce PAVAN as a novel policy framework for addressing the availability problem. This policy employs a Markov mobility model to compute the list of titles along with their respective availability latencies. A Markov mobility model considers correlated transportation patterns induced by roads and highways. Similar Markov mobility models have been used effectively in prior studies of cellular networks (e.g., [8]). Our architecture enables each C2P2 device to invoke PAVAN independent of other C2P2 devices, and specify a utility model for the clips computed to be available. A utility model associates a value for false-positives, false-negatives, and true-positives produced by a policy.

Two general areas of computer science literature appear relevant to PAVAN. We describe them in turn and show how PAVAN is novel and different from these previous approaches. First, availability of data has been studied extensively in the areas of redundant arrays of disk drives [12], parallel multi-disk and shared-nothing relational database management systems [4], and continuous media servers [2]. These techniques use redundant information to recover data lost due to failures. Redundant data might be parity bits, replicas of data, or a combination of these two. Similarly, in our architecture (see Section 2), we assume data is replicated to increase its availability. The key difference is that network connectivity is dynamic and we use mobility models to estimate when a replica of a clip is available.

Second, the discovery techniques of peer-to-peer networks resemble PAVAN. These techniques can be categorized into two groups: structured and unstructured. The structured techniques employ either a centralized index, e.g., Napster, or distributed hash table, for e.g., Chord [14], to discover available data. With an unstructured discovery technique, e.g., Gnutella [3], a device might discover available clips by performing a constrained expanding ring search using a simple query probe to determine which titles are available at the current moment. Both structured and unstructured approaches employ the current snapshot of the network, ignoring changes that occur along the time and space dimensions. Due to the dynamic nature of the C2P2 network, these approaches would yield poor results. PAVAN is novel

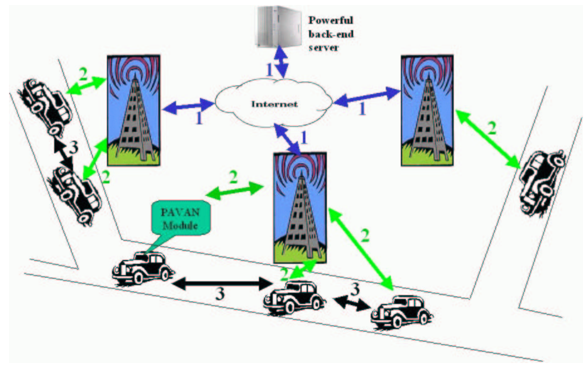


Figure 1: A hierarchical architecture.

because it considers mobility of C2P2 devices and the dynamic changes to the network.

Resource discovery in MANETs is addressed in [9]. It proposes a content architecture that assumes a node proactively determines resources within its vicinity. A node maintains a few distant nodes, termed "contacts", for resource discovery beyond its vicinity thereby taking advantage of the small world phenomena. This architecture is designed for small data transfers where it can significantly reduce the control overhead of resource discovery compared to broadcast flooding. In our application, a client's data transfer is large and long-lived, amortizing the cost of any control overhead within our hybrid architecture. A unified framework for resource discovery for large scale ad hoc networks is provided in [11]. This study considers QoS and employs self organized discovery agents that partition the network into domains. It provides for efficient retrieval by maintaining a hash index of the resources available within the network.

The rest of this paper is organized as follows. Section 2 presents a framework for the availability problem and provides its details. Section 3 describes PAVAN, its possible input and output parameters, and the methodology used to solve the availability problem. Section 4 evaluates alternative variants of PAVAN. It shows PL_t (see Table 1) provides the most accurate availability latency. Section 5 presents brief conclusions and our future research directions.

2. ARCHITECTURAL FRAMEWORK AND OVERVIEW

A vehicular ad-hoc network such as C2P2 may potentially cover a large geographical area, such as a metropolitan city. At such large distances, discovering titles available for viewing becomes a very challenging problem indeed. It is easy to see that on-demand flooding/simple query-based approaches to resource discovery within the ad-hoc network will not scale well.

Our solution is to adopt a hierarchical architecture that also leverages the existing large scale heterogeneous wired-wireless cellular network infrastructure. The infrastructure aids in the collection of localized aggregate information that can be used to distribute the decision making. Our two-tiered architecture, shown in Figure 1, consists of separate data and control networks. The data network (edges labeled 3) consists of the C2P2 vehicular ad-hoc network, facilitat-

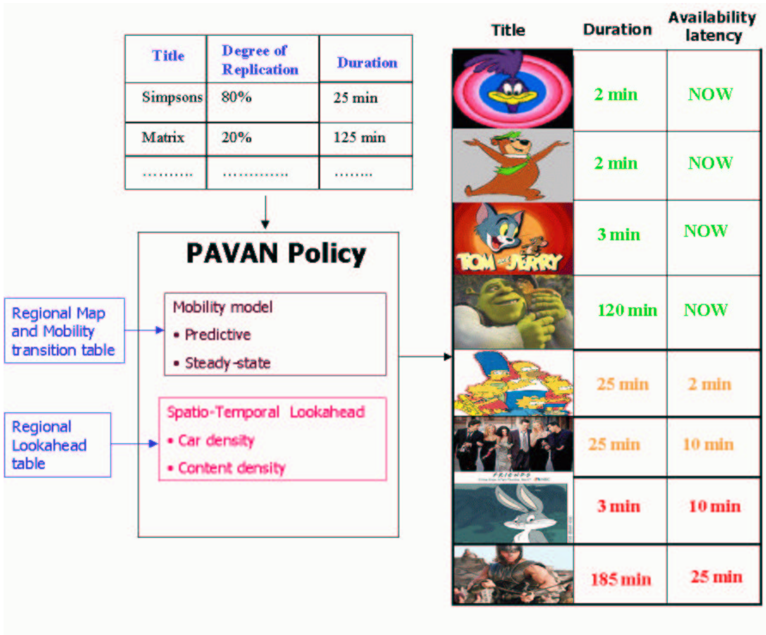


Figure 2: An overview of PAVAN, its inputs and output.

ing the video streaming and inter-node data exchanges. The control network (edges labeled 2) is a low data rate cellular network infrastructure, with base stations dividing a large geographical area into localized cells.

This architectural framework also localizes the communication in the data network mostly to within a cell. Since the display of a typical title would take a C2P2 through several cells, the availability problem essentially consists of determining if sufficient replicas of the title will be present in C2P2s in each cell in this intended path. This in turn requires sufficient information about nearby (regional) cells to be made available to the availability policy module, which resides in each individual C2P2 device.

We now provide an overview of the various components of this architecture, see Figure 1.

Data network consists of the vehicular ad-hoc network of C2P2 devices. The system storage of content is distributed among the various C2P2 devices within this network. The links in this network are all high bandwidth (tens to hundreds of Mbps each) for streaming of different titles. At each instant, the communication is localized so that it is between nodes that are moving within the same cell. We assume that every C2P2 in the same cell is network connected. A typical path between two devices in the same cell may be multi-hop. This is because the range of a cellular base station is almost certainly much larger than the range of high bandwidth network devices (e.g., 802.11a [1]) employed by C2P2 devices. The number of hops is expected to be short, on the order of 3 to 4 hops.

Control network provides three key functionalities: (i) monitoring and collection of pertinent content and mobility information from individual car's C2P2 devices to the base station; (ii) regional consolidation and storage of this information into maps, mobility models and content information by nearby base stations and remote servers within the cellular network infrastructure; and (iii) periodical updating of pertinent regional map, mobility, and content information of C2P2 devices within each cell. A base station may perform the last step by broadcasting information. Control messages are typically small and require a low data rate in the order of tens of Kilo bits per second (Kbps).

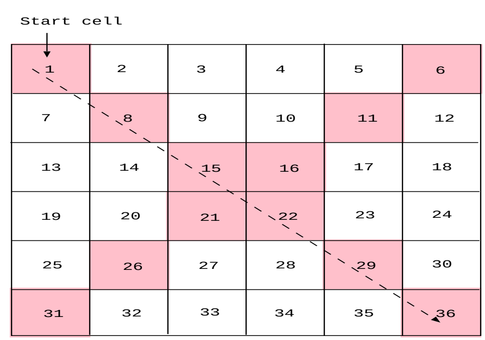


Figure 3: An example 6x6 map.

We now examine the different components of the control information being collected and broadcasted in each cell, briefly.

a. Regional Maps and Mobility Model: Several cells near each other can be grouped into a single regional map. Figure 3 illustrates such a map for a system with square cells. A base station locally monitors information about the number of C2P2 devices in its cell, which cell a device came from, and which cell a device is moving towards. This information from nearby cells is then used to construct a Markov inter-cell mobility transition table over this regional map. Time is discretized and the mobility model assumes that at each step, a C2P2 may move to any of its 8-adjacent neighbors or stay in the same cell[1]. Note that the Markov Transition table has only R^2 entries, where R is the number of cells in the local region (which may be typically about 35-50).

b. Title Replication Table: Based on regional as well as global input, information is also maintained about the ID and duration of all titles, as well as their replication levels. This table would have I rows, one for each possible title. While I might be potentially in the order of hundreds or thousands, note that each row is small and in the order of tens of bytes.

c. Regional Lookahead Table: Based on current data, a regional lookahead table is also created that shows which titles are available in each cell, and how many C2P2s are in each cell. This table would have R entries, each with about $I + log(C)$ bits of information, where C is the maximum number of C2P2s in a given cell.

The feasibility of this architecture is crucially dependent on the fact that these pieces of control information are not bandwidth-intensive. For instance, for typical numbers such as $R \approx 50, I \approx 1000, C \approx 100$, a conservative back of the envelope calculation suggests that this would amount to no more than 100 Kilo Bytes (KB). For reasonable refresh frequencies, since the same information is broadcast to all C2P2s in the cell, this requires downlink bandwidth in the order of a few Kbps in each cell. Similarly, the data being collected on the uplink at each cell from each C2P2 when it first enters the cell is only a few hundred bytes of data (containing only a bitmap of I bits indicating its available titles, and the ID of the cell it came from).

The information provided by these broadcasts to individual C2P2s is of an aggregate and regional nature. It is this information that each vehicular C2P2 device uses to make completely distributed decisions regarding the availability of titles along its intended path. The title availability decision making resides solely in each device. This distribution of information and decision making ensures that there is no single point of failure. We shall discuss our availability policy framework that makes use of this information in the next section.

3. PAVAN

We now present the details of PAVAN, a Policy framework for Availability in Vehicular Ad-hoc Networks. The output of PAVAN is the available title list which acts as a user menu showing all titles predicted to be available and also a predicted time after which they will be available to the client. An example output for PAVAN is shown in Figure 2. The prediction and presentation of the availability latency empowers users to make informed decisions.

The accuracy of PAVAN's output depends on its provided information, i.e., its input. As noted in the previous section, there are three essential pieces of information that can be provided as input to PAVAN: the title replication table, the regional mobility table, and the regional look-ahead table. The global title replication table (shown in Figure 2) is provided to all variants to PAVAN. These alternatives are different depending on whether PAVAN is provided with either the mobility table, lookahead table, or both. Intuitively, the "richer" the information input, the closer the output list is to the ideal list. There is a trade-off between obtaining richer information for the policy decision-making against the overhead of having this information broadcast from the base station to C2P2 devices.

PAVAN Policy	Input information
SS_{only}	Steady-state mobility model
SSL_t	steady-state mobility model and density of the contents in the C2P2s within a pre-specified lookahead
PL_t	predictive mobility model and density of the contents in the C2P2s within a pre-specified lookahead
PL_r	predictive mobility model and density of the C2P2s within a pre-specified lookahead

Table 1: Four variants of PAVAN.

An additional input to PAVAN is the maximum delay tolerable by a client. This provides an upper bound on the availability latency.

In its most naive and simple form, a PAVAN variant may only use the global replication table for producing the output list of titles. Based on this table and a threshold x, it decides which titles in the system are available to the client. If degree of replication for clip $i > x$ then mark clip i as available. Otherwise, clip i is marked as unavailable.

Note that the value of x is not known, hence it can vary from 0 (no C2P2 has clip i) to 100% (every C2P2 has a copy of clip i). In the absence of any other information, PAVAN's predicted list of titles can simply be obtained by making a horizontal cut across the global replication table (sorted on degree of replication) at the appropriate threshold and advertising all titles above that threshold as available. However, information about the mobility model, the content density and C2P2 density is completely ignored which makes its performance poor. Hence we ignore this variant from further discussion.

PAVAN, in addition, may be given mobility information that captures the movements of the C2P2s. We now describe the mobility model used, which is probabilistic and Markovian in nature. Each cell of the map constitutes a state. These states are self-contained and a transition from one state to another is independent of the previous history of a C2P2 in that state. Depending on the configuration of the map, as mentioned earlier, the mobility model is weighted toward the gray cells. The aggregate of the transitions from each cell (state) to every other state gives the probability transition matrix Q. $Q = [q_{ij}]$ where q_{ij} is the probability of transition from state i to state j. Using Markov chains it is possible to estimate the distribution of the steady state probabilities of C2P2s in the various cells. The steady state probabilities are independent of the initial location of the C2P2s in the cells. Note that these probabilities are also independent of the identity of each individual C2P2 and represent the distribution of the belief of the location of the C2P2s across the various cells at equilibrium.

In this way, the mobility model (MM) provided to PA-

[1]In this paper, for simplicity, we assume that the probability of movement is highly weighted toward the gray cells. For example, in Figure 3, cell 1 has 3 neighbors 2, 7, and 8. A C2P2 in cell 1 has a higher probability of either remaining in cell 1 or moving to cell 8 because they are both gray. The probability of moving to either cell 2 or 7 is lower because they are not gray.

VAN is either predictive (P) where the Q matrix is used in each step, or steady-state (SS) where only the equilibrium probabilities are used. The equilibrium probabilities are obtained by solving the equality $\Pi = \Pi * Q$, where Π is the vector representing the steady-state probabilities of being in the various cells (states).

We now describe the alternative variants of PAVAN. SS_{only} provides PAVAN with both SS and the global replication table. This means the location of all C2P2s is the same and is given by the steady-state matrix Π_{ij}. Since no information about the contents of the C2P2s is provided, SS_{only} uses the global replication table and assumes that a C2P2 contains the various titles governed by this global distribution. Aggregating this information for all C2P2s yields the Title location matrix T. Note that T has identical rows for each C2P2 since no information is known about their contents.

For each step along the path, the following procedure is applied:

Algorithm 3.1: PROCA($Steps, C2P2s, titles$)

for $step \leftarrow 1$ **to** $Steps$
  $cell\_id \leftarrow clients\_current\_cell$
  $Conf \leftarrow 0$
  for $i \leftarrow 1$ **to** $C2P2s$
    if (C2P2 i located in $cell\_id$)
      for $j \leftarrow 1$ **to** $titles$
        if (C2P2 i contains title j)
          $Conf(j, step) += \Pi(i, cell\_id) * T(i, j)$

Hence, for each title, this procedure yields the 'Confidence' of that particular title for that step along the journey of the client. The higher the confidence, the higher the predicted availability of the title at that step. At the end, the confidence for each title across all steps is aggregated into a metric that is then mapped into the list of available titles. This is achieved using the following procedure:

Algorithm 3.2: PROCB($Steps, titles$)

for $i \leftarrow 1$ **to** $titles$
  if $Conf(i,j) \geq m$ for every step j, $1 \leq j \leq$ Steps
    $Agg\_metric(i) = \sum_{j=1}^{Steps} Conf(\text{i,j})$
  else $Agg\_metric(i) = 0$

If, for title i, $Agg\_metric(i) > 0$ then that title will appear in the client's available list. In all our experiments we chose a value of $m = 1$. Intuitively, if the title has a $Confidence < 1$ at even one step, on an average less than 1 copy of that title exists at that step. Hence the client may not find a copy of that title at that step. Such a title is not shown in the predicted list. It should be noted that values of m less than 1 result in optimistic predictions, while values of m greater than 1 result in conservative predictions.

In addition to the mobility model and the global replication table, PAVAN can be provided with the density of C2P2s in a region and their contents. This can be limited to a specific area defined by a Spatio-Temporal Lookahead (STL) parameter.

3.1 Spatio-Temporal Lookahead (STL) parameter

The SSL_t variant of PAVAN consumes the global replication table, SS, and the contents of those C2P2s within a

3	3	3	3	3
3	2	2	2	3
3	2	1	2	3
3	2	2	2	3
3	3	3	3	3

Figure 4: The numbers in the cells indicate the STL value for the shaded cell numbered 1.

fixed geographical area defined by STL. A STL[2] value of k encompasses all k adjacent cells. When STL is 0, SSL_t is similar to SS_{only}. Figure 4 shows an example 5x5 map with the client occupying the shaded cell. This figure shows STL values of 1, 2, and 3. As one increases the value of STL, a client obtains information about additional cells that are farther away. Note that a cell is assumed to have eight adjacent neighbors. SSL_t enables a C2P2 to incorporate the content of all C2P2s in k adjacent cells into T. The remaining C2P2 devices are assumed to contain titles as per the global replication table.

The PL_r variant of PAVAN considers the predictive mobility model (P), the density of the C2P2s within the STL and the global replication table to produce the title availability list. As the value of STL increases, the client obtains more information about the number of C2P2s in the various cells. When STL spans all cells, the client obtains information about the number of C2P2s in each cell of the entire network.

For a given client, PAVAN knows the location of C2P2s within STL adjacent cells. For all the other C2P2s, their location is equally likely to be a cell outside the STL in the map. This combined information about the initial positions of the C2P2s yields the initial location matrix L. At each step, we compute product of L_i and Q_i where i indicates the step under consideration, Q is the transition probability matrix defined by the mobility model, and the initial value of $L_i = L$.

Note that the contents of the C2P2s are not known, hence the Title matrix T is calculated according to the global replication table. The L_i and T matrices are used with procedures PROCA (replacing Π by L) and PROCB in order to obtain the predicted list of available titles.

Finally, PL_t denotes the variant of PAVAN with the following inputs: the global replication table, P and L_t (denoted by PL_t). When $k = 1$, PL_t is provided with the content of C2P2s in its current cell, termed start cell. PL_t assumes the remaining C2P2s are equally likely to be in other cells of the network besides the start cell. This yields the initial C2P2 Location matrix L. Again, we compute product of L and Q_i at step i to obtain location matrix L_i at that step. Note that since the list of titles assigned to some of the C2P2s is known, namely the C2P2s present in the start cell, we incorporate that information in T. PL_t

[2]We use k to denote the value of STL.

Model	Weight(w_1) of a_{10}	Weight(w_2) of a_{01}	Weight(w_3) of a_{11}
1	0	0	1
2	0	-5	1
3	-1	0	1

Table 2: Three utility models to evaluate alternative variants of PAVAN.

assumes other C2P2s have the titles distributed as per the global title replication distribution. Hence, in this case, the rows of T need not all be the same. When $k > 1$, the client obtains precise information about the density and the contents of the C2P2s in the cells that are reachable within a distance of k at the current instant. Using this information the client obtains the Location matrix L_i at each step i using $L_i * Q_i$ where initially $L_i = L$. Similarly the T matrix is obtained where the information of the contents of all the C2P2s within the STL is known. L_i and T can be input to PROCA (again replacing Π by L) and PROCB to obtain $Agg\_metric(i)$ which is then converted into the client's available titles list.

4. PERFORMANCE EVALUATION

The experimental set-up consists of a $6*6$ map as shown in Figure 3. The mobility model is weighted toward the diagonal both from left to right and vice-versa (due to gray boxes). Assume that the client starts from cell 1 and travels along the path {1, 8, 15, 22, 29, 36}. Numbers in the bracket indicate the sequence of visited cell ids. At the start of a client's journey, each variant of PAVAN retrieves its required information from the control network. Each variant of PAVAN (see Table 1) produces a different predicted list.

Initially all C2P2s are distributed uniformly across the cells in the map. This is determined by a random initial seed. The distribution of titles across C2P2s is also chosen to be uniform. At each step, depending on the current C2P2 location, the C2P2 moves to one of its adjoining cell (including itself) as governed by the mobility model. Another seed determines the choice of which cell a C2P2 moves to. Each C2P2 performs six transitions according to the mobility model. The intersection of C2P2s with the cells along the path yields the actual confidence values for a particular title seen in a particular run of the simulation. For each run, a different random seed is used starting from the same initial position. For each run, at each step, the client obtains the exact distribution of titles in the network and the corresponding confidence values for each title. These values are then translated to a list of titles for this particular run (actual list) using the same procedure PROCB (see Section 3). For each run, the predicted list is compared with the actual list and the utility models presented later in Section 4.1 depict the differences. We performed the comparisons for several runs starting from the same initial C2P2 positions. Next, we varied the initial C2P2 positions by changing the initial seed. Specifically, we chose 100 different initial seeds and for each of these we ran the simulator 100 times. Thus, all presented results are averages of 10000 simulations.

4.1 Utility Models

We used three different utility models to quantify the quality of lists computed by different variants of PAVAN. These modes assign a different weight to the average number of false negatives (denoted a_{10}), false positives (denoted a_{01}), and true positives (denoted a_{11}). A false negative is a title present in the actual list but not in the predicted list. A false positive is a title present in the predicted list but not in the actual list. Finally, a true positive is a title present in both the actual and predicted lists.

All utility models are represented as:
$U = w_1 * a_{10} + w_2 * a_{01} + w_3 * a_{11}$
We implement the alternative utility models by assigning a different weight to a_{10}, a_{01}, and a_{11} (see Table 2). These models are as follows. Model 1 depends on those titles that appear correctly in both the actual and the predicted lists. So its utility value ranges from 0 to 1.

Model 2 severely penalizes those titles that appear in the predicted list but not in the actual list. It assumes that a user would be greatly dissatisfied by choosing such titles because they are not available. The utility of this model ranges in value from -5 to 1.

Model 3 penalizes those titles that appear in the actual lists but not in the predicted ones. These available titles cannot be selected by a user because they are not predicated as available. The utility of this model ranges in value from -1 to 1. Note that the penalty for these false negatives is not as significant as false positives (compare Models 2 and 3).

4.2 Results

In our experiment we used 200 C2P2s, unless stated otherwise, and 16 titles with unique ids 1, 2, $\cdots$ 16. The percentage degree of replication of a title with id i is given by:

$$Title-rep(i) = \begin{cases} 2*i & 1 \leq i \leq 10 \\ 20+5*(i\%10) & 11 \leq i \leq 16 \end{cases} \quad (1)$$

This means title 1 has 4 copies, title 2 has 8 copies, and so on until title 10. Title 11 has 25 copies, title 12 has 30 copies, and so on until title 16. Replicas of a clip are assigned to C2P2s randomly. A C2P2 may contain several different titles, but only one copy of a certain title.

The map used in the experiment was similar to Figure 3 and the client moved along the diagonal from the top left to bottom right cell. We choose this path because it demonstrates the tradeoffs associated with different variants of PAVAN quite nicely. In the following, we present the key lessons learnt from our simulation experiments.

Figure 5 presents a comparison of alternative variants of PAVAN. The graphs represent the utility values as a function of the different degrees of replication of the movie titles. The predicted lists generated by PAVAN in all cases (where applicable) were calculated using the largest STL value (here, k=6, the length of the path of the client).

Lesson 1: As the degree of replication increases beyond a certain threshold all the variants of PAVAN start showing similar utilities. Two factors impact this observation. First is the degree of title replication. Second is the predictive nature of a specific PAVAN policy. We describe each in turn.

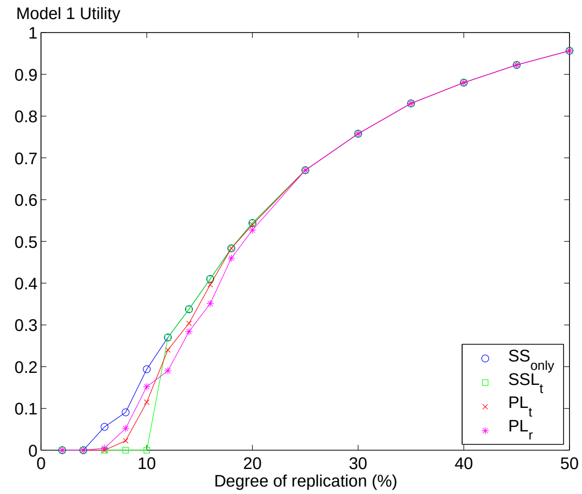
While for model 1, this replication threshold is 20%, it is approximately 50% with model 2. The general trend indicates that as the degree of replication increases, the utility value also increases and converges toward 1 (maximum utility value for model 1 and 2). This is because with the increase in the degree of replication, all variants of PAVAN find at least one copy of the title along the path of the client at each step, increasing the average number of true positives (a_{11}) to 1. Hence information provided to PAVAN in terms of the L_r and L_t does not yield better results since the global replication table, which is the base-line input to PAVAN, dominates the titles shown in the predicted lists. Thus both false-positives and false-negatives contribute an insignificant amount toward the final observed utility (for all models).

The predictive nature of a specific PAVAN policy impacts the utility observed with a model. With model 1, below 20% degree of replication, both SS_{only} and SSL_t perform marginally better than both PL_t and PL_r. This is because even for lower degrees of replication the SS variants in the absence of any C2P2 density or content density information always over-predict increasing both the true-positives and the false-positives while reducing the false-negatives. Since utility model 1 only considers true-positives, it enables SS variants to marginally outperform the other two alternatives. Hence the curves for model 3, which penalizes those clips that are present only in the actual list, becomes similar to that of utility model 1 and are not presented here.

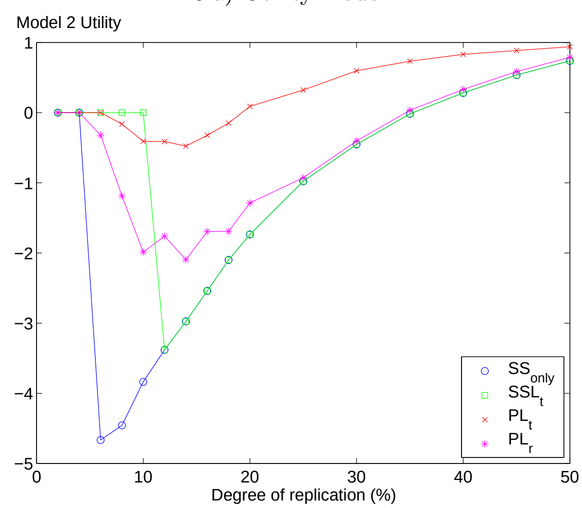
Model 2 penalizes those titles that are present in the predicted but not in the actual list. It is seen that, in general, PL_t outperforms the other variations of PAVAN. This is because since it uses information about the density and the contents of the C2P2s within the STL. Since this utility model penalizes policies that over-predict, we see that SS_{only} performs the worst followed by SSL_t. The performance of PL_r, which uses C2P2 density information and the predictive mobility model, lies between the two extremes.

In our experiments, we observed that the optimum value of STL for SSL_t, PL_t, and PL_r is a function of the utility model, the map, and the initial distribution of the C2P2s across the map. A higher value of STL does not improve the utility metric. This is because another important parameter, m (see Algorithm 3.2 and its discussion), impacts the predictions. A future research direction is to quantify the interaction between m and STL.

Lesson 2: The availability latency of alternative PAVAN policies is significantly different and impacted by both the display time of a title and trip duration. We study the impact of different title durations expressed as a fraction of the total trip duration. Hence, in our next set of experiments, we relax the requirement that the title display time be equal to the length of the path traveled by the C2P2. For each degree of replication, now there were titles whose display time varies from 1 to 6 cells. We define availability latency as the number of cells after which the client will have a title of a particular degree of replication and a particular display time available to it. Hence the possible values of the availability latency range from 0 to 6. For a title with display time 6, the availability latency is either 0 or 6. An availability latency of 0 means that at least one copy of that title was present in each of the 6 cells along the path of the vehicle. An availability latency of 6 means that at least one cell along the path of the vehicle was missing the title. For a title with display time 5, the availability latency is either 0, 1 or 6. If availability latency is 0, the vehicle encountered that title along each of the first 5 cells along its path. If the availability latency is 1, the vehicle encountered the title along the last 5 cells along its path, but not in its first cell. Again an availability latency of 6 indicated that



5.a) Utility model 1



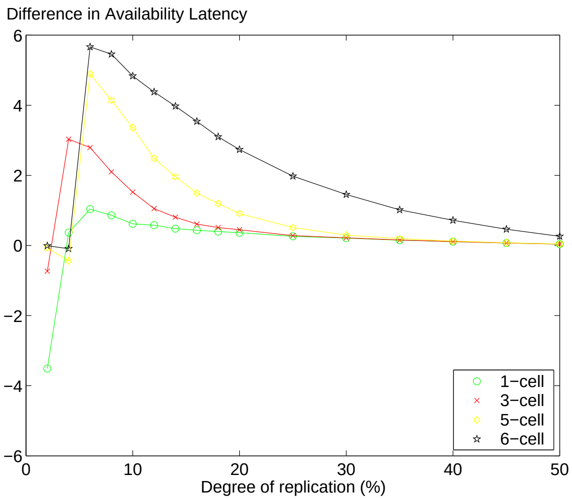
5.b) Utility model 2

Figure 5: A comparison of PAVAN with different inputs for utility models 1 and 2 as a function of the degree of replication of the titles.

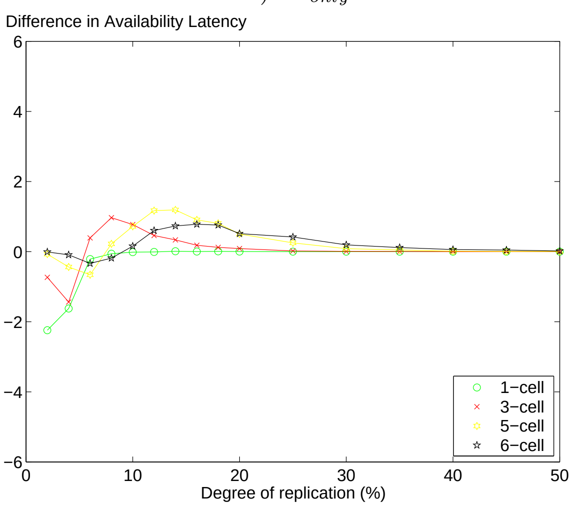
along the path of the client, there was a patch of at least 2 adjacent cells where even one copy of the title was not present. Similarly the availability latency was calculated for each of the other titles of different title display times.

Results based on utility models show SS_{only} and PL_t represent the two extremes. Hence we eliminate results from the other two variants for the remaining sets of experiments noting that their performance was always in between SS_{only} and PL_t. Figure 6.a indicates the average difference between the availability latency of SS_{only} and the actual observed latency as a function of the different degrees of title replication for different title display times. The graph shows the behavior for title display times of 1, 3, 5 and 6 cells. Figure 6.b shows the same for PL_t. Note the lower the difference in the availability latency the better the match between the predicted and the actual lists.

Lesson 3: The accuracy of availability latency estimated by PL_t is the best when compared with other alternatives. The peaks in the availability latency differ-



6.a) SS_{only}



6.b) PL_t

Figure 6: Difference in the availability latencies as a function of the degree of replication of the titles for different title display times.

ence for SS_{only} are an order of magnitude larger than that for PL_t. This is because the over-predictive behavior of SS_{only} causes the predicted availability latency to drop at a much higher rate as compared to actual availability latency. On the other hand, the conservative behavior of PL_t causes the predicted availability latency to rise at a slower rate as compared to the actual availability latency.

With SS_{only}, the predicted list is very accurate when clip display times are greater than 2 and the degree of replication is less than 5%. This is because the average availability latency is close to 6 in both cases. However, as the degree of replication increases beyond 5%, the predicted availability latency drops at a faster rate than what is seen with the actual availability latency. This difference is always positive because SS_{only} always over-predicts irrespective of the clip display time. Note that over-prediction indicates a smaller predicted available latency as compared to the actual availability latency. Beyond 25% degree of replication, availability latency of all titles except those with size 6 converge to 0 (indicating a close match between the predicted and actual availability latency). As the degree of replication increases, slowly the actual available latency catches up with the predicted available latency and the available latency difference converges to 0.

At the other extreme with PL_t, the availability latency difference always lies between -1 and +1.5 irrespective of the clip size. For all clip sizes, beyond 20% replication, the difference in the availability latency converges to 0. For degree of replication less than or equal to 5%, the difference in the availability latency becomes negative. This means that the predicted available latency is higher than the actual observed latency, thereby indicating that for lower degrees of replication PL_t is more conservative. It should be noted that in all cases, the maximum difference in the availability latency does not exceed 1.5. The reason that the difference is not always 0 and the predicted latency is not always equal to the actual one is due to the statistical variations inherent in the experiments. The mobility model is essentially probabilistic, hence if we consider extremely large scenarios, then even for lower degrees of replication, the difference in the availability latency will converge to 0.

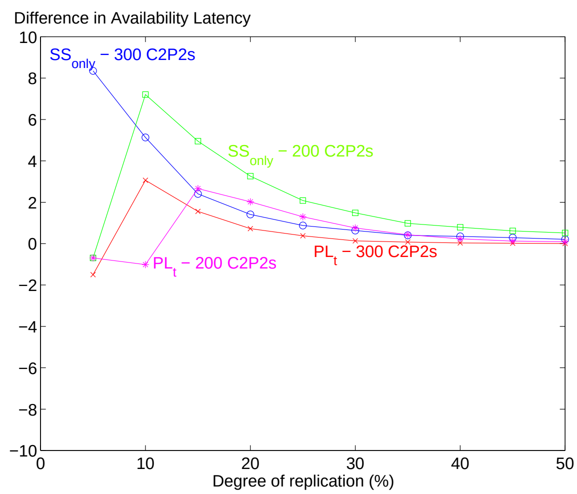


Figure 7: Comparison of difference in the availability latencies of SS_{only} and PL_t for different C2P2 densities in a 10x10 map as a function of the degree of title replication. The title display time under consideration spans 5 cells.

Lesson 4: The accuracy of the availability latency calculations for different variants of PAVAN is sensitive to C2P2 density. The trends seen with respect to the availability latency and the utility models for the different variants of PAVAN essentially remain the same when the 6x6 map used in all our previous experiments is scaled to a 10x10 square grid. Figures 7 illustrates the difference in the availability latencies for different variants of PAVAN for this larger map. We observe that the peaks of the curves for different title display times have a proportionally higher value due to the larger map size.

When the C2P2 density increases, for a given degree of replication, now there are more number of replicas (C2P2s) for each title. Hence similar trends are seen for the difference in availability latency as a function of the degree of replication but the curves for all the PAVAN variants move to the left. Moreover, the curves peak at a lower degree of replication. Figures 7 indicates this behavior of SS_{only} and PL_t in a 10x10 map for C2P2 densities of 200 and 300 when considering a title with a display time of 5 cells.

Even though we have presented the results for one map, we also considered other maps in our simulations. Obtained results show the map is an important parameter that impacts the behavior of PAVAN significantly. Specifically, a lower degree of overlap between the path of a C2P2 equipped vehicle and the gray cells in the map causes the transition probabilities to rapidly diminish toward 0. In such cases, even if the degree of replication of the titles is 100% the variants of PAVAN will always under-predict. This will be the case even if the total number of gray cells increases beyond a certain threshold because then the transition probabilities diffuse quickly. So, within a few steps, the movement prediction probabilities will diminish having very little effect on the predicted lists even in the case of 100% title replication. This effect will also be seen if the map consists of entirely non-gray cells. In such cases, the trends are similar to those seen for the lower replication titles. In conclusion, SS_{only} is appropriate for certain utility models but not all. Also the clip display time is an important parameter that impacts the title availability latency. PL_t demonstrates a competitive performance for all utility models, all clip display times and degrees of replication.

5. CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS

PAVAN is a novel policy that computes when different titles are available in an ad-hoc network of C2P2 devices. It accomplishes this by employing a Markov mobility model that consumes a regional map, a mobility transition table, and a regional look ahead table. This input data is in the order of a few hundred bytes and provided by a base station. Each C2P2 device invokes PAVAN independently. Obtain results demonstrate that one variant of PAVAN, PL_t, provides the most accurate availability latency when compared with other techniques, see Lesson 3 in Section 4. We quantified the quality of lists computed by PAVAN policies using different utility models. Obtained results show these models are sensitive to the degree of replication for a title, see Lesson 1 in Section 4. In general, the lessons of Section 4 hold true for different maps.

We intend to extend PAVAN in several ways. In the short-term, we intend to extend our evaluation to consider directionality of vehicles. This might increase the availability of clips, reducing their availability latency. Second, behavior of PAVAN must be quantified for different distributions of (1) C2P2 equipped vehicles across a geographical area, (2) titles across C2P2 devices and (3) geographical size of a cell. In the long-term, we intend to explore techniques that enable PAVAN to show both a larger number of titles and a shorter availability latency for a title. The idea here is to predict whether C2P2 devices can collaborate to construct additional replicas of a title along the path of a vehicle. With this prediction, mechanisms are needed to adjust the placement of content dynamically. We envision these mechanisms to utilize both the ad-hoc network of C2P2 devices and the participating base stations. In order to better reflect reality we would also like to evaluate the performance of PAVAN using some real world simulation traces of vehicular movements.

6. ACKNOWLEDGMENTS

This research was supported in part by NSF grant IIS-0307908, National Library of Medicine LM07061-01, and an unrestricted cash gift from Microsoft Research.

7. REFERENCES

[1] S. Bararia, S. Ghandeharizadeh, and S. Kapadia. Evaluation of 802.11a for Streaming Data in Ad-hoc Networks. In *Fourth Workshop on Applications and Services in Wireless Networks*, August 2004.

[2] S. Berson, L. Golubchik, and R. Muntz. Fault Tolerant Design of Multimedia Server. In *SIGMOD*, pages 25–36, 1995.

[3] Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, and S. Shenker. Making Gnutella-Like Peer-to-Peer Systems Scalable. In *Proceedings of the ACM SIGCOMM*, pages 407–418, August 2003.

[4] D. DeWitt, S. Ghandeharizadeh, D. Schneider, H. Hsiao, A. Bricker, and R. Rasmussen. The GAMMA Database Machine Project . *IEEE Transactions on Knowledge and Data Engineering*, 2(1), March 1990.

[5] S. Ghandeharizadeh, T. Helmi, S. Kapadia, and B. Krishnamachari. Admission Control and QoS for Continuous Media Displays in Mobile Ad-Hoc Networks of Devices. In *DMS*, 2004.

[6] S. Ghandeharizadeh and B. Krishnamachari. C2P2: A Peer-to-Peer Network for On-Demand Automobile Information Services. In *First International Workshop on Grid and Peer-to-Peer Computing Impacts on Large Scale Heterogeneous Distributed Database Systems (Globe'04)*, 2004.

[7] Z. Haas. A new routing protocol for the reconfigurable wireless networks. In *In Proc. of the IEEE Int. Conf. on Universal Personal Communications*, 1997.

[8] B. Hajek, K. Mitzel, and S. Yang. Paging and Registration in Cellular Networks: Jointly Optimal Policies and an Iterative Algorithm. In *INFOCOM*, 2003.

[9] A. Helmy. Efficient resource discovery in wireless adhoc networks: Contacts do help. *Resource Management in Wireless Networking*, 2004.

[10] D. B. Johnson and D. A. Maltz. Dynamic source routing in ad hoc wireless networks. In Imielinski and Korth, editors, *Mobile Computing*, volume 353. Kluwer Academic Publishers, 1996.

[11] J. Liu, Q. Zhang, B. Li, W. Zhu, and J. Zhang. A unified framework for resource discovery and qos-aware provider selection in ad hoc networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 6(1):13–21, 2002.

[12] D. A. Patterson, G. A. Gibson, and R. H. Katz. A case for redundant arrays of inexpensive disks (raid). In H. Boral and P. Larson, editors, *Proceedings of the 1988 ACM SIGMOD International Conference on Management of Data, Chicago, Illinois, June 1-3, 1988*, pages 109–116. ACM Press, 1988.

[13] P. Sinha, R. Sivakumar, and V. Bharghavan. CEDAR: a core-extraction distributed ad hoc routing algorithm. In *INFOCOM (1)*, pages 202–209, 1999.

[14] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *Proceedings of the 2001 conference on applications, technologies, architectures, and protocols for computer communications*, pages 149–160. ACM Press, 2001.