A Predictive Bandwidth Reservation Scheme Using Mobile Positioning and Road Topology Information

Wee-Seng Soh, *Member, IEEE,* and Hyong S. Kim, *Member, IEEE*

***Abstract*— In cellular networks, an important practical issue is how to limit the handoff dropping probability efficiently. One possible approach is to perform dynamic bandwidth reservation based on mobility predictions. With the rapid advances in mobile positioning technology, and the widespread availability of digital road maps previously designed for navigational devices, we propose a predictive bandwidth reservation scheme built upon these timely opportunities. In contrast to the common practice of utilizing only incoming handoff predictions at each cell to compute the reservations, our scheme is more *efficient* as it innovatively utilizes both incoming and outgoing handoff predictions; it can meet the same target handoff dropping probability by blocking fewer new calls. The individual base stations are responsible for the computations, which are shown to be simple enough to be performed in real-time. We evaluate the scheme via simulation, along with five other schemes for comparison. Simulation results show that those schemes that rely on positioning information are significantly more efficient than those that do not. Our scheme's additional use of the road topology information further improves upon this advantage, bringing the efficiency closer to the bound set by a benchmark scheme that assumes perfect knowledge about future handoffs.**

***Index Terms*— handoff prioritization, call admission control, mobile positioning, mobility prediction.**

I. Introduction

In recent years, there has been a rapid increase in wireless network deployment and mobile device market penetration. With vigorous research that promises higher data rates, future wireless networks will likely become an integral part of the global communication infrastructure. However, there are some unique problems in cellular networks that challenge their service reliability. In addition to the problems introduced by fading, user mobility places stringent requirements on network resources; a call may be dropped during a handoff attempt when the new cell does not have sufficient bandwidth. From a user's point of view, the dropping of an ongoing call is generally more frustrating than the blocking of a new call. Therefore, handoff-requests are usually prioritized over new call requests by reserving some bandwidth at each base station (BS) that could only be utilized by incoming handoffs. Since any such reservation inevitably increases the call blocking probability of new calls (P_{CB}), and decreases the system's utilization, it is extremely important that the reservations are made as sparingly as possible, while meeting the handoff dropping probability (P_{HD}) target that is deemed to be acceptably low by most customers. In other words, the key objective is to meet the customers' expectation at the lowest possible cost.

In early work on the handoff prioritization problem, a static approach is proposed [1], in which a fixed portion of the radio capacity is permanently reserved for incoming handoffs. However, such a static approach cannot handle variable load and mobility [2]. In a non-stationary environment, the reservation required to achieve the same P_{HD} target actually fluctuates with load and mobility. A fixed reservation cannot meet the P_{HD} target all the time unless it is large enough to accommodate the worst-case scenario; this leads to higher P_{CB} than necessary. In contrast, a dynamic approach that adjusts the reservation according to anticipated handoffs may potentially result in a lower P_{CB} for the same P_{HD} target.

The best tradeoff between P_{CB} and P_{HD} can only be achieved if the dynamics of every mobile terminal (MT), such as its path and its arrival/departure times in each cell, are known in advance. However, such an ideal scenario is unlikely. The next best option is to predict their mobility, and perform reservations using these predictions. Many predictive schemes have been proposed in the literature. For example, Liu *et al.* [3] uses pattern matching techniques and a self-adaptive extended Kalman filter for next-cell prediction based on cell sequence observations, signal strength measurements, and cell geometry assumptions. In [4], Levine *et al.* propose the concept of a shadow cluster – a set of BSs to which a MT is likely to attach in the near future. The scheme estimates each MT's probability of being in any cell within the cluster for future time intervals, based on its dynamics and call holding patterns in the form of probability density functions (pdfs). Other examples of predictive reservation schemes can be found in [2], [5]–[9]. In the process of meeting the same P_{HD} target, a more efficient scheme can accomplish the task with a lower P_{CB} than a less efficient one. Since the efficiency of a predictive reservation scheme has a direct impact on the operators' revenues, there are strong incentives to develop more efficient schemes.

In the US, the FCC mandates that cellular-service providers must be able to pinpoint a wireless emergency call's location to within 125 m. This spurs research in mobile-tracking techniques. One promising approach is the integration of a global positioning system (GPS) receiver in each MT. According to [10], assisted GPS positioning methods could yield an accuracy of 20 m during 67% of the time. During 2003-2009, a new batch of GPS satellites will be launched to include two additional civilian carrier frequencies that could potentially yield an accuracy of 1 m [11]. The European Space Agency also plans to launch their own system (GALILEO), which targets an accuracy of 1 m (95% of the time within 10 m) [12]. As more breakthroughs in positioning techniques take place, fueled by the strong interest in location-based services from the industry, future MTs are likely equipped with reasonably

accurate location-tracking capability. The time is thus ripe for active research into how such inherent capability may be harnessed for QoS provisioning in cellular networks. Specifically, we are interested in designing a predictive bandwidth reservation scheme that utilizes real-time positioning information. This could give rise to better prediction accuracy and greater adaptability to time-varying conditions than previous methods, which is crucial for more timely and efficient reservations.

While there are previous attempts to perform positioning-based predictive reservation [3], [6], [8], none of them has addressed the fact that the cell boundary is fuzzy and irregularly shaped due to terrain characteristics and obstacles that interfere with radio propagation. Instead, either hexagonal or circular boundaries are assumed. Also, no previous work has utilized road topology information for predictive reservation until we first proposed the idea in [13], [14]. Since MTs that are carried in vehicles are the ones with high mobility, the use of road topology information would likely improve the performance of predictive reservation schemes. Another important observation from existing work is that only *incoming* handoff predictions at each BS are used to adjust its reservation. We argue that more efficient tradeoffs between P_{CB} and P_{HD} may be achieved if both *incoming* and *outgoing* handoff predictions are used. For example, when a possible incoming handoff is predicted, but there are also outgoing handoffs that are predicted to release sufficient bandwidth in time, no additional reservation would be attempted for the incoming handoff. In contrast, reservation schemes that only utilize incoming handoff predictions would still attempt to reserve additional bandwidth in such a scenario, which could result in unnecessary blocking of new calls.

In [9], we propose a reservation scheme that utilizes mobility predictions based on mobile positioning information. It is the first scheme that considers irregular cell boundaries. The scheme uses linear extrapolation from a MT's recent positions to predict its handoff cell and time, whereby the cell boundary is approximated as a series of points around the BS, computed using previously recorded handoff locations. In this paper, we propose a novel predictive reservation scheme that utilizes road topology knowledge, in addition to the MT's positioning information. It could potentially achieve more accurate predictions at the cost of increased complexity, but the resulting improvement in reservation efficiency may justify this cost. The proposed scheme consists of two components. The first component, the *mobility prediction* module, defines the prediction tasks to be performed by the individual BSs. The second component, the *dynamic bandwidth reservation* module, defines the way both incoming and outgoing handoff predictions are used to adjust the reservation at each BS.

An important point to emphasize here is that we do not use the MT's positioning information to decide whether a handoff should be initiated, because it usually depends on the received signal strength measurements, error rates, interference, as well as handoff protocols used [15]. Instead, we only use the positioning information for predicting the MT's future handoff time and target cell, so as to adjust the reservations.

The remainder of this paper is organized as follows. In Section II, we present our mobility prediction module that utilizes both mobile positioning and road topology information. In Section III, we describe our dynamic bandwidth reservation module, which innovatively uses both incoming and outgoing handoff predictions at each BS to boost the reservation efficiency. Section IV describes the simulations that compare the proposed scheme's performance with several other schemes. Finally, we give our conclusions in Section V.

II. Mobility Prediction Module

The mobility prediction module requires each MT in an active call to report its position to the serving BS every ΔT (say, 1 sec). This consumes a small amount of wireless bandwidth (several bytes per update), which might be negligible for future broadband services. The overhead can also be reduced by suspending the updates when the MT is within a threshold distance from the BS, where a handoff is unlikely to occur anytime soon. For packet services, the header overheads due to such updates may be reduced by piggybacking the position information with other data packets whenever possible.

We now give an overview of the mobility prediction module. The predictions are performed periodically by the BSs, which are expected to have sufficient computational and storage resources. Each BS maintains a database that stores the required information. During a prediction, the BS identifies a set of possible paths from every active MT's current position that may lead to a handoff within a threshold time $T_{\text{threshold}}$, and generates a 4-tuple for each such path, in the form of [target cell, prediction weight, lower prediction limit, upper prediction limit]. The *target cell* is the predicted new cell along that path. The *prediction weight* is a real number within $[0, 1]$ that indicates how likely the prediction is correct. The *lower prediction limit* (LPL) is a lower statistical bound for the remaining time from handoff (t_{remain}), with given probability ζ_{L}, i.e., $P[t_{\text{remain}} \geq \text{LPL}] = \zeta_{\text{L}}$. The *upper prediction limit* (UPL) is an upper statistical bound for t_{remain} with given probability ζ_{U}, i.e., $P[t_{\text{remain}} \leq \text{UPL}] = \zeta_{\text{U}}$. Note that ζ_{L} and ζ_{U} are input parameters that dictate the values of LPL and UPL.

We first describe below the prediction database that is kept at each BS, and its corresponding maintenance procedure. The prediction algorithm is then described in Section II-B.

A. Prediction Database

Each BS maintains a unique database that stores the essential information required for making predictions, including the road topology within its radio coverage area. We refer to the road between two neighboring junctions as a *road segment*, and identify each segment using a junction pair (j_1, j_2), where a junction is an intersection of roads (e.g., T-junction). The approximate coordinates of each junction are stored. Since a road segment may contain bends, it can be broken down further into piecewise-linear line segments, whose end coordinates are also recorded. All these coordinates could be extracted from existing digital road maps previously designed for GPS-based navigational devices. We do not expect frequent updates to these maps because new roads are not constructed very often, while existing road layouts are seldom modified.

The database also stores some important information about each road segment. Since two-way roads would likely have

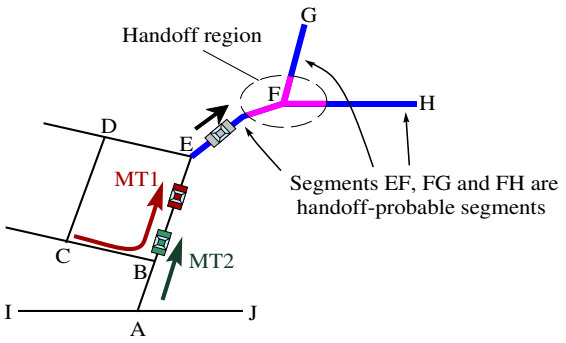

Fig. 1. Utilizing road topology information for mobility prediction.

different characteristics for each direction, the database stores information corresponding to opposite directions separately. The following summarizes what the database stores:

- Identity of neighboring segments at each junction.
- Transition probabilities between neighboring segments (computed from the paths taken by previous MTs).
- Statistical data of time taken to transit each segment.
- Statistical data regarding possible handoffs along each segment, such as probability of handoff, time in segment before handoff, and handoff positions.

With the exception of the first item listed above, the other database entries need to be updated regularly to adapt to the current traffic conditions. As for its memory requirement at a typical BS, it is expected to range only between several hundred kilobytes to several megabytes, because the roads are represented efficiently as piecewise-linear line segments.

In reality, the transition probabilities between neighboring road segments would likely vary with time and traffic. For stochastic processes with statistics that vary slowly with time, it is often appropriate to treat the problem as a succession of stationary problems. We model the transition between neighboring road segments as a second-order Markov process, and assume stationarity between the database updates to simplify the computations. Based on this model, the conditional distribution of a MT choosing a neighboring segment, given all its past segments, is dependent only on the current segment and the immediate prior segment. Using Fig. 1 as an illustration, consider two MTs (MT1 and MT2) that are currently traveling from junction B towards junction E. MT1 came from segment CB, while MT2 came from segment AB. The conditional probability of MT1 going to segment EF will be different from that of MT2, i.e., $P[s_{k+1}{=}\text{EF}|s_k{=}\text{BE}, s_{k-1}{=}\text{CB}]$ and $P[s_{k+1}{=}\text{EF}|s_k{=}\text{BE}, s_{k-1}{=}\text{AB}]$, respectively, where s_k is the current segment that the MT transits. Note that our stationarity assumption implies that the above conditional probabilities are independent of the value of k.

When a call just begins, the MT's prior segment is unknown because it was not tracked previously. Thus, we also need the first-order conditional distribution at each segment, derived from a subset of the data used for deriving the second-order conditional distribution. For example, if the prior segments of MT1 and MT2 in Fig. 1 are unknown, their conditional probabilities of going to segment EF are both $P[s_{k+1}{=}\text{EF}|s_k{=}\text{BE}]$.

A road segment is regarded as a "handoff-probable segment" (HPS) if there is a non-zero probability that a MT transiting the segment would make a handoff-request. From the handoff-requests that are previously observed, the target handoff cell is recorded, along with the statistical information regarding the time spent in the HPS prior to making the requests, and the positions where the requests are made.

Using the model described above, we could determine via the chain rule the conditional probabilities of reaching and handing off at each of the HPSs from segments that are several hops away. We could also predict the remaining time before handoff for each possible path, using the statistical information associated with each segment along the path.

A database update occurs every T_{DB}. After each update, the BS starts collecting the data required for the next update. Note that it is also reasonable to use historical data from the same time-of-the-day and day-of-the-week for the update, in addition to the data collected recently over the last T_{DB}. This could compensate for the lack of data samples occasionally.

In the following, we explain how the database is maintained. Table I shows the notations used, while Fig. 2 shows the update procedure. We first empty both $\mathcal{S}_{\text{HPS}}$ and $\mathcal{S}_{\text{RSV}}$ (Lines 1 and 2), as we are regenerating them using the new data. From Lines 3 to 13, we sequentially examine every road segment within the BS's coverage area, one at a time. Lines 4 and 5 evaluate the first and second order transition probabilities from the segment to its neighboring segments, based on the paths of MTs previously served by the BS. The transition probabilities for U-turns are excluded as they are rare, but can be included if desired. Line 6 estimates the pdf of the time spent by previous MTs in the segment, based on histograms with appropriate bin size. Line 7 computes the probability that a MT would request a handoff while transiting the segment. If handoffs have occurred along this segment previously, the segment is identified as a HPS, and entered into both $\mathcal{S}_{\text{HPS}}$ and $\mathcal{S}_{\text{RSV}}$ (Lines 9 and 10). Its membership in $\mathcal{S}_{\text{RSV}}$ signifies that MTs transiting this segment are potential candidates for bandwidth reservation. Lines 11 to 13 simply evaluate the database entries that describe the handoff behavior of MTs in this segment.

From Lines 14 to 24, we make a second pass through each road segment, again sequentially. For each segment s_{ab}, we reset $\mathcal{R}_{X,\text{HPS}}(s_{ab})$ so that it will be regenerated using newly computed database entries (Line 15). For each hop-limited route ($\leq X$ hops) that originates from segment s_{ab}, we test whether its last segment is a HPS (Lines 16 and 17). Note that a "route" must include the origin segment s_{ab}, and at least one other segment. A hop limit is specified to reduce unnecessary computations, since a MT that is still many segments away from a HPS is unlikely to request a handoff anytime soon. Also, note that $\mathcal{R}_X(s_{ab})$ is pretty much static, and is modified only when there are changes to the road topology within the BS's coverage; it does not need to be recomputed every database update. If the examined route's last segment is a HPS, we estimate the pdf $m_{\text{HO},ab|\varphi}(t)$ of the time taken to transit φ' and part of the last segment $s_{\text{last}}(\varphi)$ before handoff (Line 18). It is obtained from the convolution of the pdfs $f_{\text{transit}}(t)$ of segments in the partial route φ', and also the pdf $g_{\text{HO}}(t)$ of the last segment $s_{\text{last}}(\varphi)$ of route φ. For example, if the segment

TABLE I

NOTATIONS USED FOR EXPLAINING DATABASE MAINTENANCE.

Notation	Meaning
$T_{\text{threshold}}$	Threshold time dictating whether a 4-tuple needs to be generated for a route.
$T_{\text{thres\_max}}$	Maximum $T_{\text{threshold}}$ allowed.
$\mathcal{S}$	Set of road segments within BS's coverage area.
s_{ab}	Directional segment from junction j_a to j_b.
$\mathcal{N}(j_a)$	Set of neighboring junctions of junction j_a.
$\mathcal{N}_{\text{cells}}$	Set of neighboring cells next to the cell of interest.
$\mathcal{S}_{\text{HPS}}$	Set of handoff-probable segments (HPSs) in $\mathcal{S}$.
$\mathcal{S}_{\text{RSV}}$	Set of segments in which only MTs here are examined for the need to make reservations.
$P[s_{k+1}\|s_k]$	1st order conditional transition probability, i.e., P[transit to s_{k+1}|currently s_k].
$P[s_{k+1}\|s_k,s_{k-1}]$	2nd order conditional transition probability, i.e., P[transit to s_{k+1}|currently s_k, previously s_{k-1}].
$C_{\text{HO}}(s_{ab})$	Most probable target handoff cell if handoff occurs along s_{ab}, where $C_{\text{HO}}(s_{ab}) \in \mathcal{N}_{\text{cells}}$.
$P_{\text{HO}}[s_{ab}]$	P[handoff along s_{ab}|MT is currently on s_{ab}].
$f_{\text{transit},ab}(t)$	pdf of time taken to transit s_{ab}.
$g_{\text{HO},ab}(t)$	pdf of time spent in s_{ab} before handoff in s_{ab}.
$h_{\text{HO},ab}(d)$	pdf of distance from j_b where handoff occurs, when MT is on s_{ab}.
X	Hop limit of routes that are considered.
$\mathcal{R}_X(s_{ab})$	Set of all possible routes within X hops originating from s_{ab}. A route $\varphi \in \mathcal{R}_X(s_{ab})$ is a sequence of segments, starting with s_{ab}: $\{s_{ab}s_{bc}\ldots s_{yz}\}$.
$s_{\text{initial}}(\varphi)$	Initial segment of route φ.
$s_{\text{last}}(\varphi)$	Last segment of route φ.
φ'	Route φ without its initial and last segments, i.e., $\{\varphi\} = \{s_{\text{initial}}(\varphi)\} \cup \{\varphi'\} \cup \{s_{\text{last}}(\varphi)\}$.
$m_{\text{HO},ab\|\varphi}(t)$	pdf of time taken to transit φ' and part of last segment $s_{\text{last}}(\varphi)$ before handoff in $s_{\text{last}}(\varphi)$.
$M^{-1}_{\text{HO},ab\|\varphi}(q)$	qth quantile of time taken to transit φ' and part of last segment $s_{\text{last}}(\varphi)$ before handoff in $s_{\text{last}}(\varphi)$.
$\mathcal{R}_{X,\text{HPS}}(s_{ab})$	A subset of routes from $\mathcal{R}_X(s_{ab})$, each of which terminates with a HPS, and, excluding the remaining time in current segment s_{ab}, has a median time to handoff that is within $T_{\text{thres\_max}}$.
$P_{\text{HO}}[\varphi\|s_k]$	1st order conditional probability that a MT in s_k follows $\varphi \in \mathcal{R}_{X,\text{HPS}}(s_k)$ and hands off at $s_{\text{last}}(\varphi)$.
$P_{\text{HO}}[\varphi\|s_k,s_{k-1}]$	2nd order conditional probability that a MT in s_k follows $\varphi \in \mathcal{R}_{X,\text{HPS}}(s_k)$ and hands off at $s_{\text{last}}(\varphi)$.

we are currently processing is s_{ab}, and we consider one of its routes, $\varphi = \{s_{ab}, s_{bc}, s_{cd}, s_{de}\}$. This route has three hops, with partial route $\varphi' = \{s_{bc}, s_{cd}\}$. Its last segment $s_{\text{last}}(\varphi)$ is s_{de}, which is assumed to be a HPS here. Then,

$$m_{\text{HO},ab|\varphi}(t) = f_{\text{transit},bc}(t) \otimes f_{\text{transit},cd}(t) \otimes g_{\text{HO},de}(t). \quad (1)$$

Note that $m_{\text{HO},ab|\varphi}(t)$ does not include the time taken to complete the current segment s_{ab}, which will be estimated using a MT's actual speed during a prediction. Once the pdf $m_{\text{HO},ab|\varphi}(t)$ is obtained, the median time $M^{-1}_{\text{HO},ab|\varphi}(0.5)$ can be easily calculated. If it is within the limit $T_{\text{thres\_max}}$, we add the route φ to the set $\mathcal{R}_{X,\text{HPS}}(s_{ab})$, and include the segment s_{ab} in $\mathcal{S}_{\text{RSV}}$ (Lines 20 and 21). We then compute via the chain rule the conditional probabilities that MTs currently in segment s_{ab} would follow this route and hand off at its last segment (Lines 22 and 23). Finally, the quantiles $M^{-1}_{\text{HO},ab|\varphi}(1-\zeta_{\text{L}})$ and $M^{-1}_{\text{HO},ab|\varphi}(\zeta_{\text{U}})$ are computed for this route (Line 24), which are needed for computing the prediction limits LPL and UPL.

```
1  S_HPS ← ∅
2  S_RSV ← ∅
3  for each s_ab ∈ S
4      evaluate P[s_{k+1}=s_bx | s_k=s_ab]
                 ∀j_x ∈ N(j_b) − {j_a}
5      evaluate P[s_{k+1}=s_bx | s_k=s_ab, s_{k−1}=s_ya]
                 ∀j_x ∈ N(j_b) − {j_a}, ∀j_y ∈ N(j_a) − {j_b}
6      evaluate f_{transit,ab}(t)
7      evaluate P_HO[s_ab]
8      if P_HO[s_ab] > 0
9          then S_HPS ← S_HPS ∪ {s_ab}
10             S_RSV ← S_RSV ∪ {s_ab}
11             evaluate C_HO(s_ab)
12             evaluate g_{HO,ab}(t)
13             evaluate h_{HO,ab}(d)
14 for each s_ab ∈ S
15     R_{X,HPS}(s_ab) ← ∅
16     for each φ ∈ R_X(s_ab)
17         if s_last(φ) ∈ S_HPS
18             then evaluate m_{HO,ab|φ}(t) and M^{-1}_{HO,ab|φ}(0.5)
19                  if M^{-1}_{HO,ab|φ}(0.5) ≤ T_thres_max
20                      then R_{X,HPS}(s_ab) ← R_{X,HPS}(s_ab) ∪ {φ}
21                           S_RSV ← S_RSV ∪ {s_ab}
22                           evaluate P_HO[φ|s_k=s_ab]
23                           evaluate P_HO[φ|s_k=s_ab, s_{k−1}=s_ya]
                                      ∀j_y ∈ N(j_a) − {j_b}
24                           evaluate M^{-1}_{HO,ab|φ}(1−ζ_L), M^{-1}_{HO,ab|φ}(ζ_U)
```

Fig. 2. Prediction database update procedure.

In the following, we give examples of how the database entries $P_{\text{HO}}[\varphi|s_k, s_{k-1}]$ and $P_{\text{HO}}[\varphi|s_k]$ can be computed. Reusing the topology in Fig. 1, suppose we are computing the above entries for segment s_k=BE, and we are interested in the route φ = {BE, EF, FH}. These entries represent the conditional probabilities that a MT currently traveling along segment BE would go through segment EF and enter segment FH, and finally make a handoff-request while in segment FH. Although segment EF is also a HPS, the route φ = {BE, EF, FH} assumes that the MT does not hand off in segment EF. A route that assumes handoff in EF will be regarded as a different route, that is, {BE, EF}. Therefore, the conditional probabilities for φ must account for the probability that the MT does not make a handoff-request while traveling along segment EF. For the first-order conditional probability, which does not assume knowledge about the prior segment, it is obtained as

$$\begin{aligned} &P_{\text{HO}}[\varphi{=}\{\text{BE,EF,FH}\}|s_k{=}\text{BE}] \\ &= P[s_{k+1}{=}\text{FH}|s_k{=}\text{EF}, s_{k-1}{=}\text{BE}] \\ &\quad \cdot P[s_{k+1}{=}\text{EF}|s_k{=}\text{BE}]\cdot(1-P_{\text{HO}}[\text{EF}])\cdot P_{\text{HO}}[\text{FH}]. \end{aligned} \quad (2)$$

Since segment BE has two prior segments (CB and AB) where MTs may come from, there are two second-order conditional probabilities. For example, the one with prior segment CB is

$$\begin{aligned} &P_{\text{HO}}[\varphi{=}\{\text{BE,EF,FH}\}|s_k{=}\text{BE}, s_{k-1}{=}\text{CB}] \\ &= P[s_{k+1}{=}\text{FH}|s_k{=}\text{EF}, s_{k-1}{=}\text{BE}] \\ &\quad \cdot P[s_{k+1}{=}\text{EF}|s_k{=}\text{BE}, s_{k-1}{=}\text{CB}]\cdot(1-P_{\text{HO}}[\text{EF}])\cdot P_{\text{HO}}[\text{FH}]. \end{aligned} \quad (3)$$

An important point to emphasize here is that the database update procedure is not computationally intensive. Each BS maintains its own database, and the update only occurs once every T_{DB} (say, 1 hr). The computational requirement scales linearly with the number of road segments under the BS's coverage area ($|\mathcal{S}|$), but scales exponentially with the hop limit X. For instance, suppose most junctions are cross-junctions, the number of routes to be considered scales approximately with the factor $3^X|S|$. One way to reduce this complexity is to impose an additional distance threshold, $D_{\text{thres\_max}}$, on the set of routes that can be included in $\mathcal{R}_X(s_{ab})$, so as to exclude unreasonably long routes. The threshold $D_{\text{thres\_max}}$ may then be chosen as the farthest distance that any MT could accomplish within $T_{\text{thres\_max}}$. Note, however, that it is generally sufficient to have a small X (say, 2 or 3). In our simulations based on a 3 GHz Intel Pentium IV CPU, the typical database update time for $X = 2$ is in the order of several seconds, which is very small compared to the database update interval T_{DB}.

B. Mobility Prediction Algorithm

Having seen the prediction database update procedure, we now describe the prediction algorithm. In order to perform the predictions, the BS needs to map each MT's current position onto the correct road segment within the road topology database (a process known as map-matching [16], [17]). Here, we do not describe how the map-matching is performed, as it depends on the accuracy of the positioning techniques used. Instead, we assume for simplicity that the MT's current road segment and estimated speed are already computed based on its recent positions. Interested readers can refer to the relevant literature from Intelligent Transportation Systems (ITS) research for additional information, such as [16], [17].

Recall that during the database update, a small number of road segments are placed into the set S_{RSV}. The prediction algorithm only examines those MTs that are currently traveling in these segments, because they have the greatest potential of making handoff-requests within $T_{\text{threshold}}$. In the following, we present the prediction algorithm performed for a *single* MT i that is traveling in segment $s^i_{ab} \in \mathcal{S}_{\text{RSV}}$. The key idea is to identify a set of possible paths from MT i's current position that may result in a handoff within $T_{\text{threshold}}$, and generate a 4-tuple for each such path. Note that only those paths that belong to the pre-computed set $\mathcal{R}_{X,\text{HPS}}(s^i_{ab})$ need to be examined.

Table II shows the additional notations used, while Fig. 3 presents the prediction algorithm for MT i. In Line 1, we empty the prediction output set $\mathcal{Z}^i$, as new predictions will be made. Line 2 checks that the MT is not stationary, otherwise we exit the algorithm. Next, in Line 3, we estimate the MT's remaining distance from the end of its current segment. The time for the MT to reach this end is then estimated (Line 4). From Lines 5 to 12, we examine those pre-computed candidate routes that might lead to handoffs. For each route, we estimate its LPL(UPL) as the sum of two components, namely, the estimated time taken to finish the current segment, and the LPL(UPL) of the time taken to transit the remaining route and hand off at the very last segment. Note that the quantiles $M^{-1}_{\text{HO},ab|\varphi}(1-\zeta_{\text{L}})$ and $M^{-1}_{\text{HO},ab|\varphi}(\zeta_{\text{U}})$ have already been pre-computed during the database update. If the LPL, $\hat{t}^i_{\text{L}}(\varphi,\zeta_{\text{L}})$, falls within the threshold time of the most probable target handoff cell associated with this route, a 4-tuple prediction is generated. The prediction weight is either the first or second order conditional probability of route φ, depending on whether we know the previous segment of MT i (Lines 8 to 11). Then, in Line 12, we insert the 4-tuple prediction into the set $\mathcal{Z}^i$.

TABLE II
ADDITIONAL NOTATIONS USED TO PRESENT PREDICTION ALGORITHM.

Notation	Meaning
v^i	Estimated speed of MT i.
s^i_{ab}	Current road segment in which MT i is traveling.
s^i_{prev}	Previous segment of MT i (may or may not be known).
$d^i_{\text{EOS}}(s^i_{ab})$	MT i's estimated distance from end-of-segment, j_b.
$t^i_{\text{EOS}}(s^i_{ab})$	MT i's estimated time from end-of-segment, j_b.
$T_{\text{thres}}(C_j)$	$T_{\text{threshold}}$ of neighboring cell C_j (dynamically adjusted).
$\hat{c}^i_{\text{target}}(\varphi)$	MT i's most probable target handoff cell if it follows route φ and hands off at $s_{\text{last}}(\varphi)$, i.e., $\hat{c}^i_{\text{target}}(\varphi) = C_{\text{HO}}(s_{\text{last}}(\varphi))$.
$w^i(\varphi)$	Prediction weight specifying the probability that MT i may follow route φ and hands off at $s_{\text{last}}(\varphi)$.
$\hat{t}^i_{\text{L}}(\varphi,\zeta_{\text{L}})$	LPL of MT i's remaining time from handoff (t^i_{remain}) if it follows route φ and hands off at $s_{\text{last}}(\varphi)$, such that $P[t^i_{\text{remain}} \geq \hat{t}^i_{\text{L}}(\varphi,\zeta_{\text{L}})] = \zeta_{\text{L}}$.
$\hat{t}^i_{\text{U}}(\varphi,\zeta_{\text{U}})$	UPL of MT i's remaining time from handoff (t^i_{remain}) if it follows route φ and hands off at $s_{\text{last}}(\varphi)$, such that $P[t^i_{\text{remain}} \leq \hat{t}^i_{\text{U}}(\varphi,\zeta_{\text{U}})] = \zeta_{\text{U}}$.
$\hat{t}^i_{\text{L}}(s^i_{ab},\zeta_{\text{L}})$	LPL of t^i_{remain} if MT i hands off in s^i_{ab}.
$\hat{t}^i_{\text{U}}(s^i_{ab},\zeta_{\text{U}})$	UPL of t^i_{remain} if MT i hands off in s^i_{ab}.
$\mathcal{Z}^i$	Set of 4-tuple predictions for MT i. Each 4-tuple has the following form: [target cell, prediction weight, LPL, UPL]. For a prediction that MT i may follow route φ and hands off at $s_{\text{last}}(\varphi)$, the corresponding 4-tuple is: $[\hat{c}^i_{\text{target}}(\varphi), w^i(\varphi), \hat{t}^i_{\text{L}}(\varphi,\zeta_{\text{L}}), \hat{t}^i_{\text{U}}(\varphi,\zeta_{\text{U}})]$. If s^i_{ab} is a HPS, then the 4-tuple for a prediction that a handoff may occur along s^i_{ab} itself is: $[C_{\text{HO}}(s^i_{ab}), P_{\text{HO}}[s^i_{ab}], \hat{t}^i_{\text{L}}(s^i_{ab},\zeta_{\text{L}}), \hat{t}^i_{\text{U}}(s^i_{ab},\zeta_{\text{U}})]$.

If the MT is currently within a HPS, Lines 13 to 17 determine whether a 4-tuple prediction needs to be generated. Here, we introduce two additional quantiles for MT i, needed for calculating the LPL and UPL of the predicted time from handoff. As they are dependent on the MT's position within the segment, they must be recomputed during each prediction. Let D_{ab} be a random variable representing the MT's distance from junction j_b where a handoff occurs. Note that its pdf, $h_{\text{HO},ab}(d)$, has been pre-computed during the database update. Suppose the MT is currently at a distance D_t from junction j_b, and has not made a handoff-request. Knowing this, we can derive a conditional pdf, $h_{\text{HO},ab}(d|D_{ab} < D_t)$, for $d < D_t$:

$$h_{\text{HO},ab}(d|D_{ab} < D_t) = \frac{h_{\text{HO},ab}(d)}{P[D_{ab} < D_t]}, \tag{4}$$

where $P[D_{ab} < D_t]$ can be obtained by integrating $h_{\text{HO},ab}(d)$ from 0 to D_t. Note that $h_{\text{HO},ab}(d|D_{ab} < D_t) = 0$ for $d \geq D_t$. From the above conditional pdf shown in (4), its conditional cumulative distribution function (cdf) can be obtained as:

$$H_{\text{HO},ab}(d|D_{ab} < D_t) = \int_0^d h_{\text{HO},ab}(u|D_{ab} < D_t)\ du. \tag{5}$$

1 $\mathcal{Z}^i \leftarrow \emptyset$
2 **if** MT i is not stationary
3 **then** compute $d^i_{\text{EOS}}(s^i_{ab})$
4 $t^i_{\text{EOS}}(s^i_{ab}) \leftarrow d^i_{\text{EOS}}(s^i_{ab})/v^i$
5 **for** each $\varphi \in \mathcal{R}_{\text{X,HPS}}(s^i_{ab})$
6 $\hat{t}^i_{\text{L}}(\varphi, \zeta_{\text{L}}) \leftarrow t^i_{\text{EOS}}(s^i_{ab}) + M^{-1}_{\text{HO},ab|\varphi}(1-\zeta_{\text{L}})$
7 $\hat{t}^i_{\text{U}}(\varphi, \zeta_{\text{U}}) \leftarrow t^i_{\text{EOS}}(s^i_{ab}) + M^{-1}_{\text{HO},ab|\varphi}(\zeta_{\text{U}})$
8 **if** $\hat{t}^i_{\text{L}}(\varphi, \zeta_{\text{L}}) \le T_{\text{thres}}(\hat{c}^i_{\text{target}}(\varphi))$
9 **then if** s^i_{prev} is known
10 **then** $w^i(\varphi) \leftarrow P_{\text{HO}}[\varphi | s_k = s^i_{ab}, s_{k-1} = s^i_{\text{prev}}]$
11 **else** $w^i(\varphi) \leftarrow P_{\text{HO}}[\varphi | s_k = s^i_{ab}]$
12 $\mathcal{Z}^i \leftarrow \mathcal{Z}^i \cup \{[\hat{c}^i_{\text{target}}(\varphi), w^i(\varphi), \hat{t}^i_{\text{L}}(\varphi, \zeta_{\text{L}}), \hat{t}^i_{\text{U}}(\varphi, \zeta_{\text{U}})]\}$
13 **if** $s^i_{ab} \in \mathcal{S}_{\text{HPS}}$
14 **then** $\hat{t}^i_{\text{L}}(s^i_{ab}, \zeta_{\text{L}}) \leftarrow [d^i_{\text{EOS}}(s^i_{ab}) - H^{-1}_{\text{HO},ab}(\zeta_{\text{L}} | D < d^i_{\text{EOS}}(s^i_{ab}))]/v^i$
15 $\hat{t}^i_{\text{U}}(s^i_{ab}, \zeta_{\text{U}}) \leftarrow [d^i_{\text{EOS}}(s^i_{ab}) - H^{-1}_{\text{HO},ab}(1-\zeta_{\text{U}} | D < d^i_{\text{EOS}}(s^i_{ab}))]/v^i$
16 **if** $\hat{t}^i_{\text{L}}(s^i_{ab}, \zeta_{\text{L}}) \le T_{\text{thres}}(C_{\text{HO}}(s^i_{ab}))$
17 **then** $\mathcal{Z}^i \leftarrow \mathcal{Z}^i \cup \{[C_{\text{HO}}(s^i_{ab}), P_{\text{HO}}[s^i_{ab}], \hat{t}^i_{\text{L}}(s^i_{ab}, \zeta_{\text{L}}), \hat{t}^i_{\text{U}}(s^i_{ab}, \zeta_{\text{U}})]\}$

Fig. 3. Prediction algorithm for MT i traveling in segment s^i_{ab}.

With the above conditional cdf, it is straightforward to approximate any q^{th} conditional quantile $H^{-1}_{\text{HO},ab}(q|D_{ab} < D_t)$. By estimating the time that the MT would take to reach two specific quantile points, namely $H^{-1}_{\text{HO},ab}(\zeta_{\text{L}}|D_{ab} < D_t)$ and $H^{-1}_{\text{HO},ab}(1-\zeta_{\text{U}}|D_{ab} < D_t)$, we are able to specify the LPL and UPL for a possible handoff that might occur along s_{ab}.

In Lines 14 and 15, we obtain the LPL and UPL as the estimated time taken to reach the two quantile points, $H^{-1}_{\text{HO},ab}(\zeta_{\text{L}}|D < d^i_{\text{EOS}}(s^i_{ab}))$ and $H^{-1}_{\text{HO},ab}(1-\zeta_{\text{U}}|D < d^i_{\text{EOS}}(s^i_{ab}))$. If the LPL, $\hat{t}^i_{\text{L}}(s^i_{ab}, \zeta_{\text{L}})$, falls within the threshold time of the cell $C_{\text{HO}}(s^i_{ab})$, we insert the corresponding 4-tuple into the prediction set $\mathcal{Z}^i$ (Lines 16 and 17).

As mentioned earlier, the above algorithm only performs predictions for a single MT i. We need to repeat the algorithm for all active MTs that are currently traveling in segments that belong to the set S_{RSV}. Although a typical BS may handle several hundred active calls at anytime, only several tens of MTs are likely to be traveling in these segments that require predictions. Also, as can be seen from the algorithm, most of the calculations are simple, and they utilize mainly pre-computed information stored in the database. The slightly more time-consuming operations are the computations of the quantiles $H^{-1}_{\text{HO},ab}(\zeta_{\text{L}}|D < d^i_{\text{EOS}}(s^i_{ab}))$ and $H^{-1}_{\text{HO},ab}(1-\zeta_{\text{U}}|D < d^i_{\text{EOS}}(s^i_{ab}))$. However, by estimating all pdfs using histograms with appropriate bin sizes, and by paying special attention to the computational efficiency, each of the above quantiles typically only requires the computational intensity of several hundred additions. In our simulations, each BS completes all its predictions within several tens of milliseconds.

III. Dynamic Bandwidth Reservation Module

In this section, we describe our dynamic bandwidth reservation module, and explain how the 4-tuple predictions are used. Unlike most existing schemes that only utilize incoming handoff predictions to adjust their reservations, we utilize both incoming and outgoing handoff predictions to achieve more efficient tradeoffs between P_{HD} and P_{CB}. In the following, we first describe the system model assumed. We then explain the logic behind our approach, before presenting the algorithms.

A. System Model

Although it is suggested in [5] and [18] that some adaptive applications might accept a lower bandwidth at the expense of lower call quality during congestion, we do not consider them here, as they may make it harder to visualize the advantages of using positioning and road topology information for mobility predictions. Similar to [2], we also preclude delay-insensitive applications that can tolerate long handoff delays, as well as, soft handoffs in CDMA systems, in which a MT can simultaneously connect with two or more BSs. All these preclusions may be added as future extensions. In our model, we only consider applications that require fixed bandwidth guarantees. We assume that the minimum bandwidth granularity that may be allocated to any call is 1 *bandwidth unit* (BU) [2], [4]. For example, a voice call may require 1 BU, while a constant-bit-rate video call may require several BUs [2], [4].

We follow the common assumption of existing reservation schemes that each BS j has a fixed capacity of $C(j)$ BUs [2]. Note that additional research may be performed to extend our scheme to systems with time-varying capacity (including CDMA systems in which the capacity depends on the target interference [19]). Given the bandwidth demand of individual calls, the BS performs admission control to ensure that the total demand of all active calls does not exceed $C(j)$.

In order to prioritize handoffs over new calls, each cell must reserve some bandwidth that can only be used by incoming handoffs. Specifically, each BS j has a "reservation target" $R_{\text{target}}(j)$ that is being updated regularly based on mobility predictions. A new call request is accepted if the remaining bandwidth after its acceptance is at least $R_{\text{target}}(j)$, i.e.,

$$C(j) - b_{\text{used}}(j) - b_{\text{new}} \ge R_{\text{target}}(j), \tag{6}$$

where b_{new} is the bandwidth required by the new call, and $b_{\text{used}}(j)$ is the bandwidth allocated to existing calls in cell j. Note that $R_{\text{target}}(j)$ is merely a target, not the actual bandwidth available to the incoming handoffs. The BS can only attempt to meet this target by rejecting new call requests, while waiting for existing calls to release their bandwidth when they end, or hand off to other cells. For a handoff request, the admission control is more lenient; it is admitted so long as there is sufficient remaining capacity, regardless of $R_{\text{target}}(j)$:

$$C(j) - b_{\text{used}}(j) \ge b_{\text{handoff}}, \tag{7}$$

where b_{handoff} is the bandwidth needed by the handoff call.

We assume that all rejected new call requests are cleared, and subsequent requests are independent of the previous requests. When a BS cannot accommodate an incoming handoff, we assume that it is dropped. We do not consider handoff queuing here, although it would likely improve our scheme's performance (and that of other schemes simulated for comparison), so as to focus on the advantages of using positioning and road topology information for mobility predictions.

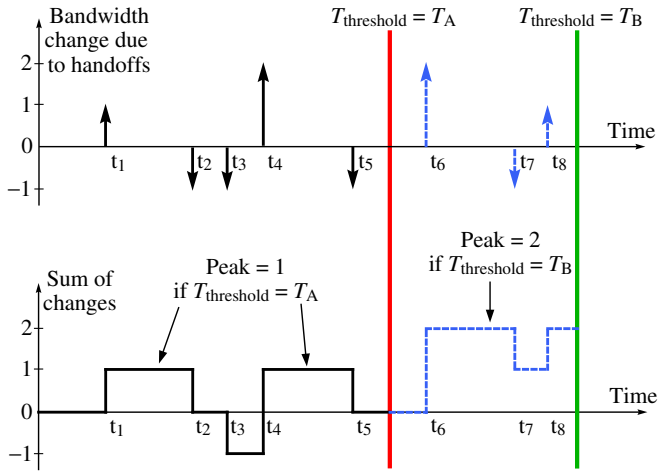

Fig. 4. If we have perfect knowledge about handoffs up to time $T_{\text{threshold}}$.

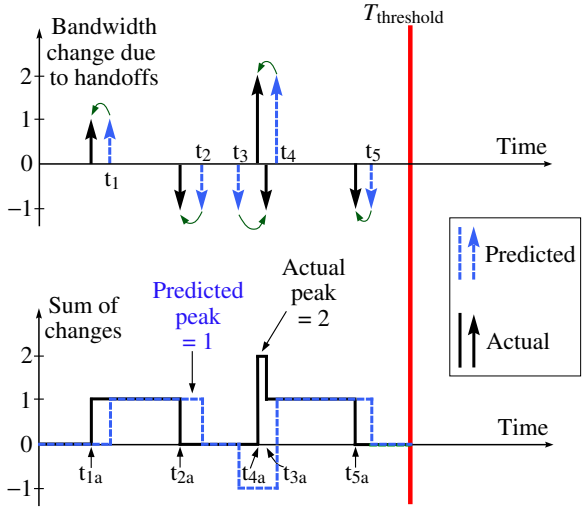

Fig. 5. Effects of prediction errors in handoff timings.

B. Logic Behind Our Approach

In order to better understand the logic leading to the proposed scheme, we first ask ourselves the following question:

> *Suppose we have perfect knowledge about all the incoming/outgoing handoffs that will occur within a limited time into the future, how much bandwidth should be reserved to prevent any of these incoming handoffs from being dropped?*

Fig. 4 shows an example that helps answer this question. Here, we assume perfect knowledge about future handoffs up to time $T_{\text{threshold}}$. The top figure shows the changes in bandwidth demand due to handoffs, and the time they occur. An incoming handoff leads to a positive change, while an outgoing handoff leads to a negative change. The bottom figure shows the sum of bandwidth changes over time due to these handoffs. Suppose $T_{\text{threshold}} = T_{\text{A}}$, meaning that the BS is only interested in preventing handoff dropping up to time T_{A}. By summing up all the bandwidth changes over $[0, T_{\text{A}}]$, we realize that the peak additional bandwidth requirement within $[0, T_{\text{A}}]$ is 1 BU. This implies that *if we succeed* in reserving 1 BU, we can ensure that *all* incoming handoffs within $[0, T_{\text{A}}]$ will not be dropped. Thus, an appropriate $R_{\text{target}}(j)$ at time $t = 0$ is 1 BU. In contrast, for a reservation scheme that does not utilize outgoing handoff information (e.g., [2]), only the positive bandwidth changes are considered. However, if we were to set $R_{\text{target}}(j)$ to 3 BUs, we are actually over-reserving bandwidth, such that new calls may be blocked unnecessarily within $[0, T_{\text{A}}]$.

As mentioned earlier, $R_{\text{target}}(j)$ is merely a target. If there are insufficient existing calls that release bandwidth, $R_{\text{target}}(j)$ cannot be met. This causes some incoming handoffs to be dropped, despite having prior knowledge about them. However, this becomes less likely if the BS has more time to meet the target. The time $T_{\text{threshold}}$ can be viewed as the time given to the BS to set aside the required bandwidth to avoid dropping a handoff. Referring to Fig. 4 again, notice that the handoffs beyond time T_{A} are shown as dotted lines. This information is currently not considered by the BS, hence it will set $R_{\text{target}}(j)$ to 1 BU. Suppose the BS has 2 BUs of spare capacity at time $t = 0$. If a new call from MT x needs 1 BU, the BS will accept the call because it still satisfies $R_{\text{target}}(j)$ after accepting the call. However, if no existing call ends before time t_6, then the spare bandwidth at time t_6 remains at 1 BU, thus causing the incoming handoff to be dropped. In contrast, if we have set $T_{\text{threshold}}$ to T_{B}, then $R_{\text{target}}(j)$ would have been 2 BUs. The BS would then have rejected the new call request from MT x so as to maintain its spare capacity at 2 BUs. Consequently, the incoming handoff at time t_6 will not be dropped. This shows that it is possible to reduce P_{HD} by giving the BS earlier notice, which could be done by increasing $T_{\text{threshold}}$. Thus, we could vary $T_{\text{threshold}}$ to adjust P_{HD}. Note that $R_{\text{target}}(j)$ increases monotonically with $T_{\text{threshold}}$.

The scenario above is for the ideal case of having perfect knowledge about handoffs, which is unlikely in real-life. We now examine a more realistic scenario, whereby we only have handoff predictions. Fig. 5 illustrates the possible effects of prediction errors in handoff timings. Here, handoffs are predicted at t_1, t_2, t_3, t_4 and t_5, but the actual handoffs occur at $t_{1\text{a}}, t_{2\text{a}}, t_{3\text{a}}, t_{4\text{a}}$ and $t_{5\text{a}}$. Based on the predictions, the peak sum is 1 BU. However, the actual peak is 2 BUs. This increases the likelihood that the incoming handoff at time $t_{4\text{a}}$ may be dropped. A closer look reveals that the error in predicted peak arises because the predicted sequence of a pair of incoming and outgoing handoffs is wrong. The outgoing handoff is predicted to occur (at t_3) before the incoming handoff (at t_4), but the incoming handoff actually occurs (at $t_{4\text{a}}$) before the outgoing handoff (at $t_{3\text{a}}$). This reversal of the predicted and actual sequences causes the actual peak to become larger than the predicted peak for the example shown[1]. An interesting point to note is that, if, on the other hand, an incoming handoff is predicted to occur before an outgoing handoff, but the actual sequence is reversed, then the actual peak might be lower than the predicted peak. However, this type of prediction error is benign because it does not increase the chance of handoff dropping; it may only result in over-reservation of bandwidth.

[1]A reversal in the predicted and actual sequences does not always result in a difference between the predicted and actual peaks. This is because the peak observed within the time interval $[0, T_{\text{threshold}}]$ may have been caused by some other incoming handoff.

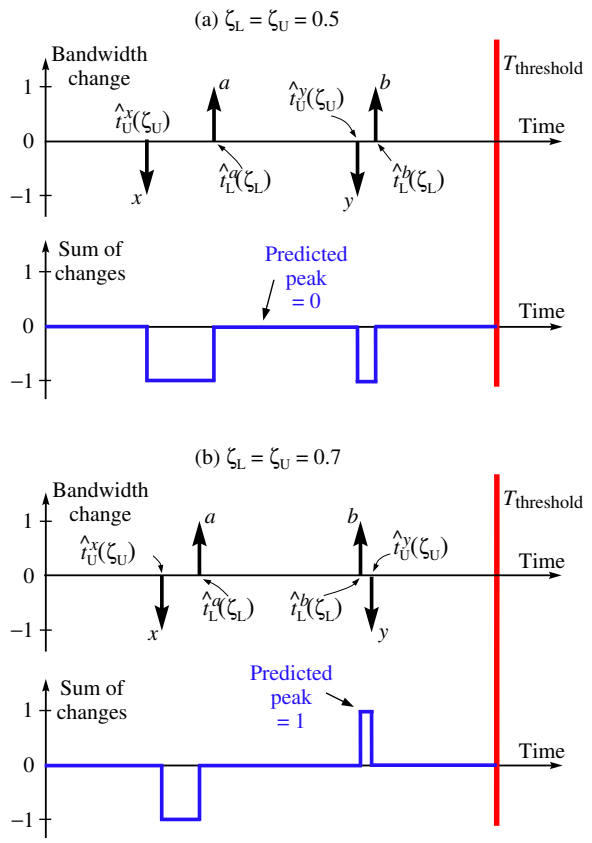

Fig. 6. Effects of varying the values of ζ_L and ζ_U.

From the above, we observe that it is undesirable when an incoming handoff occurs earlier than its predicted time, and also when an outgoing handoff occurs later than its predicted time. Thus, we would like to reduce the likelihood of these scenarios. Recall that each 4-tuple prediction consists of a LPL and a UPL. Suppose we always use the LPLs and UPLs as the predicted times for incoming and outgoing handoffs, respectively. By specifying ζ_L and ζ_U to be larger than 0.5, we can introduce some biases into the predicted times, and reduce the likelihood of the above scenarios. Let us consider a simple example as follows. Suppose BS i predicts that MTs x and y would hand off to other cells with UPLs $\hat{t}^x_U(\zeta_U)$ and $\hat{t}^y_U(\zeta_U)$, respectively. Also, its neighboring BS j predicts that MTs a and b would hand off into BS i with LPLs $\hat{t}^a_L(\zeta_L)$ and $\hat{t}^b_L(\zeta_L)$, respectively. We assume that there are no other predicted handoffs that are leaving or entering BS i within $[0, T_{threshold}]$. When BS i receives the predictions from BS j, it determines the bandwidth that it needs to reserve. Figure 6(a) shows the timing diagram and peak computation for $\zeta_L = \zeta_U = 0.5$. Here, each MT's actual handoff time is equally likely to be either earlier or later than its predicted handoff time. Nothing has been done to alleviate the likelihood of the aforementioned undesirable scenarios. A closer look at Figure 6(a) reveals that MT y's predicted departure is very close to the predicted arrival of MT b. This pair of predictions carries a relatively high risk that MT b may actually arrive before MT y departs, which increases the chance of MT b being dropped. Suppose we wish to bias the predicted times by specifying a larger value for ζ_L and ζ_U. Figure 6(b) shows a possible outcome if ζ_L and ζ_U were set to 0.7. With the biases, MT b is now predicted to arrive before MT y departs. A predicted peak of 1 BU is obtained, and we have now reduced the risk of underestimating the peak requirement. Although the biases also moved the predicted times of MTs x and a closer to each other, their predicted sequence remains the same. Therefore, if the injected biases are small, the predicted arrival and departure sequence for those handoffs that are sufficiently far apart would probably remain the same, as though no biases have been injected. However, these biases would have the benefit of capturing and correcting those predictions that are close enough to result in under-reservation at the slightest prediction error.

Note that ζ_L and ζ_U are design parameters whose optimal values are best determined through experimentation in real cellular networks. A general rule of thumb is to set a value that is within the range of $0.5 \sim 0.7$. Any value that is under 0.5 will actually increase the likelihood of under-reservation, while a value that is too high may render the predictions too conservative and result in excessive over-reservation.

Having seen these key concepts, we describe below how each BS dynamically adjusts its $T_{threshold}$ to meet the desired P_{HD}. Section III-D then explains how a BS adjusts its R_{target}.

TABLE III
NOTATIONS USED IN ALGORITHM FOR ADJUSTING $T_{THRESHOLD}$.

Notation	Meaning
T_{thres_max}	The maximum $T_{threshold}$ value allowed.
T_{thres_min}	The minimum $T_{threshold}$ value allowed.
T_{thres_init}	The initial $T_{threshold}$ value.
n_{HO}	The number of handoffs counted.
n_{HD}	The number of handoff droppings counted.
$P_{HD,target}$	The desired P_{HD} target.
w_{obs}	Observation window size.
μ	Scaling factor (an experimentally determined parameter).

C. Adjusting $T_{threshold}$ at each BS

In Section III-B, we have seen that the P_{HD} experienced by incoming handoffs may be indirectly controlled by adjusting $T_{threshold}$. However, the value of $T_{threshold}$ for the same desired P_{HD} would probably be different in each cell, as it likely depends on the cell's coverage area, subscriber density, and so on. It might even fluctuate with user mobility and traffic load. Since there is no obvious way to compute $T_{threshold}$, we utilize an adaptive algorithm to adjust its value for any given P_{HD}. Table III shows the notations we have used in our algorithm, while the actual algorithm is shown in Fig. 7.

The adaptive algorithm attempts to maintain approximately one handoff dropping out of every w_{obs} incoming requests. If there is no handoff dropping within w_{obs} handoffs, $T_{threshold}$ will be decreased by 1 sec if it is larger than the minimum value. A fresh observation window will be restarted when the current window is exhausted. If more than one handoff dropping is observed within the window, $T_{threshold}$ is immediately increased by 1 sec if it has not yet reached the maximum value. When this happens, the observation window is also restarted.

For a desired P_{HD} target, the value of w_{obs} is chosen to be $\lceil \mu / P_{HD,target} \rceil$, where μ is a scaling factor close to 1. Ideally, if

```
1  w_obs = ⌈μ/P_HD,target⌉;
2  T_threshold ← T_thres_init; n_HO ← 0; n_HD ← 0;
3  while (system running)
4    if (incoming handoff-request occurs)
5      then n_HO ← n_HO + 1;
6           if (handoff accepted)
7             then if (n_HO ≥ w_obs)
8                    then if ((n_HD = 0) & (T_threshold > T_thres_min))
9                           then T_threshold ← T_threshold − 1;
10                   n_HO ← 0; n_HD ← 0;
11            else n_HD ← n_HD + 1;
12                 if (n_HD > 1)
13                   then if (T_threshold < T_thres_max)
14                          then T_threshold ← T_threshold + 1;
15                   n_HO ← 0; n_HD ← 0;
```

Fig. 7. Algorithm used by each BS to adjust its $T_{\text{threshold}}$.

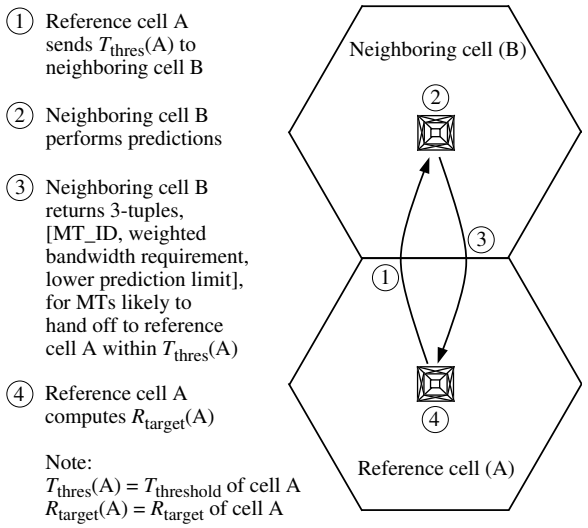

Fig. 8. Procedure performed every T_{predict} for computing R_{target}.

the algorithm were to succeed in achieving exactly one handoff dropping every w_{obs} handoffs, then w_{obs} should be simply $1/P_{\text{HD,target}}$. However, through our simulations, we discover that the P_{HD} obtained this way slightly deviates from the desired target $P_{\text{HD,target}}$ by an approximately constant factor (about $1.2{\sim}1.25$). A possible explanation for this observation is that handoffs are bursty and the best that our adaptive algorithm could achieve is to allow the value of $T_{\text{threshold}}$ to fluctuate around its optimal value. This causes the average number of handoff droppings per w_{obs} observations to deviate slightly from 1. To compensate for the above difference, the scaling factor μ is introduced for the calculation of w_{obs}. Note that μ shall be determined experimentally in an actual system.

D. Adjusting R_{target} at each BS

The predictions used to compute R_{target} are made periodically every T_{predict}, which is a design parameter. If the predictions are very frequent, they are more accurate but the computational requirement at each BS also increases. On the other hand, they become less accurate when they are far apart, and the tradeoff between P_{HD} and P_{CB} becomes less efficient.

Fig. 8 shows the procedure that is repeated every T_{predict}. For clarity, we only show two cells; cell A is our reference cell for which we demonstrate the computation of its R_{target}, while cell B is one of A's neighboring cells. Note that steps 1–3 are performed simultaneously for every neighbor of cell A. Also, cell A concurrently serves as a neighboring cell for cell B; the procedure also applies when they interchange their roles.

An assumption made here is that inter-BS signaling is possible via the radio access network connecting the BSs. Also, such signaling messages are given high priority so that they can be delivered with the smallest possible delay. The following describes each step of the procedure:

Step 1: Reference cell A transmits its $T_{\text{threshold}}$ to neighboring cell B, which uses this to decide what prediction information needs to be sent to A.

Step 2: Neighboring cell B generates 4-tuple predictions for its outgoing handoffs. Note that cell A itself will also be performing predictions at the same time for its role as some other cells' neighbor (not shown).

Step 3: For every 4-tuple prediction that picks cell A as the target cell, and whose LPL is within cell A's $T_{\text{threshold}}$, the neighboring cell B transmits part of the prediction to cell A as a 3-tuple, with the format [MT_ID, weighted bandwidth requirement, predicted time]. The *MT_ID* is a unique identifier for each MT, the *weighted bandwidth requirement* is the product of the prediction weight and the MT's bandwidth requirement, while the *predicted time* is the LPL.

Step 4: As cell A receives the 3-tuples from cell B, they are inserted into a sorted list according to their predicted times in ascending order. These are the incoming handoff predictions. Cell A then examines its own 4-tuple predictions for outgoing handoffs. For those with UPLs within cell A's $T_{\text{threshold}}$, they are also inserted into the list, but in the form of 3-tuples with format [MT_ID, $-$weighted bandwidth release, predicted time]. The *weighted bandwidth release* is the product of the prediction weight and the bandwidth released when the MT leaves. The *predicted time* is its UPL. Upon completing the sorted list, the bandwidth change from every entry is summed in order. The overall peak discovered is then assigned to R_{target}.

Although the predictions are performed every T_{predict}, a BS may adjust its R_{target} between successive predictions when it acquires updated information. Specifically, R_{target} is adjusted when any of the following events occurs:

1) A previously predicted incoming handoff within the list has taken place.
2) A previously predicted outgoing handoff within the list has either handed off or ended its call.
3) A previously predicted incoming handoff within the list has either ended its call without handoff, or has handed off to a different cell. The BS needs to be informed by the neighboring BS that has previously sent the 3-tuple.

Note that only event 3 requires inter-BS signaling; events 1 and 2 occur locally, thus the BS already has the information. When any updated information is acquired, the BS removes the affected entry from its sorted list, and recomputes R_{target}.

It may be noticed that the adjusting of R_{target} has not considered the *likelihood* that some predicted incoming handoffs

may end their calls before arriving. Although we do recompute R_{target} *after* encountering any such event (see event 3 above), the algorithm's efficiency could potentially improve if we were to incorporate the above likelihood into R_{target} even before they occur. Suppose the distributions of the call durations are known, and let $P_{\text{dur},i}(t > \tau_0 + \tau_1 | t > \tau_0)$ be the probability that MT i would last at least another τ_1 sec, knowing that it has already lasted τ_0 sec. The above likelihood can then be incorporated, by multiplying the weighted bandwidth requirements of the predicted incoming handoffs by their respective probabilities $P_{\text{dur},i}$, when calculating R_{target}. Note, however, that the above procedure may be quite tedious to implement, unless the call durations are exponentially distributed.

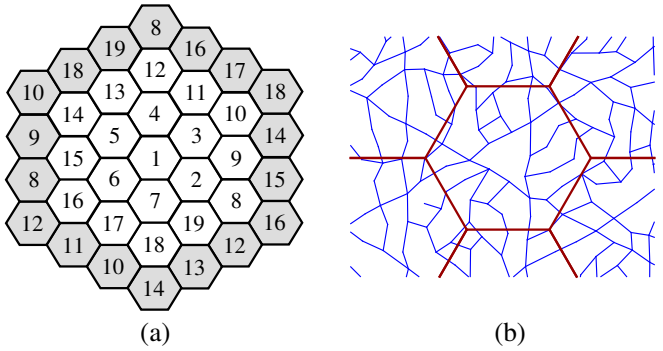

Fig. 9. (a) Simulation network with wrap-around at the network boundary, (b) a sample road layout generated using heuristic rules.

IV. Simulations and Results

A. Simulation Model

To facilitate the evaluation, a novel simulation model was designed. In contrast to commonly adopted models in which the MTs are assumed to travel in straight lines for random distances before undergoing random direction changes, our model incorporates road layouts that place constraints on MTs' paths. The simulation network consists of 19 wireless cells. In order to eliminate the boundary effects that make it difficult to comprehend the results, the cells at the boundary are made to wrap around [2], [8], as shown in Fig. 9(a). Whenever a MT travels out of the network boundary, it is re-injected into the network via the appropriate wrap-around cell, as though a handoff has occurred from outside the network. The road topology in each cell is randomly generated using heuristic rules; real maps are not used because the roads also need to wrap around at the network boundary. The road layouts are designed to resemble those found in city areas. Fig. 9(b) shows an example of the road topology that was randomly generated.

Unlike many existing simulation models in which the handoffs occur at either the circular or hexagonal cell boundaries, our handoff positions are randomly distributed. Suppose R is the designed *cell radius* (assumed to be 1000 m), typically defined as the distance from the BS to the vertex of the hexagonal cell. When a MT is between $1.1R$ and $1.2R$ from the BS, we assume that a handoff will occur during its transit through this region. The time at which the handoff occurs is a random variable that is uniformly distributed over the total time spent in the region. The target BS is assumed to be the nearest neighboring BS at the time when the handoff occurs (although this may not be the case in real life). We do not claim that the model resembles the handoff position distribution in a real cellular network; its main purpose is to create an *irregular* handoff region with some *uncertainty*, so as to evaluate the performance of different dynamic reservation schemes.

To make the problem more interesting, we introduce traffic lights at the road junctions. The lights are assumed to change every 30 sec. Each road segment is assigned a speed limit, chosen from the set $\{40, 50, 60\}$ km/h with equal probability. The MT's speed is Gaussian-distributed, with its mean being the speed limit of that particular road segment. The standard deviation is assumed to be 5 km/h, and the speed is truncated to a limit of three standard deviations from its mean. Once again, the above is merely a simulation assumption that does not necessarily resemble a real MT's speed distribution.

We do not assume any particular positioning technology, as new breakthroughs will continue to emerge. For this reason, we chose to ignore the effects of positioning errors in all but one set of simulations, because their distribution and correlation are dependent on the positioning technology. The actual map-matching techniques for mapping the raw positioning data onto the digital road maps are also beyond the scope of this paper, as they are usually designed based on the distribution and correlation of the positioning errors. Note, however, that more efficient map-matching techniques can be utilized when more accurate positioning technologies become available. For instance, when the positioning errors are much smaller than the separation between neighboring road segments, a MT's current road segment can be easily obtained as the nearest segment from its reported position.

Each cell is assumed to have a fixed capacity C of 100 BUs. For simplicity, all MTs' bandwidth requirements are assumed to be symmetric, although the scheme may be modified to handle asymmetric requirements. Our traffic model is similar to the one used in [2]. New calls are generated according to Poisson distribution with rate λ (calls/sec/cell) in each cell. The initial position of a new call and its destination can be on any road segment with equal probability. The MT then follows the shortest path between its origin and destination. Similar to [2], a call request is either of type "voice" (requires 1 BU) or of type "video" (requires 4 BUs), with probabilities R_{vo} and $1 - R_{\text{vo}}$, respectively. In the simulations, R_{vo} is set to 0.5. All MTs are assumed to have the same P_{HD} requirement, regardless of their call types. The holding times for both types of calls are assumed to be exponentially distributed, with mean 180 sec. We define the *normalized offered load* per cell as

$$L_{\text{norm}} = \frac{[1 \cdot R_{\text{vo}} + 4 \cdot (1 - R_{\text{vo}})] \cdot \lambda \cdot 180}{C}. \tag{8}$$

In this paper, we mainly present the results for $L_{\text{norm}} = 1$. The interval between predictions, T_{predict}, is 5 sec. We also assumed that all MTs have positioning capability, although a real network would probably include some MTs that do not (which might degrade performance). Finally, the probabilities ζ_{L} and ζ_{U} that affect the prediction limits are both 0.65, as they are found to perform best for the simulation model used.

B. Other Schemes Simulated For Comparison

In the remaining paper, we refer to our scheme as the *road topology based* scheme (RTB). We have also simulated five other bandwidth reservation schemes for comparison purposes:

1) Benchmark scheme: This idealized scheme assumes perfect knowledge about every MT's *next* cell and handoff time. It uses the same algorithms in Sections III-C and III-D for adjusting $T_{\text{threshold}}$ and R_{target}. However, it uses the actual handoff times to compute R_{target}, instead of prediction limits.

2) Reactive scheme: This scheme is purely reactive, and gives a bound for the worst efficiency of predictive schemes. The basic idea is to adapt the BS's R_{target} according to the number of handoffs dropped over w_{obs} handoff-requests. We utilize the same adaptive algorithm presented in Fig. 7 that was originally designed for adjusting $T_{\text{threshold}}$. Instead of adjusting $T_{\text{threshold}}$ (which does not exist here), the algorithm is used for adjusting R_{target} directly. If no handoff is dropped among w_{obs} handoff-requests, R_{target} is decremented by 1 BU. If more than one handoff is dropped, R_{target} is incremented by 1 BU.

3) Choi's AC1 scheme: This is one of the three schemes (AC1, AC2, and AC3) proposed in [2]. In their simulations using 1-D cell layout, AC3 performed best. However, when simulated using our model with 2-D cell layout, AC1 has the best performance, while AC2 and AC3 are over-conservative with much worse efficiency than the Reactive scheme (lower bound). Hence, we only present the results for AC1 here. The scheme works by estimating the probability that a MT would hand off into a neighboring cell within an estimation time window T_{est}, based upon its previous cell, and its extant sojourn time. The neighboring cell's R_{target} is then increased by the MT's bandwidth requirement, weighted by the estimated probability. The T_{est} of each cell is dynamically adjusted based on the measured handoff dropping ratio among a number of handoffs recently observed, so as to meet the desired P_{HD}.

4) Linear extrapolation (LE) scheme: This scheme is similar to the one we proposed in [9]. Although it also uses mobile positioning information for predictive bandwidth reservation, it does not utilize any road topology information. Instead, it uses linear extrapolation over recently observed positions to predict a MT's next cell and handoff time.

5) RTB with path knowledge (RTB_PK) scheme: In the RTB scheme, there is uncertainty about a MT's next road segment, so a prediction needs to consider all possibilities. In practice, it might be possible to predict a MT's path using an adaptive algorithm [20] that could learn a user's mobility profile. The MTs could also be using the routes computed by an ITS navigation system, from which the path may be extracted. Here, the RTB_PK scheme assumes the extreme case whereby a MT's path is *always* known. Note, however, that even when the MT's path is known, we do not know beforehand the exact time and position that the handoff might occur. This distinguishes the scheme from the Benchmark scheme.

C. Simulation Results

We now present the simulation results. All the results shown are the averages over the 19 cells in the simulation network. Sufficient simulations have been performed such that the 95% confidence interval for any P_{HD} is within ± 0.00015 from its sample mean, and for the case of P_{CB}, it is within ± 0.001.

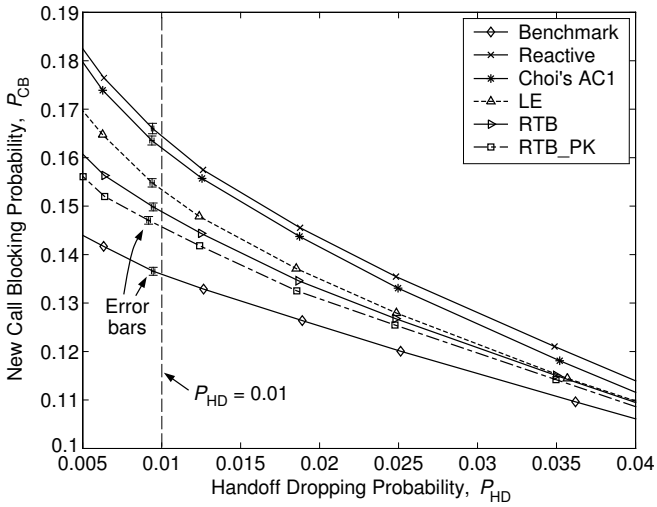

Fig. 10. P_{CB} versus P_{HD} for different schemes at $L_{\text{norm}} = 1.0$.

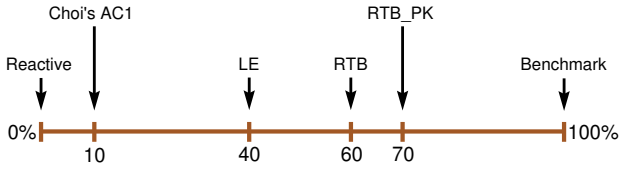

Fig. 11. Approximate normalized efficiencies at $P_{\text{HD}} = 0.01$.

Without handoff prioritization, both P_{CB} and P_{HD} are 0.075. This is unacceptably high for P_{HD}. Fig. 10 shows the plots of P_{CB} versus P_{HD} for the six schemes simulated. For each scheme, the target P_{HD} is varied so as to illustrate its tradeoff with P_{CB}. A curve that is closer to the origin implies that the scheme offers a more efficient tradeoff between P_{CB} and P_{HD}. This is because the scheme is able to achieve the same P_{HD} target by blocking fewer new calls (smaller P_{CB}).

Among the six schemes, the Benchmark scheme is most efficient. It naturally acts as a bound, because it has complete knowledge of when and where the next handoff will occur for every MT. The Reactive scheme, on the other hand, has the worst efficiency. It has no prediction capability, and merely adapts R_{target} according to the number of handoffs being dropped over an observation window of past handoff-requests.

In order to better visualize the relative efficiencies among the schemes, we use the following normalization technique. For any chosen P_{HD} value, we obtain the absolute difference in P_{CB} between each scheme and the Reactive scheme, and then normalize it with respect to the absolute difference in P_{CB} between the Benchmark scheme and the Reactive scheme. The normalized value is then represented as a percentage, referred to as the *normalized efficiency* (NE) of that scheme. Note that the Reactive scheme has an NE of 0%, while the Benchmark scheme has an NE of 100%. For a P_{HD} target of 0.01, the approximate NEs of the various schemes are shown in Fig. 11.

We now examine the performance of the remaining four schemes. As can be seen in Fig. 11, Choi's AC1 scheme performs only slightly better than the Reactive scheme. This is

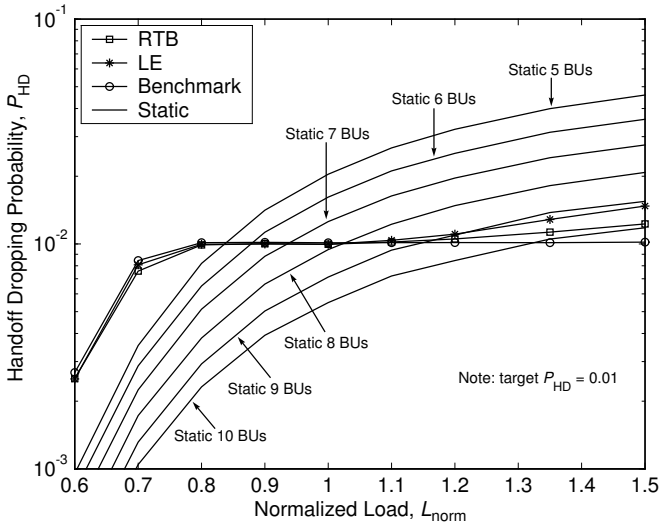

Fig. 12. P_{HD} versus offered load for different schemes.

probably because it is inadequate to make handoff predictions based only on each MT's previous cell and the time already spent in the cell. On the other hand, the LE, RTB, and RTB_PK schemes are much more efficient than Choi's AC1 scheme. This can be attributed to their use of positioning information for mobility predictions, which are more accurate, and also their use of both incoming and outgoing handoff predictions for reservation. It is observed that the LE scheme is able to outperform Choi's AC1 scheme significantly, even though it merely uses a simple linear extrapolation approach for making predictions. For the RTB scheme, the additional road topology knowledge allows it to perform better than the LE scheme. The RTB_PK scheme, which eliminates the uncertainty in predicting the MTs' future paths, further improves upon the RTB scheme, although the improvement is not very dramatic.

We would like to point out that the RTB_PK scheme is not implementable in real-life, because it is unlikely that *all* MTs' paths will be known beforehand. The purpose of simulating this scheme is to examine the *maximum* performance gain over the RTB scheme if it were possible to obtain prior knowledge of MTs' paths. In an actual cellular network, we could have a mixture of MTs with and without known paths. Hence, a hybrid scheme, whose performance is expected to be between that of the RTB and RTB_PK schemes, may be implemented if desired. Since the simulation results of the RTB_PK scheme have shown limited gain over the RTB scheme, there may be little incentive to implement the hybrid scheme.

In the above simulations, the time interval between predictions, T_{predict}, was 5 sec. Since the performance of the predictive schemes would likely deteriorate when T_{predict} becomes larger, we repeated the simulations with $T_{\text{predict}} = 10$ sec to examine its effects. For the same target P_{HD}, the P_{CB}'s of the RTB_PK, RTB, and LE schemes increased only modestly; they increased by no more than 0.003 (or 2% of their original P_{CB}'s), and still outperform Choi's AC1 scheme significantly.

An important characteristic of any dynamic bandwidth reservation scheme is its ability to meet the desired P_{HD} target even when the load varies. This differentiates the dynamic schemes from the static approach, which uses fixed reservation regardless of the load. We now illustrate the above using additional simulations. Fig. 12 shows the P_{HD} values obtained by statically reserving a number of BUs when L_{norm} is varied, as well as those obtained by the RTB, LE, and Benchmark schemes for a target P_{HD} of 0.01. As can be seen, a static reservation that meets the target P_{HD} at a particular L_{norm} can violate this target significantly at higher L_{norm}, while suffering from over-reservation at lower L_{norm} (where more new calls are being rejected than necessary). The dynamic schemes, on the other hand, perform extremely well under varying load. The Benchmark scheme is able to meet the P_{HD} target throughout. For the LE and RTB schemes, even at $L_{\text{norm}} = 1.5$, the P_{HD}'s are approximately 0.0125 and 0.011, respectively. These small violations might be tolerable in most practical applications [2].

Some additional observations can also be made from Fig. 12 as follows. Recall that the Benchmark scheme uses the same algorithms presented in Sections III-C and III-D to adapt its R_{target} and $T_{\text{threshold}}$, except that it has perfect knowledge about each MT's next handoff, whereas the LE and RTB schemes need to rely on mobility predictions. Since the Benchmark scheme can meet the P_{HD} target even at high load regions, we can conclude that our proposed algorithms used for adapting R_{target} and $T_{\text{threshold}}$ work extremely well. This also implies that the small deviations from the target P_{HD} for both LE and RTB schemes probably arise due to the errors in their mobility predictions, which are inevitable in all predictive schemes. The fact that the RTB scheme's P_{HD} is better than the LE scheme's P_{HD} also reinforces the above argument, because the RTB scheme is more accurate than the LE scheme.

In Section III-B, we have explained the importance of utilizing both incoming and outgoing handoff predictions for adjusting the reservation in each cell. Here, we demonstrate via simulations that the reservation efficiencies of such schemes are indeed better than those schemes that only utilize incoming handoff predictions. We consider three additional schemes, which are variants of the Benchmark, LE, and RTB schemes. In these variants, the predictions about outgoing handoffs from each cell are purposely withheld when computing R_{target}. Fig. 13 shows the P_{CB} versus P_{HD} plots for these variants and their original schemes. We also reproduce the plot for Choi's AC1 scheme, which does not utilize outgoing handoff predictions as well. From the plots, we observe that the variants are much less efficient than their original counterparts. This justifies the use of outgoing handoff predictions for reservations. Another important observation is that even without using the outgoing handoff predictions, the variants of both the LE and RTB schemes still outperform Choi's AC1 scheme. This again demonstrates the advantages of using positioning information for predictions, in contrast to the latter which performs predictions based on each MT's previous cell and the time it has already spent in the current cell.

In the simulations, we have used the value 0.65 for the parameters ζ_{L} and ζ_{U} of the RTB scheme. We now demonstrate the effects of varying these two important parameters, and explain why we have chosen the above value. Although it is possible to select different values for these two parameters, we have found through our simulations that they perform very

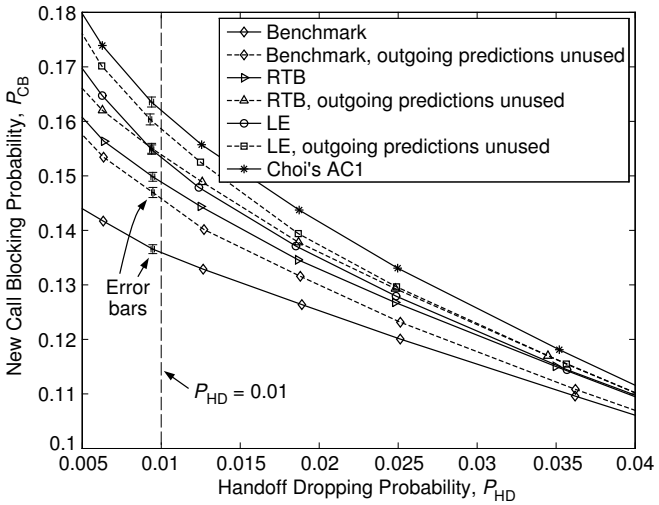

Fig. 13. Plots showing deterioration in performance when outgoing handoff predictions are not used.

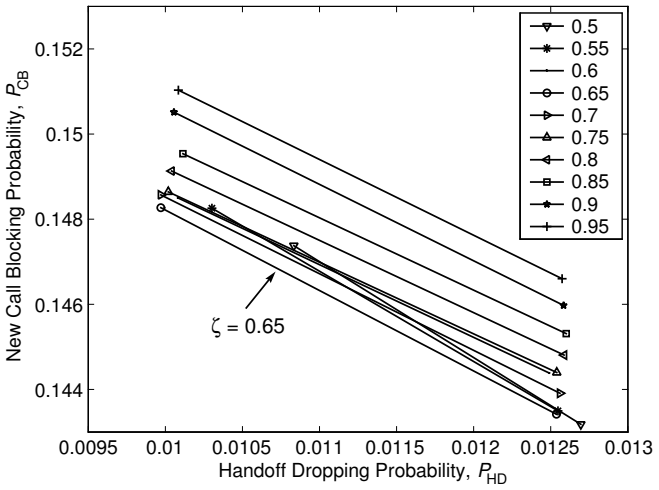

Fig. 14. P_{CB} versus P_{HD} when the values of $\zeta = \zeta_{\mathrm{L}} = \zeta_{\mathrm{U}}$ are varied.

well when their values are identical. For simplicity, we let $\zeta_{\mathrm{L}} = \zeta_{\mathrm{U}}$, and simply refer to them as ζ. Fig. 14 shows a number of P_{CB} versus P_{HD} plots for the RTB scheme, where each plot corresponds to a different value of ζ, ranging from 0.5 to 0.95 with a step size of 0.05. For each value of ζ, we obtained two data points by setting the target P_{HD} values to 0.01 and 0.0125. As can be seen, the efficiency improves as ζ increases from 0.5 to 0.65, and then degrades consistently as ζ increases beyond 0.65. Therefore, we chose $\zeta = 0.65$.

As all the simulations described thus far have not considered positioning errors, we conclude this section by demonstrating their possible effects using a simple error model. An exponentially correlated Gaussian random variable with zero mean, standard deviation σ, and correlation coefficient $e^{-1/\tau}$ is added to the MT's actual position along the road segment every time when it reports its position. Note that it is common for a MT's consecutive position readings to have correlated errors, when using an actual positioning technology such as the GPS. Here, the degree of correlation is determined by τ, whereby $\tau \approx 0$ sec gives uncorrelated errors. The degree of correlation is important for the RTB scheme, because we estimate the MT's speed using several of its most recently reported positions. The results for several different combinations of σ and τ are shown in Fig. 15. When the errors are uncorrelated, the RTB scheme's performance deteriorates considerably as σ increases. For the case where $\sigma = 30$ m with uncorrelated errors, the RTB scheme could no longer meet the target P_{HD} of 0.01, but stops at around 0.012. On the other hand, when the errors are correlated, the RTB scheme's performance does not deteriorate as much – the stronger the correlation (larger τ), the smaller the deterioration. This important observation shows that the RTB scheme is more sensitive towards errors in speed estimation, rather than the actual positioning errors.

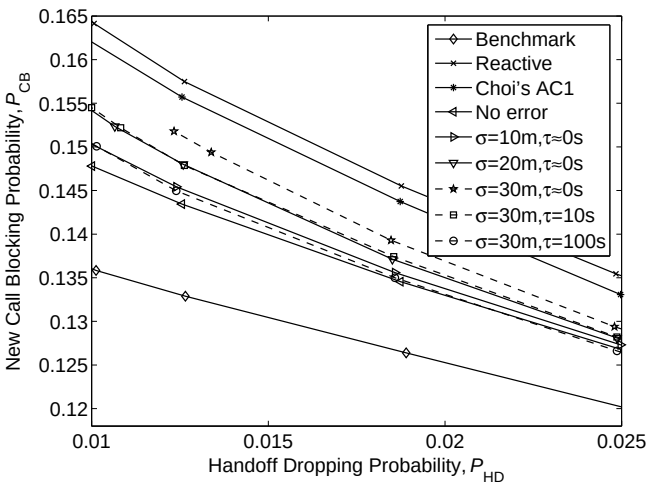

Fig. 15. Plots showing performance deterioration due to positioning errors.

V. Conclusion

We have proposed a novel predictive bandwidth reservation scheme that utilizes mobile positioning and road topology information. It is built upon the assumption that future MTs are likely equipped with reasonably accurate positioning capability, and is inspired by the widespread availability of digital road maps that were previously designed for navigational devices. In contrast to previous predictive methods that use positioning information, our scheme is more practical as it does not assume that the cell boundaries are hexagonal or circular. Also, the use of the road topology information is expected to yield more accurate predictions than before.

Our scheme consists of two modules. The mobility prediction module defines the prediction tasks to be undertaken by the BSs. Each BS periodically generates a number of 4-tuple predictions associated with MTs that are likely to hand off within a threshold time. They are then used by the dynamic bandwidth reservation module for adjusting the reservations. Our scheme is unique as it innovatively uses both incoming and outgoing handoff predictions to achieve more efficient reservations, unlike other schemes that merely uses incoming handoff predictions. It can also be implemented in real-time.

We evaluated our scheme via simulation, and compared it with five other schemes. Our simulation model incorporates road layouts that place constraints on the MTs' paths, which is more realistic than existing models. The relative

performance of the various schemes, with their normalized efficiencies shown in brackets, can be summarized as: Reactive (0%) < Choi's AC1 (10%) < LE (40%) < RTB (60%) < RTB_PK (70%) < Benchmark (100%). The huge jump in efficiency from Choi's AC1 scheme to the LE scheme highlights the advantages of using positioning information for predictions. With the added advantage from using road topology information, the RTB scheme outperforms the LE scheme. The RTB_PK scheme, which assumes prior knowledge of all MTs' paths, shows limited improvement over the RTB scheme. Hence, there is little incentive to implement an RTB-RTB_PK hybrid scheme even if the paths of some MTs may be known.

We have also shown that our scheme only degrades modestly when the prediction time interval increases from 5 sec to 10 sec. In addition, the P_{HD} only deviates slightly from its target when the normalized load is 1.5. In order to justify our claim that *both* incoming and outgoing handoff predictions should be used to achieve better reservation efficiency, we also simulated the variants of the Benchmark, LE, and RTB schemes that do not consider outgoing handoffs. These variants exhibit significant degradation when compared to their original schemes. Nevertheless, both the LE and RTB variants still outperform Choi's AC1 scheme, thus demonstrating the advantages of using positioning information for predictions.

The possible effects of positioning errors have been studied using a simple error model. Although the RTB scheme's efficiency deteriorates with the errors' standard deviation, it is found that a stronger correlation between consecutive positioning errors would lead to a smaller performance degradation. This implies that the RTB scheme is more sensitive to errors in speed estimation, rather than the actual positioning errors.

REFERENCES

[1] D. Hong and S. S. Rappaport, "Traffic model and performance analysis for cellular mobile radio telephone systems with prioritized and nonprioritized handoff procedures," *IEEE Trans. Veh. Technol.*, vol. VT-35, no. 3, pp. 77–92, Aug. 1986.

[2] S. Choi and K. G. Shin, "Adaptive bandwidth reservation and admission control in QoS-sensitive cellular networks," *IEEE Trans. Parallel Distributed Systems (TPDS)*, vol. 13, no. 9, pp. 882–897, Sep. 2002.

[3] T. Liu, P. Bahl, and I. Chlamtac, "Mobility modeling, location tracking, and trajectory prediction in wireless ATM networks," *IEEE J. Select. Areas Commun.*, vol. 16, no. 6, pp. 922–936, Aug. 1998.

[4] D. A. Levine, I. F. Akyildiz, and M. Naghshineh, "A resource estimation and call admission algorithm for wireless multimedia networks using the shadow cluster concept," *IEEE/ACM Trans. Networking*, vol. 5, no. 1, pp. 1–12, Feb. 1997.

[5] C. Oliveira, J. B. Kim, and T. Suda, "Adaptive bandwidth reservation scheme for high-speed multimedia wireless networks," *IEEE J. Select. Areas Commun.*, vol. 16, no. 6, pp. 858–874, Aug. 1998.

[6] M.-H. Chiu and M. A. Bassiouni, "Predictive schemes for handoff prioritization in cellular networks based on mobile positioning," *IEEE J. Select. Areas Commun.*, vol. 18, no. 3, pp. 510–522, Mar. 2000.

[7] W.-S. Soh and H. S. Kim, "Adaptive bandwidth reservation in hierarchical wireless ATM networks using GPS-based prediction," in *Proc. IEEE VTC*, Amsterdam, Netherlands, Sep. 1999, pp. 528–532.

[8] A. Aljadhai and T. Znati, "Predictive mobility support for QoS provisioning in mobile wireless environments," *IEEE J. Select. Areas Commun.*, vol. 19, no. 10, pp. 1915–1930, Oct. 2001.

[9] W.-S. Soh and H. S. Kim, "Dynamic guard bandwidth scheme for wireless broadband networks," in *Proc. IEEE INFOCOM*, Anchorage, Alaska, USA, Apr. 2001, pp. 572–581.

[10] Y. Zhao, "Standardization of mobile phone positioning for 3G systems," *IEEE Commun. Mag.*, pp. 108–116, Jul. 2002.

[11] E. A. Bretz, "X marks the spot, maybe," *IEEE Spectrum*, pp. 26–36, Apr. 2000.

[12] J. Benedicto, S. E. Dinwiddy, G. Gatti, R. Lucas, and M. Lugert, "GALILEO: satellite system design and technology developments," European Space Agency, Tech. Rep., Nov. 2000.

[13] W.-S. Soh and H. S. Kim, "QoS provisioning in cellular networks based on mobility prediction techniques," in *Proc. World Telecommunications Congress (WTC)*, Paris, France, Sep. 2002.

[14] ——, "QoS provisioning in cellular networks based on mobility prediction techniques," *IEEE Commun. Mag.*, pp. 86–92, Jan 2003.

[15] N. D. Tripathi, J. H. Reed, and H. F. Vanlandingham, "Handoff in cellular systems," *IEEE Pers. Commun.*, pp. 26–37, Dec. 1998.

[16] Y. Zhao, *Vehicle Location and Navigation Systems*. Artech House, 1997, ch. 4.

[17] W. Kim, G.-I. Jee, and J. G. Lee, "Efficient use of digital road map in various positioning for ITS," in *Proc. IEEE Position Location and Navigation Symposium*, Mar. 2000, pp. 170–176.

[18] C.-T. Chou, K. G. Shin, "Analysis of adaptive bandwidth allocation in wireless networks with multilevel degradable quality of service," *IEEE Trans. Mobile Computing*, vol. 3, no. 1, pp. 5–17, Jan. 2004.

[19] X. Wang, R. Ramjee, and H. Viswanathan, "Adaptive and predictive downlink resource management in next generation CDMA networks," in *Proc. IEEE INFOCOM*, Hong Kong, China, Mar. 2004.

[20] A. Bhattacharya and S.K. Das, "LeZi-update: an information-theoretic framework for personal mobility tracking in PCS networks," *ACM/Kluwer Wireless Networks J.*, vol. 8, no. 2-3, pp. 121–135, Mar. 2002.

Wee-Seng Soh (S'95–M'04) received the B.Eng. (Hons) and M.Eng. degrees in electrical engineering from the National University of Singapore in 1996 and 1998, respectively. In 1998 he was awarded the Overseas Graduate Scholarship by the National University of Singapore to study at Carnegie Mellon University, where he obtained his Ph.D. degree in electrical and computer engineering in 2003.

Since 2004, he has been with the Department of Electrical and Computer Engineering, National University of Singapore, where he is currently an Assistant Professor. Prior to joining NUS, he was a postdoctoral research fellow in the Electrical Engineering and Computer Science Department at the University of Michigan. His research interests include architectures, protocols, and performance analysis of wireless ad hoc and sensor networks, as well as cellular networks.

Hyong S. Kim (S'84–M'90) received the B.Eng. (Hons) degree in electrical engineering from McGill University in 1984, and the M.A.Sc. and Ph.D. degrees in electrical engineering from the University of Toronto in 1987, and 1990, respectively.

Since 1990, Dr. Kim has been at Carnegie Mellon University, where he is currently the Drew D. Perkins Chaired Professor of Electrical and Computer Engineering. His primary research areas are advanced switching architectures, fault-tolerant, reliable, and secure network architectures, and optical networks. His pioneering work on switch architectures has influenced many switching system designs in telecom industry. His Tera ATM switch architecture developed at CMU has been licensed for commercialization. He worked in Northern Telecom in 1992 as a research consultant addressing issues in high-speed network architectures. In 1995, Dr. Kim founded Scalable Networks, a Gigabit-Ethernet switching startup. Scalable Networks was later acquired by FORE Systems in 1996. He was at FORE Systems working on the company's technology roadmap until 1998. In 2000, Dr. Kim founded AcceLight Networks, an optical switching startup, and was CEO of AcceLight Networks until 2002. He is an author of over 70 published papers and holds over 10 patents in networking technologies. He was an editor for IEEE/ACM Transactions on Networking from 1995 to 2000.