Predictive Routing of Contexts in an Overlay Network

Hahnsang Kim and Kang G. Shin
Real-Time Computing Laboratory
Department of Electrical Engineering and Computer Science
The University of Michigan
Ann Arbor, MI 48109-2121, USA
{hahnsang, kgshin}@eecs.umich.edu

***Abstract*—While mobile nodes (MNs) undergo handovers across inter-wireless access networks, their contexts must be propagated for seamless re-establishment of on-going application sessions, including IP header compression, secure Mobile IP, authentication, authorization, and accounting services, to name a few. Routing contexts via an overlay network either on-demand or based on prediction of an MNs' mobility, introduces a new challenging requirement of context management. This paper proposes a *context router* (CXR) that manages contexts in an overlay network. A CXR is responsible for (1) monitoring of MNs' cross-handover, (2) analysis of MNs' movement patterns, and (3) context routing ahead of each MN's arrival at an AP or a network. The predictive routing of contexts is performed based on statistical learning of (dis)similarities between the patterns obtained from vector distance measurements. The proposed CXR has been evaluated on a prototypical implementation based on an MN mobility model in an emulated access network. Our evaluation results show that the prediction mechanisms applied on the CXR outperform a Kalman-filter-based method [34] with respect to both prediction accuracy and computation performance.**

I. Introduction

Inter-wireless technologies, ranging from IEEE 802 networks such as Wi-Fi, WiMax, and personal area networks, to non-802 networks such as cellular networks, are rapidly converging. This trend has led mobile users to carry multimedia-access devices that operate across heterogeneous networks, without disrupting on-going sessions. At the same time, the IEEE 802.21 Standard [13] puts the ability of *vertical* and *horizontal handovers* in practice—handovers between domains with different management policies are referred to as *cross-handovers*. Furthermore, mobile applications impose new challenging requirements of reducing additional delays [19], e.g., fast key establishment between the involved entities for secure network accesses. A 'context' that contains such information is essential to fast re-establishment of involved flows/sessions as MNs cross domain boundaries. In particular, a mobile's context is routed to a target point of attachment (e.g., an access point (AP)) ahead of its arrival, avoiding disruption of on-going sessions. Contexts of this kind vary with the underlying applications, requiring an effective and scalable context management.

A network-layer-based protocol, called CXTP [23], is specified for the purpose of routing contexts. CXTP provides an option for coping with *smooth handovers* of MNs equipped with inter-wireless technologies, in conjunction with the IEEE 802.21. At the same time, keeping on-going application sessions without disruption requires a 'map' via which to make corresponding contexts available to a target AP before the mobile's arrival. The map can be represented by a neighbor graph [24] that exhibits the logical connectivity of APs based on MN paths. Furthermore, smooth handovers can be facilitated by the accurate prediction of the MN's location in future. There has been extensive research on prediction mechanisms, e.g., applying a Kalman filter [18], a sequential Monte Carlo filtering [36], or via a probabilistic modeling [9]. Despite this extensive research, a bridge to establish between the prediction and the efficient routing mechanisms still lacks, thus requiring context management independently of inter-wireless access networks.

There are two main challenges in developing an integrated framework for context management. First, a framework should be able to route contexts to corresponding 'receivers' (e.g., routers) even when the prediction turns out to be false; at the same time, the overhead of the routing is kept to a minimum. Correct extraction of features from MNs' movements is of utmost importance for prediction. Second, the framework should be easy to deploy, i.e., its deployment should not require any new infrastructure.

In this paper we propose a *context router* (CXR) that monitors, detects, and analyzes MNs' movements and cross-handovers, consisting of monitoring, analysis, and routing components. The monitoring component observes and detects the likelihood of an MN's cross-handover and then tracks directional changes in the MN's movements, resulting in an association pattern. The analysis component gauges (dis)similarities between the observed pattern and the *a priori* established pattern associated with a distribution of CXR candidates, narrowing the CXR candidates down to a few. The routing component then routes (the replicas of) a corresponding context to the selected candidates. We emulate an MN's movements over a grid of an access network, based on a specified mobility model, then evaluating the accuracy of our prediction mechanisms.

The contributions of this paper are two-fold, addressing the

978-1-4244-3487-9/09/$25.00 © 2009 IEEE

two challenges listed above. First, the CXR keeps track of changes in association with APs during an MN's movements, generating a *time-history vector of angles* (TVA). The TVA is an abstraction of an MN's movement pattern with respect to a monitoring level, i.e., the focus of the MN's movements is on APs rather than its actual location. This abstraction is effective for computation performance with granularity of a measuring time period and also allows an overlay of CXRs to be independent of the access network, gradually learning its topology and thus easing the deployment of CXRs. χ^2-distance or edit distance is calculated to gauge (dis)similarity of an observed pattern to a database of movement patterns, enhancing the true-positive rate. Second, the deployment of even only a few CXRs exhibits effective context manageability and scalability; it also improves the compatibility with standard protocols such as CXTP [23] and the IEEE 802.21 Standard [13].

The rest of this paper is organized as follows. Section II provides a background of a context transfer protocol, the concept of contexts and the applications using them. Section III describes a mobility model describing MNs' movements. Section IV describes the design of the CXR that consists of the monitoring of the MNs' movement patterns and cross-handovers, the analysis of movement patterns, and two routing methods. Also, implementation issues are discussed there. Section V evaluates the accuracy of our prediction mechanisms. We discuss the related work in Section VI and conclude the paper in Section VII.

II. Preliminaries

A. Context Transfer Protocol

CXTP [23] is a protocol by which contexts are sent and received with the aim of quickly re-establishing context transfer-candidate services, such as those of AAA registration keys for Mobile IP [28], IP header compression [7], [35], and AAA message exchange for the IEEE 802.1X [6], [25], without requiring an MN to explicitly perform all protocol flows for these services from scratch. CXTP deals with two distinct scenarios. In the first scenario, a previous access router (pAR) receives a context transfer (CT) trigger from the MN with the help of the IEEE 802.21 Standard and then predictively routes a context-transfer-data (CTD) message containing an involved context to a next access router (nAR), called 'predictive routing.' However, how to select the target nAR in a predictive manner is beyond the scope of the specification of CXTP—filling this gap is part of goals of this paper. In the second scenario, when the predictive routing in the first scenario fails, an nAR receives a CT trigger from the MN, then sending a context transfer request (CT-Req) message to the pAR. In response to the CT-Req message, the pAR routes a CTD message to the nAR that replies with a CTD-reply (CTDR) message. The second scenario is called 'on-demand routing'. These two routing methods are elaborated when the design of our framework is presented.

B. Contexts and Applications

Three types of contexts, depending on the transfer-candidate services, are considered. First, a context for Mobile IP contains an AAA registration key required for a secure association [28]. When an MN undergoes handovers and attaches itself to an foreign agent (FA), the MN requests a registration key from its home AAA server via the FA. With the registration key received, the MN verifies a reply message from the FA, creating a mobility security association with the FA [28]. However, since the generation of the registration key requires message exchanges via the AAA infrastructure [4] (e.g., a foreign AAA server contacts the MN's home AAA server), a significant delay is incurred [35]. Such delays can be greatly reduced by routing a context containing the registration key to the FA from the MN's home agent (HA) or the previous FA.

Second, a context for IP header compression [7] contains a compression state used for compression or decompression, current values of all relevant header fields (e.g., the presence/absence of IPv4 header or UDP checksums enabled/disabled), information on IP identifier field changes for fragmentation, and increase in sequence number/timestamps in the packet stream. IP header compression is effective for reducing the large header overhead, especially when speech data for IP telephony over a wireless link is carried via RTP [30]. That is, a packet has an IP header (20 octets for IPv4 or 40 octets for IPv6), a UDP header (8 octets), and an RTP header (12 octets) for a total of either 40 or 60 octets. This long header has significant redundancy between not only header fields within the same packet header, but also contiguous packets. With IP header compression applied, in turn, two-fold or three-fold capacity increase, depending on the variability of header size during a context establishment,can be achieved [35], in which the context that results from state machines [7] established on a compressor and a decompressor plays a role. Therefore, when the MN undergoes handovers, routing a context allows the avoidance of establishing such state machines from scratch, resulting in reduction in handover delay.

Third, like the registration keys for Mobile IP, a context for the IEEE 802.1X Standard [6] contains AAA key material. The IEEE 802.1X Standard provides port-based network access control for devices that attempt to attach to a LAN port. The framework that is positioned behind an AP has three options in access control: (1) no authentication in which no filter is applied, (2) WEP [3] that was originally designed to prevent wireless communication from eavesdropping, but in turn has serious security weakness, and (3) open authentication—all packets are filtered out except for EAPOL (EAP over LAN) [6]—in which any of third-party security mechanisms is applied in conjunction with the IEEE 802.11i [1]. In particular, the 802.11i framework is specified jointly with the AAA infrastructure. The AAA client/authenticator (AP) exchanges messages with the AAA server, e.g., RADIUS [29] or Diameter [8], for the authentication of the MN, resulting in a pairwise master key (PMK) of 256-bit size for the AAA

server, the AAA client (AP), and the MN. The PMK is used to derive a pairwise temporary key (PTK), with the AAA client and the MN communicating with each other. The PTK is a per-AP key that is associated with a specific AP, allowing for the protection of the wireless link between the AP and the MN. Thus, the MN's re-association with another AP requires a new PTK, probably, in the case of 'roaming' or cross-handovers, interacting with the AAA server. This interaction incurs a significant delay that is proportional to the distance between two involved AAA servers [14], [15]. Thus, routing a context containing the PMK(s) for authentication reduces the authentication latency [16], [31].

III. Emulating Mobile Nodes

This section presents a simple mobility model by which an MN's movements are emulated over a virtual inter-wireless access network. The presentation of the model is solely for the evaluation purpose. A mobility model typically selects two or more of elements, i.e., speed, distance, angle, destination, and travel time [37], according to some probability distribution that determines movements. Usually, the selection of these elements is independent of a series of movements (of a single MN) and thus a combination of some of these elements and a type of probability distribution to be used for each combination result in different mobility models [37].

A. Mobility Model

Our model selects speed (v), given a time period, that determines a distance with a normal probability distribution. Independently of temporal properties of speed, direction (θ) that only affects the spatial properties of a MNs' distribution is also chosen. Given a location (l) of an MN at discrete time t, our model is specified as:

$$l' = M(l,\ v,\ \theta,\ t), \tag{1}$$

where time-varying θ_t is determined with a normal probability distribution with mean μ_d and variance σ_d. θ_t is derived from the MN's current direction θ_i as can be seen in Fig. 1. θ_i is calculated, provided the MN's previous and current positions, i.e., $(x_{i-1},\ y_{i-1})$ and $(x_i,\ y_i)$, as follows:

$$\theta_i = \tan^{-1}\left(\frac{y_i - y_{i-1}}{x_i - x_{i-1}}\right). \tag{2}$$

The calculation of each θ_i is based on a *relative anchor*, a base line perpendicular to the previous direction, thus resulting in avoidance of unrealistic abrupt changes in direction. At $t+1$, next direction θ_{i+1} is determined by calculating $\theta_{t+1} + \theta_i - \frac{\pi}{2}$, where θ_{t+1} is also a random variable. The MN's speed, v, is represented by a Gaussian random variable with a mean μ_v and a variance σ_v. The speed is updated at every a given time unit, also representing how far the MN goes.

Fig. 2-(a) shows an example of an MN's movement trace based on the mobility model (Eq.-(1)), with μ_d, σ_d, μ_v, σ_v each set to $\pi/2$, 1, 10mph, and 1.

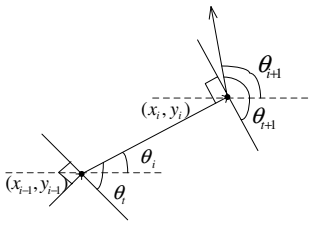
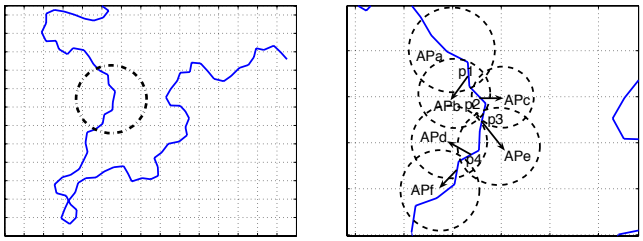


Fig. 1. Relative-anchor-based directional changes



(a) An MN's movement trace (b) Zoomed-in-on circle

Fig. 2. An MN's movement trace and association with APs

B. Underlying Hypothesis of Our Approach

We are interested in a sequence of the MN's association with APs while the MN undergoes handovers. To this end, first, APs assume to be randomly uniformly distributed, covering all pathes of the MN's movements. Second, each AP's signal strength varies, resulting in the different size of their coverage. Under these two assumptions, Fig. 2-(b) shows the MN's association with APs, zooming in on a portion of its trace displayed in Fig. 2-(a). At first, the MN keeps associated with ap_a until reaching at position p_1 in the path. At p_1 at which the ap_b's signal strength is greater than that of ap_a, the MN's association is switched into ap_b. The MN, afterwards, gets connected to ap_c at position p_2 due to the same reason as to ap_b. Just before position p_3, three signals from ap_c, ap_d, and ap_e compete and since ap_e's signal strength at p_3 is greater than that of the other two, the MN associates with ap_e. The cycle of disassociation and association continues based on the signal strength, accordingly resulting in a sequence of the MN's association, i.e., ap_a, ap_b, ap_c, ap_e, ap_d, and ap_f. The sequence is translated into a time-varying sequence with sojourn time specified at each AP, e.g., $(ap_a,\ t_a)$.

IV. The Context Router

The CXR is responsible for monitoring the MN's movements, detecting the likelihood of its cross-handovers, analyzing the MN's movement pattern, and routing the corresponding context to selected CXR(s). Besides doing these, the CXR maintains three databases. Next we describe each of these processes and databases.

A. Monitoring

Timely detection of the likelihood of an MN's cross-handover is of great importance to the predictive routing of contexts. For detection of cross-handovers, a trigger message

is transmitted from the MN to the corresponding CXR via IEEE 802.21. The trigger message, however, does not usually include routing information on the corresponding CXR, e.g., the identity and the IP address of a target CXR. Thus, the corresponding CXR needs to identify the MN, requiring its context. If the information is not available, all relevant protocol flows will be re-established from scratch. Alternatively, the CXR is notified of the cross-handover detection whenever the MN is associated with an AP belonging to the boundary of a domain that has a unique policy for network management (e.g., a security policy and a fast handover policy). The notification can be performed by keeping track of the MN's movements.

Tracking the MN's association with APs results from an AP queue (APQ) with which the MN is registered on the CXR. The CXR, cxr_j, verifies if the AP (with which the MN is associated) is on the domain boundary. To this end, a boundary neighbor register (BNR) on cxr_j is defined as a triple (ap_j, cxr_k, ap_k), where ap_j is an AP on the boundary that cxr_j controls and ap_k is an AP that cxr_k controls, adjacent to ap_j. The relation between ap_j, cxr_k, and ap_k is established and registered when the MN undergoes a cross-handover from ap_j to ap_k. This, however, does not necessarily mean that cxr_j and cxr_k are physically adjacent to each other; they are "virtual" neighbors to each other (i.e., one reaches the other through an n-hop route) while cxr_k also adds (ap_k, cxr_j, ap_j) to its BNR.

For an MN's directional pattern, a time-varying association sequence results in a time-history vector of angles (TVA). For instance, given a time-varying association sequence [(ap_a, 4), (ap_b, 5), (ap_c, 5), (ap_e, 4), (ap_d, 6), (ap_f, 4)], similarly to an approach found in [9], the sequence is first interpreted as [ap_a, ap_b, ap_c, ap_e, ap_d, ap_d, ap_f] with the time interval set to 3. Once the association sequence is provided, the difference in angles of a pair of ap_a and ap_b, i.e., θ_{ab}, is calculated by applying Eq. (2). Likewise, θ_{bc} is calculated with ap_b and ap_c. This calculation repeats until the last with a pair of ap_d and ap_f completes. This way, the corresponding TVA is yielded as [θ_{ij}, θ_{jk}, $\cdots$, θ_{mn}]. Next, the direction changes are encoded by quantizing angles:

$$\frac{2\pi}{n}(k-1) \leq \theta < \frac{2\pi}{n}k,\ k \leq n \in \mathbb{Z}. \tag{3}$$

Each direction, then, is labeled with an index of the quantized angle, k. Throughout the remainder of this paper, angles are assumed to be partitioned into eight directions (n=8). Note, however, that this model can be generalized with the number of directions adjusted. As a result of the encoding, the corresponding TVA is transformed into an index sequence [i, k, $\cdots$, m], where i, k, and $m \leq n$.

B. Analysis

An index sequence corresponding to a TVA is either added, as a new pattern, to a pattern data register (PDR), or discarded when found similar, given a threshold presented in Eq. (5), to another pattern in the PDR. The PDR, thus, is a set of entities each of which consists of a unique pattern and a corresponding distribution. The distribution, pd, is formed by pairs of a neighbor CXR and a statistical degree, i.e.,

$$pd = \{(cxr_1,\ dg_1), \cdots,\ (cxr_n,\ dg_n)\}. \tag{4}$$

Using this distribution, we select neighbor CXRs to which a pattern exhibiting MN is likely to be bound with a certain probability. The selection of CXRs involves two main steps: matching patterns and filtering out CXRs that have a low degree.

1) Matching Patterns: Matching patterns is to gauge (dis)similarities between an observed TVA and each pattern stored in the PDR. The similarity is represented by a distance that can be calculated by *editing* one pattern to make it same as the other (i.e., edit distance [11]), or alternatively applying χ^2-distance [11]. The advantage of the two methods, taken into account in this paper, is to provide a threshold, given two pattern sequences a and b, with which to adjust the similarity:

$$f(a,\ b) < \delta_d, \tag{5}$$

where f is either the editing function, e, or the χ^2-distance calculating function. The smaller the δ_d, the more fine-grained matching can be achieved, and the more specific the group in which the TVAs are accepted as same. In this sense it is important to group TVAs based on their unique feature.

Function e takes two indexed TVAs: $a_1 \cdots a_n$ and $b_1 \cdots b_n$, forming an $(n+1) \times (n+1)$ matrix. The matrix is set as follows: $e(1,\ 1) = 0$, $e(1,\ b_1 \cdots b_n) = 0$, and

$$e(a_1 \cdots a_i, 1) = \begin{cases} 1, & i = 1 \\ \infty, & 1 < i \leq n. \end{cases}$$

To make a same as b, then, a sample in a is replaced or deleted, or the same as a sample in b is inserted, repeatedly, until the end of a. Function e is specified as

$$e(a_i,\ b_j) = \begin{cases} e(a_{i-1},\ b_{j-1}) & \text{if } a_i = b_j \\ min \begin{pmatrix} e(a_{i-1},\ b_{j-1}) + C_r, \\ e(a_{i-1},\ b_j) + C_d, \\ e(a_i,\ b_{j-1}) + C_i \end{pmatrix} & \text{if } a_i \neq b_j, \end{cases} \tag{6}$$

where C_r, C_d, and C_i are replacing, deleting, and inserting costs, respectively. They all are set to 1 in our evaluation.

Alternatively, χ^2-distance-based techniques are found in diverse areas, such as scene-change detection in image sequences [27] and anomaly detection [17]. It is simple to calculate the χ^2-distance:

$$\chi^2(a,\ b) \ = \ \sum_{i=1}^{n} \frac{(a_i \ - \ b_i)^2}{(a_i \ + \ b_i)}. \tag{7}$$

Clearly, $\chi^2 = 0$ if and only if all of samples in a match those of b. The higher the value of χ^2, the less likely the observed TVA fits the expected pattern. Computation performance of applying the χ^2-distance function appears better than the editing function, as we will show in Section V.

2) Filtering: A pattern that matches an observed TVA results in the corresponding distribution (see Eq. (4)). This distribution reflects the likelihood of which neighbor CXRs the MN will be bound to, allowing for selectively choosing neighbor CXRs to route the MN's context. Either of two following filtering methods for the selection of neighbor CXRs is applied on each CXR. First, we use a filter based on Chebyshev inequality [26]. The Chebyshev inequality, given a normalized pd with a random variable X with a finite mean μ and a finite standard deviation σ, provides a relation between σ and $|X - \mu|$, such that

$$P(|X - \mu| \geq \frac{\sigma}{\sqrt{\delta_p}}) \leq \delta_p. \quad (8)$$

The inequality relation holds for any dropout rate $\delta_p > 0$. This inequality-based filter is effective if pd is a normal distribution [26], reflecting a strong relationship between a given pattern and its association with a neighbor CXR.

Alternatively, a cutoff-based filter is a simple filter using a threshold, δ_r, on a scale of 0 to 1 by which top n % neighbor CXRs from the normalized pd are chosen. When $\delta_r = 0$, no CXRs are selected, so only the on-demand routing, which will be described shortly, remains effective. When $\delta_r = 1$, all neighbor CXRs are flooded with contexts. When $0 < \delta_r < 1$, contexts are routed to selected neighbor CXRs.

C. Routing

After filtering is performed, (the replicas of) contexts are routed to selected neighbor CXRs either in a predictive manner or on-demand. Contexts are stored in a context register (CR) on each corresponding neighbor CXR. If the MN arrives at one of the selected neighbor CXRs (nCXR) in which the MN's context is immediately available, it will not experience any extra delays in fetching its context from the previous CXR (pCXR) (i.e., predictive routing). In contrast, the MN may be eventually bound to an unexpected neighbor CXR, i.e., a context miss. When a context miss occurs, the CXR fetches the corresponding context from the pCXR (i.e., on-demand routing). After the context is routed either in a predictive manner or on-demand, the association between the pCXR and the nCXR, and between their APs are added to, or updated on both the pCXR's and the nCXR's BNRs. The same holds for the corresponding distribution (Eq. (4)) in their PDR.

D. Limiting

The growth of context replicas varies, depending on MNs located on the domain boundary and their cross-handover pattern. In particular, some MNs are likely to cross the boundary and eventually stay thereon or cross back while some MNs may cross the boundary back and forth in a short time period, i.e., a *ping-pong effect*. These events cause a rapid increase in the number of replicas. In order to limit their excessive growth, two rules are applied. The first rule is to limit the buffer size for replicas. When the buffer is full, the replicas in the buffer are invalidated in the least-recently-used order—invalidation is less costly than deletion—by simply toggling off a validity bit. This rule is effective against the rapid growth of replicas. The second rule is to limit the time in keeping the replicas in the CR. That is, incoming replicas are timestamped with a timer threshold. When the timer expires, the replicas in the CR are scanned in comparison with the threshold, the timeout, and the timestamped time, invalidating stale replicas. This rule has a tradeoff between storage savings and performance.

E. Implementation

Fig. 3 illustrates the software implementation of the CXR, and its interfaces to another CXR via CXTP [23] and to MNs via the IEEE 802.21 standard (that is responsible for handover negotiation and layer-2 connectivity).

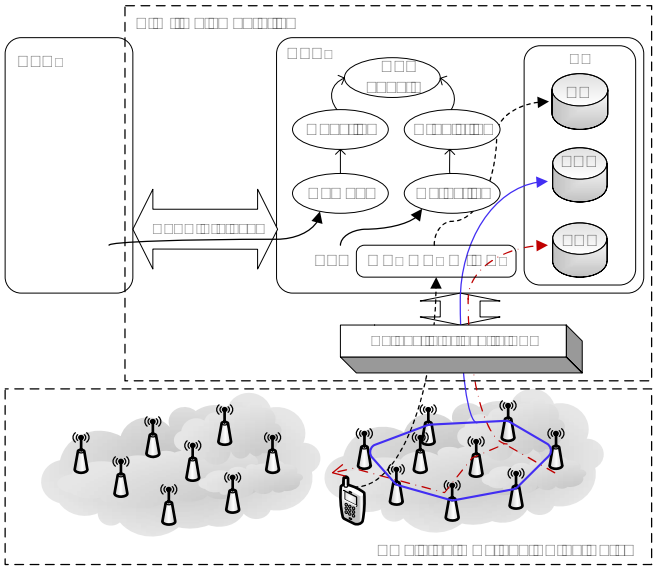


Fig. 3. The CXR software implementation. An access network is formed by APs whose location is represented by an xy-coordinate on a grid. Individual MNs move over the grid based on their own mobility model with the values of system parameters changed.

The APQ on the CXR is a registration list on which MNs currently staying in the domain are recorded along with their sojourn times and association with APs; using this information, MNs' movements are tracked. Given a time interval, the APQ is scanned to find whether any MN is associated with APs inside the boundary in reference to the BNR. If this is the case, the detection state is transitioned to the prediction state. In the prediction state, the predictive routing of contexts is performed according to the cross-handover probability (e.g., success in predictive routing as independent Bernoulli trials). If the cross-handover probability is greater than the threshold, the prediction algorithms are applied, routing the context. The pseudo code of the predictive routing is given in Fig. 4. On the other hand, the demand state is initiated by a request from other CXRs (e.g., by a CT-Req message in CXTP), ultimately being transitioned to the reaction state in which the on-demand routing is performed.

1: δ_p and δ_c: probability threshold;
2: δ_c: cutoff threshold;
3: $eProb$: an estimated crossover probability;
4: pd: the probability distribution from the BNR;
5: Cxt: a context from the CR;
6: **if** $eProb > \delta_p$
7: **if** Cutoff turned-on
8: $CXRs \leftarrow$ CutOffSelection(δ_c, pd);
9: **else**
10: $CXRs \leftarrow$ StatProbSelection(pd);
11: **end if**
12: Route Cxt to $CXRs$;
13: **end if**

Fig. 4. Pseudo code: Predictive routing of contexts

V. Evaluation

A success in the predictive routing of contexts implies substantial reduction in handover delays. Such delays, especially involving message exchanges by underlying applications, greatly vary, depending on two end-points of their communication. For instance, in the case where two AAA servers are involved for secure communications, the average delay in establishing security association is proportional to the round-trip time between the two servers [14]; our evaluation focuses on prediction accuracy. At the same time, since a great deal of MNs cross the domain boundary, computation performance for the prediction mechanisms is also an important factor to evaluation. Thus, the metrics used to indicate prediction and computation performance include accuracy and CPU utilization. The accuracy is represented by $\frac{TP+TN}{TP+TN+FP+FN}$, where TP (true positive), TN (true negative), FP (false positive), and FN (false negative) are respectively the number of times the selection of a neighbor CXR is correctly estimated, an MN's staying at a CXR is correctly estimated, an MN is bound to an unpredicted CXR, and an MN leaves its current CXR although it was predicted to stay thereon.

We evaluate prediction accuracy compared with the Kalman filter [34]. In applying the Kalman filer, a state is specified by the MN's location (x and y) and speed (dx/dt, dy/dt), and we set values in process and measurement noise covariances, Q and R, to 0.1 and 1, respectively. In addition to these parameters, we set values in transition matrix A and measurement matrix H as

$$\begin{pmatrix} 1 & 0 & t & 0 \\ 0 & 1 & 0 & t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}.$$

Fig. 5 shows the effectiveness of using the Kalman filter; estimation accuracy is determined by two factors: measurement and past states. At the time of initialization, however, the past state is unknown, and hence, the Kalman filter performs in an ad hoc manner. Even with a poor initialization, it quickly converges to the true values.

In this section, we first describe the emulation of MNs and access networks. Then, after assessing system parameters

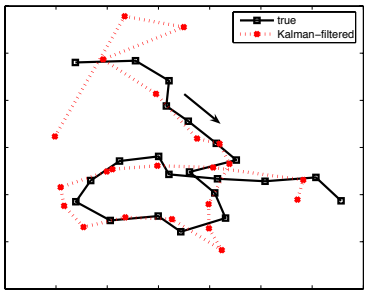


Fig. 5. Movement path estimation with the Kalman filter

defined in our prediction mechanisms, prediction accuracy is evaluated with the local optimal values of the system parameters obtained. Finally, we analyze computation performance for the prediction mechanisms.

A. Emulation of MNs and Access Networks

APs are assumed to be randomly and uniformly distributed, forming a virtual access network over a grid. On the grid, each MN moves, according to its own mobility model in Eq. (1), associated with different APs. At a given time t, the MN's location is obtained and then the location of the AP closest to the MN is calculated by applying a simple 2-level hash function such that APs on the grid are filtered based on the MN's x coordinate, resulting in an array of APs closest to the x coordinate. The array is again filtered based on the MN's y coordinate, thereby obtaining the location of the AP closest to the MN. This way, at $t + \Delta t$, the location of the MN and the corresponding AP are calculated. The time interval, Δt, is adjusted, based on the MN's maximum speed, V_{max}, in Eq. (9). Δt should be small enough to capture a series of the MN's association with APs. Clearly, as V_{max} becomes large, Δt should be small. Thus, the granularity of Δt is determined as

$$\frac{d}{V_{max}} = \Delta t, \tag{9}$$

where d is the distance between the AP with which the MN is currently associated and the point at which the MN's association was changed. Obviously, d is smaller than the distance, D, between the centers of two involved cells, assuming that the two cells' coverage areas (circles) overlap to the extent that $d = \frac{D}{2}$.

B. Assessment of System Parameters

System parameters considered for the evaluation include the TVA length combined with the matching threshold δ_d, the cutoff threshold δ_r, and the dropout rate δ_p. In order to assess these parameters, we create 100 MNs that move, based on their mobility model, and observe their movement patterns while each MN undergoes handover 1000 times. First, when the TVA length increases from 6 to 15 with δ_d set so as to let two TVAs match 70% of times, the 9- and 13-length TVAs yield a local highest prediction accuracy. When δ_d is set to let 60% of two TVAs match, the 11-length TVA results in a local highest prediction accuracy. Thus, both the TVA length and

its threshold must be tuned together for the best results, based on the edit distance technique applied. We set the TVA length and δ_d to 9 and 70%, respectively, throughout the evaluation. In addition to these parameters, we set $\delta_r = 2$ and $\delta_p = 0.5$ which, in turn, appear almost constant.

For the ping-pong effect, the predictive routing is applied in general even for the case where the MN has just crossed the domain boundary, and yet, depending on the cross-handover probability, the threshold needs to be adjusted. That is, when the MN is unlikely to cross back immediately, the predictive routing may not be applied just after the MN's cross-handover. It is, therefore, important to find out an optimal cross-handover probability threshold that is tuned to MNs' movement patterns. Fig. 6 shows the relation between the threshold and the false alarm rate including the false negative and the false positive rates. To some degreee, the lower the threshold, the more aggressive in the predictive routing—it occurs more frequently. When the threshold decreases below the value of 0.3, the false alarm rate goes up to 0.44, implying more MNs that have crossed the boundary are likely to stay thereon although they are predicted to cross back. When the threshold increases above the value of 0.4, the false alarm rate also increases up to 0.49, implying more MNs are likely to cross back although they are predicted to stay thereon. Accordingly, the cross-handover probability threshold ranging from 0.3 to 0.4 is determined to cope with the ping-pong effect.

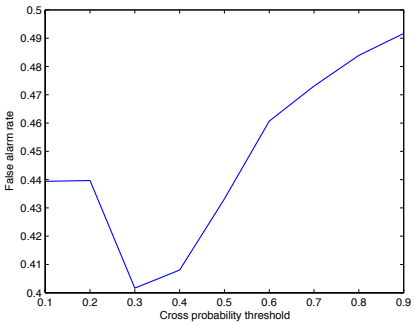


Fig. 6. False alarm vs. cross-handover probability threshold

C. Evaluation Results

Section IV described 4 prediction mechanisms: cutoff-based and inequality-based mechanisms are combined with the edit-distance and the χ^2-distance functions. Prediction accuracy is related to not only the similarity measurement techniques, but also the classification of patterns into groups. Each group is represented by a probability distribution pd on which the prediction mechanisms depend. In general, the inequality-based mechanisms are effective for the distribution normally distributed with even a large variance, while the cutoff-based mechanism operates well with various distributions. As shown in Table I, prediction accuracy offered by the inequality-based mechanisms almost equals that by the cutoff-based ones. In which case, the selection of a similarity measurement technique determines the performance of prediction accuracy. That is, applying the edit distance technique achieves up to 1.4 times as high prediction accuracy as applying χ^2-distance.

Combined prediction mechanisms	Accuracy
Cutoff + edit-distance	0.5187
Inequality + edit-distance	0.5367
Cutoff + χ^2-distance	0.4010
Inequality + χ^2-distance	0.3903
The Kalman filtering	0.3480
The flooding	0.5695

TABLE I
AVERAGE PREDICTION ACCURACY COMPARISON

Since the Kalman filtering yields an estimated next location of the MN, the AP closest to the estimated location is selected and then the neighbor CXR that controls the AP is selected. Note that the AP may not be on the domain boundary, resulting in being unavailable in the BNR. In general, this case requires additional information on binding between a neighbor CXR and its APs to be available to another CXR in practice. In contrast, our prediction mechanisms capture an AP association pattern based on the MN's movements, ultimately limiting the number of neighbor CXRs to which to route a corresponding context; thus for fairness purpose, the cutoff-based and inequality-based mechanisms select one neighbor CXR. Also, the Kalman-filter-based estimation is deferred until the BNR is populated and the same holds for our mechanisms. As shown in Table I, the inequality-based mechanisms with the edit distance and the χ^2-distance functions outperform the Kalman filtering by 54% and 12%, respectively. Also, little difference in prediction accuracy is made between the inequality-based mechanism with χ^2-distance applied and the flooding method of routing contexts to all neighbor CXRs (i.e., $\delta_r = 1$) by 6%. Rather, our mechanisms offer an advantage over the flooding method in that the storage overhead for the flooding method is proportional to the number of neighbor CXRs, while the overhead of our mechanisms is almost constant.

Combined prediction mechanisms	CPU utilization
Cutoff + edit-distance	44.58%
Inequality + edit-distance	43.91%
Cutoff + χ^2-distance	149.40%
Inequality + χ^2-distance	152.29%
The Kalman filtering	41.89%

TABLE II
AVERAGE CPU UTILIZATION COMPARISON

When it comes to computation performance, the χ^2-distance-applying mechanisms outperform the edit-distance mechanisms most requiring comparison operations, at the expense of prediction accuracy. As shown in Table II, regarding the cutoff-based mechanisms, applying χ^2-distance achieves 3 times as good CPU utilization as applying the edit distance technique. The same also holds for the inequality-based mechanisms by 3.5 times. Furthermore, all the 4 prediction mechanisms are proven to be more (up to 3.6 times) effective in CPU utilization than the Kalman-filtering-based mechanism. Accordingly, the inequality-based mechanism with χ^2-distance is the best option for computation performance, while the cutoff-based or the inequality-based mechanism with the edit distance technique applied is the one for prediction accuracy.

VI. Related Work

We now review related work that was not previously discussed in Section II. Extensive studies on context routing have been done for various purposes, e.g., roaming management [20], intra-domain fast handover [5], [12], secure connectivity [25], but we will focus on prediction mechanisms, since CXR is a predictive context routing framework.

A variety of techniques for improving mobility prediction in cellular networks have been proposed, e.g., use of user mobility profile [2], [22] which is a combination of historic route records and predictive patterns of an MN's paths, and use of road topology knowledge [32] and an MN's positioning information [21], [32]. The same importance of mobility prediction to service provisioning, e.g., bandwidth reservation, also holds in wireless local area networks. Song *et al.* [33] presented a case study that quantifies the possible gains with mobility prediction in a VoIP application. Chou and Shin [10] improve a packet buffering-and-forwarding mechanism, thereby resulting in smooth handovers.

VII. Conclusion

As mobile nodes move around, their contexts must be effectively and efficiently managed either on-demand or predictively (proactively). In this paper we presented a context router (CXR), an integrated framework for context management, with the aim of furthering users' mobility and seamless services of their applications on their mobile devices. We began by characterizing mobiles' movement patterns and then designed the CXR with three important steps. We analyzed prediction accuracy for routing (the replicas of) the corresponding contexts ahead of mobiles' arrival, and comparatively evaluated computation performance with our prediction mechanisms and the Kalman filter. We addressed two challenges: (1) providing an integrated framework combined with the context routing methods and the prediction mechanisms with mobiles' movement patterns extracted from association with APs, and (2) deriving the efficacy, deployability, and scalability of context management. Our CXR is shown to seamlessly provide services with proper contexts.

References

[1] Bernard Aboba et al. Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications: Specification for robust security. Technical Report IEEE Std 802.11i/D3.1, IEEE, 2003.

[2] Ian Akyildiz and Wenye Wang. The predictive user mobility profile framework for wireless multimedia networks. *ToN*, 12(6):1021–1035, Dec. 2004.

[3] WEP algorithm. http://www.isaac.cs.berkeley.edu/isaac/wep-faq.html.

[4] Authentication Authorization and Accounting IETF WG. http://www.ietf.org/html.charters/aaa-charter.html.

[5] Novella Bartolini and Emiliano Casalicchio. A performance analysis of context transfer protocols for qos enabled internet services. *Computer Networks: J. of CTN*, 50(1):128–144, Jan. 2006.

[6] Les Bell et al. IEEE Standard for Local and metropolitan networks-Port-Based Network Access Control. Technical Report ANSI/IEEE Std 802.1X-2001, IEEE, 2001.

[7] Carsten Bormann et al. RObust Header Compression (ROHC): Framework and four profiles: RTP, UDP, ESP, and uncompressed. RFC 3095, Jul. 2001.

[8] Pat R. Calhoun, John Loughney, Erik Guttman, Glen Zorn, and Jari Arkko. Diameter Base Protocol. RFC 3588, Sep. 2003.

[9] Sunghyun Choi and Kang G. Shin. Predictive and adaptive bandwidth reservation for hand-offs in QoS-sensitive cellular networks. In *SIGCOMM*, pages 155–166, Vancouver, British Columbia, Sep. 1998. ACM.

[10] Chun-Ting Chou and Kang G. Shin. Smooth handoff with enhanced packet buffering-and-forwarding in wireless/mobile networks. *Wirel. Netw.*, 13(3):285–297, 2007.

[11] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. ISBN 0-471-05669-3. Wiley-Interscience, second edition, 2001.

[12] Ha Duong, Arek Dadej, and Steven Gordon. Proactive context transfer and forced handover in ieee 802.11 wireless lan based access networks. *SIGMOBILE: MC2R*, 9(3):32–44, 2005.

[13] Vivek Gupta, Subir Das, and David Cypher. http://www.ieee802.org/21/.

[14] Hahnsang Kim and Hossam Afifi. Improving Mobile Authentication with New AAA Protocols. In *ICC*, pages 497–501, Anchorage, USA, May 2003. IEEE.

[15] Hahnsang Kim, Walid Ben-Ameur, and Hossam Afifi. Toward efficient mobile authentication in wireless inter-domain. In *ASWN*, Berne, Switzerland, July 2003.

[16] Hahnsang Kim, Kang G. Shin, and Walid Dabbous. Improving cross-domain authentication over wireless local area networks. In *SecureComm*, pages 127–138, Athens, Greece, Sep. 2005. IEEE.

[17] Hahnsang Kim, Joshua Smith, and Kang G. Shin. Detecting energy-greedy anomalies and mobile malware variants. In *MobiSys*, pages 239–252, Breckenridge, Colorado, USA, Jun. 2008. ACM.

[18] Minkyong Kim, David Kotz, and Songkuk Kim. Extracting a mobility model from real user traces. In *INFOCOM*, Barcelona, Apr. 2006. IEEE.

[19] Rajeev Koodli and Charles E. Perkins. Fast Handovers and Context Transfers in Mobile Networks. *SIGCOMM: CCR*, 31(5), 2001.

[20] Minsoo Lee and Sehyun Park. A context-aware seamless interoperator roaming management framework in 4g networks. *Trans. on Comm.*, E90-B(11):3015–3023, 2007.

[21] Ben Liang and Zygmunt Haas. Predictive distance-based mobility management for pcs networks. *ToN*, 11(5):718–732, Oct. 2003.

[22] Tong Liu, Paramvir Bahl, and Imrich Chlamtac. Mobility modeling, location tracking, and trajectory prediction in wireless atm networks. *JSAC*, 16(6):922–936, Aug. 1998.

[23] John Loughney, Madjid Nakhjiri, Charles Perkins, and Rajeev Koodli. Context transfer protocol. RFC 4067, July 2005.

[24] Arunesh Mishra, Minho Shin, and William Arbaugh. Context Caching using Neighbor Graphs for Fast Handoffs in a Wireless Network. In *INFOCOM*, Hong Kong, Mar. 2004. IEEE.

[25] Arunesh Mishra, Minho Shin, and William Arbaugh. Pro-active Key Distribution using Neighbor Graphs. *Wireless Comm. Magazine*, 11(1):26–36, Feb. 2004.

[26] Athanasios Papoulis and S. Unnikrishna Pillai. *Probability, Random Variables and Stochastic Processes*. ISBN 0-07-366011-6. McGraw-Hill, fourth edition, 2002.

[27] N.V. Patel and I.K. Sethi. Compressed video processing for cut detection. *Vision, Image and Signal Processing*, 143(5):315–323, October 1996.

[28] C. Perkins and P. Calhoun. AAA registration keys for mobile IP. work in progress, IETF Draft, 2003.

[29] Carl Rigney, Steve Willens, Allan C. Rubens, and William A. Simpson. Remote authentication dial in user service. RFC 2865, June 2000.

[30] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. A transport protocol for real-time applications. RFC 1889, Jan. 1996.

[31] Minho Shin, Justin Ma, Arunesh Mishra, and William A. Arbaugh. Wireless network security and interworking. *Proceedings of the IEEE*, 94(2):455–466, Feb. 2006.

[32] Wee-Seng Soh and Hyong S. Kim. Dynamic bandwidth reservation in cellular networks using road topology based mobility predictions. In *INFOCOM*, Hong Kong, Mar. 2004. IEEE.

[33] Libo Song, Udayan Deshpande, Ulas Kozat, David Kotz, and Ravi jain. Predictability of wlan mobility and its effects on bandwith provisioning. In *INFOCOM*, Barcelona, Spain, Apr. 2006. IEEE.

[34] Greg Welch and Gary Bishop. An introduction to the kalman filter. Technical Report TR 95-041, UNC at Chapel Hill, July 2006.

[35] Cedric Westphal and Rajeev Koodli. IP header compression: A study of context establishment. In *WCNC*, pages 1025 – 1031. IEEE, Mar. 2003.

[36] Zigang Yang and Xiaodong Wang. Joint mobility tracking and hard handoff in cellular networks via sequential monte carlo filtering. In *INFOCOM*, pages 968–975, NYC, June 2002. IEEE.

[37] Jungkeun Yoon, Mingyan Liu, and Brian Noble. Sound mobility models. In *MobiCom*, pages 205–216, San Diego, California, Sep. 2003. ACM.