CORE: Connectivity Optimization via REinforcement Learning in WANETs

Alexander Gorovits, Karyn Doke, Lin Zhang, Mariya Zheleva, Petko Bogdanov
Department of Computer Science, University at Albany—SUNY
Emails:{agorovits,kdoke,lzhang22,mzheleva,pbogdanov}@albany.edu

***Abstract*—While mobile devices are ubiquitous, their supporting communication infrastructure is cost-effective only in densely populated urban areas and is often lacking in rural settings. This lack of connectivity leads to lost opportunities in applications such as rural emergency preparedness and response. Peer-to-peer exchange that uses predictable human mobility can enable delay-tolerant information access in rural settings. We propose, an adaptive distributed solution for device-to-device Connectivity Optimization via REinforcement Learning (CORE) in wireless adhoc networks. Our solution is designed for collaborative distributed agents with intermittent connectivity and limited battery power, but predictable mobility within short temporal horizons. We seek to maximize the utility of connection attempts while keeping the power expenditure within a predefined battery budget. Agents learn to adaptively make automated decisions for when to attempt connections and exchange information, based on a local RL model of their mobility and that of other agents they learn about from exchanges. Using both synthetic and real-world mobility traces, we demonstrate that agents are able to materialize 95% of the possible connections using 20% of their battery and successfully adapting to changes in the underlying mobility patterns within several days of learning.**

I. Introduction

Modern smartphone software typically relies on a client-server communication paradigm and continuous Internet access. There are, however, many non-centralized application settings including communications in rural infrastructure-challenged areas [41], disaster response coordination and information dissemination and coordination for displaced populations [24] and social movements [27]. The above have led to proposals of wireless ad hoc networks (WANETs) and corresponding routing protocols that enable end-to-end delay-tolerant connectivity [14]. The focus in WANET research is often on the network layer, since routing in a constantly evolving ad hoc topology poses significant challenges. Ensuring connectivity at the data link layer in WANETs, however, also poses important challenges, especially for modern energy-hungry smartphone devices utilized for navigation, photography, gaming and augmented reality on top of communication.

Ad hoc connectivity at the data link layer among personal smartphones (peers) hinges on adaptive modeling of human mobility in the context of constrained power resources. Specifically, devices require predictive decision making for *when and where* to attempt to connect to peers, since continuous peer discovery is infeasible in terms of energy demand [11]. Past analysis of mobile phone traces has demonstrated that human mobility, although variable, is largely predictable [23], [31], thus, opening the door for model-driven data link connectivity. However, mobility in rural areas, during social unrest or disaster response and other application scenarios for WANETs, may not be fully stationary. For example, recent analysis of human mobility during natural disasters showed deviations from non-disaster periods [36]. Hence our main research question in this work is: *How to maximize the number of peer-to-peer (p2p) connections in WANETs under battery constraints by adaptive distributed modeling of individual and population mobility?*

978-1-6654-4108-7/21/$31.00 ©2021 IEEE

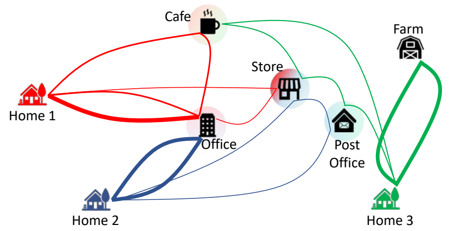


Fig. 1: An example summary of daily trajectories (thicker lines represent more traveled routes) for three people in a rural area where cellular or broadband connectivity is lacking or intermittent. Person 1 (red) and 2 (blue) travel to the same office and back to their homes and occasionally make stops to a cafe, a store and the post office. Person 3 (green) travels to a farm the store, cafe and the post office. The "typical" trajectories are spatio-temporal, i.e. while all three go to the store, they might be going at different times.

Consider, for example, emergency-related information exchange among rural residents from a community with missing or limited broadband and cellular connectivity (Fig 1). Given a finite battery, deciding when to enable wireless interfaces and discover peers is critical for successful p2p connections and exchange with minimal battery footprint. For the example in Fig 1, assume that person 3 (green) has access to the Internet only at their home. They will receive and store up-to-date emergency information on their device at home and it will be critical to learn when and where their typical trajectories intersect with those of other residents (e.g., at the cafe, store and post office) so that the information can be pushed to others at times when they are co-located, albeit offline. While we will keep rural emergency information exchange as our example application, optimizing mobility-informed data link connectivity is equally important in other WANET applications such as connectivity in refugee camps, during disaster-caused network disruptions and as part of social movements coordination.

There are three key challenges in maximizing p2p connections based on predictive mobility modeling. First, the optimization and decision making needs to be *distributed* with partial information at each device. This requirement in

WANETs is by design as the premise is that global connectivity is impossible, hence devices need to "learn" about others' mobility based on encounters. Second, smartphone batteries impose *energy constraints* on peer discovery and connections for exchange. Finally, although human mobility is largely predictable [23], [31], over a long time horizons and upon perturbations (e.g., natural disasters), there may be changes to typical trajectories. Hence, discovery needs to be *adaptive* to individual and global mobility changes.

To address the above challenges, we propose $CORE$ (Connectivity Optimization via REinforcement learning): a data link protocol for WANETs that is distributed, adaptive to changes in device mobility and energy-aware. To enable distributed decisions, we learn (i) a device-specific local and (ii) global trajectory mobility models, where the latter is updated upon successful mobility information exchanges with peers. We employ a reinforcement learning (RL) strategy for connection attempt decisions attuned to recent mobility observations, and thus capable of adapting to changes in the mobility patterns. $CORE$ is energy-aware as it learns to distribute a fixed energy budget for p2p connectivity in time. Using both synthetic and real-world traces, we demonstrate that $CORE$ agents are able to materialize up to 95% of the possible connection opportunities. Agents are also able to quickly adapt to changes in the underlying mobility patterns.

The contributions of this work are as follows:

- We propose an adaptive ad hoc protocol $CORE$ that maximizes device-to-device connections under power usage constraints based on mobility-aware reinforcement learning.
- We characterize a realistic power consumption model for Android smartphones communicating over WiFi Direct.
- We evaluate $CORE$ on synthetic and real-world mobility traces and demonstrate that agents can materialize up to 95% of potential connections with 20% of their battery capacity.
- We demonstrate that $CORE$ can adapt to changes in the underlying mobility, employing only a short window of observations exchanged among devices.

II. Related Work

WANET delay tolerant communication research in dates back to the PRNET project [16] over 5 decades ago. This rich research area encompasses a wide variety of applications: emergency response and coordination [1]–[3], data dissemination [12], [20], [21], defense [17], sensing [25], [28], distributed mobile computation [30] and others. Works addressing challenges in the protocol stack make specific assumptions about the data link layer and focuses on multi-hop routing [14], [37], media access control [22] and multi casting [26]. For example, protocols for routing typically assume a short-term "fixed" topology in which information is routed without global knowledge. Our work on battery-efficient distributed link formation is complementary to protocols for routing as it focuses on predictive modeling of peer connection opportunities.

Modeling and mining human mobility based on real longitudinal traces have demonstrated that human mobility exhibits recurrent patterns and can be successfully predicted with significant accuracy [23], [31]. In the fields of geographical information systems and data mining, human mobility observations are typically modeled as geo-spatial trajectories [13]. A number of analytics methods and corresponding applications have been proposed for trajectory data including hotspot detection [33], [34], trajectory mining [40], [42] and trajectory clustering [4], [5]. While the above work has wide implications for urban transportation, planning and resource management, advertisement and recommendations, to the best of our knowledge such models have not been previously employed to optimize WANET connectivity.

N, T	Number of agents, number of time points within a day
$L_i^t = \{l_i^t\}$	Location history of agent a_i, l_i^t is location at time t
B	Daily budget in i) number of connections or ii) % battery
b_i^t	Current available budget for agent i at time t
M	Agent's memory for past interactions, in #days
D_i^g, D_i^l	Agent's models for global and local (own) behavior
α	Confidence interval for expected reward comparison
$\mu_{k,t}, \Sigma_{k,t}$	Mean and covariance for trajectory k at time t
$sp_{k,t}$	Trajectory aggregate conditional probability
w_k	Trajectory weight
$P^D(l,t)$	Total probability density at location l and time t, by model D
β	Model persistence/decay coefficient

TABLE I: Notations used throughout the paper.

Reinforcement learning (RL) is a natural fit for our problem setting since devices are exposed to limited and incrementally acquired local information. Battery constraints and non-stationary also require a balance between prior (exploitation) and new observations (exploration)— a classical trade-off in RL. The model in $CORE$ has parallels to multi-armed bandits [35], [39], particularly their budgeted versions and more generally budgeted RL [6], [18]. However, the specific dependencies between arms (spatio-temporal decisions) prevent simultaneous choice of all arms and the shared budget across time introduces unique challenges for our setting. Existing spatio-temporal applications of RL focus on resource allocation including sensor placement [9], [38] and bike [19] or taxi sharing [15]. Multi-agent coordination in different non spatio-temporal settings has also been proposed for inference and coordinated channel selection [8], [32].

III. Problem Formulation

Before we formally define the problem we introduce some necessary notation summarized in Table I. We assume a set of N participating agents (devices), where each agent $a_i, i \in [1, N]$ records their spatio-temporal location $l_i^t = \{lat, lon\}$ at regular time points $t = 0 \ldots T$ throughout a day. Irregular/missing locations can be interpolated via standard techniques. Since smartphones run multiple applications, only a limited fraction of their battery can be dedicated for peer discovery without limiting their general usability. Hence, in our framework each device has a finite budget B for p2p connections across time points T. We first assume that B is measured in number of discovery attempts and in Sec. IV-D redefine it as realistic battery drain for Android devices establishing p2p Wi-Fi Direct connections.

When a device establishes a connection with a peer, it shares its location history and that of other devices it has interacted with for a predefined past interval of M days. Thus, an agent a_i has access to its own location history L_i^M and the partial location histories of a subset of other devices $\{L_j^M\}, j \neq i, j \in [1, N]$. Note that successful exchanges lead to more information about others' mobility and even more successful exchanges in turn. Based on the above input and definitions we are ready to define our connectivity optimization problem:

Definition 1: **Connectivity Optimization for WANETs:** *Given the location history L_i^M of agent a_i, the partial location histories of peers L_j^M, and a remaining budget b_i^t at time t, design a decision function $d(i,t)$ for a connection attempt at time t to maximize the daily successful connections.*

The above definition seeks to maximize a_i's connections, and since we assume a collaborative system of N symmetric agents, our goal is to maximize the total realized connections. Note that if the actual future locations for all agents during the day is known, the problem of scheduling when to connect can be trivially solved. In our setting of human mobility, however, the agent does not know i) its future locations and ii) that of peers, and hence we need to rely on predictive models for both. In addition, the daily mobility may not be stationary, and thus, our models need to adapt to changes in mobility.

IV. $CORE$: Reinforcement Learning Optimization

The goal of each agent is to maximize successful connection attempts while utilizing as much of the available budget as possible. Thus, our connection attempt decision function $d(i,t)$ should be able to weigh the costs (risk of spent budget for unsuccessful connections) and benefits (successful connections) of a decision at the current and future times within the day. Since future locations are unknown, the agent should be able to predict its own future locations and that of others to assess trade-offs at time t. Agents should also adapt to changes in the daily mobility patterns. To tackle these challenges, we propose a reinforcement learning approach, named $CORE$, for the connection decision problem. $CORE$ learns to i) forecast the mobility of the current agent and that of peers, and ii) optimally decide when and where to connect. In reinforcement learning terms, the reward is computed implicitly as connection success, states are derived from agents' time, location, and models, and actions are attempts to connect.

A. Modeling and predicting agent mobility

Recall that an agent a_i has full knowledge of its location history L_i^t (or trajectory) and partial knowledge of those of peers $L_j^t, j \leq N$. We partition observations into daily trajectories as a day is a natural resolution for recurrent mobility and a full battery recharge. To model daily mobility D, we employ a Gaussian Mixture (GM) to represent trajectories where each component in the mixture represents a cluster of observed histories. A GM model has a number of advantages: 1) it unifies the representation of global and agent-level behavior, 2) it represents spatio-temporal densities in temporally consistent manner due to trajectory-level weightings, and 3) it allows for efficient updates upon new observations.

Each cluster is a time series pair $(\mu_k, \Sigma_k), k = \{1...K\}$ of mean locations $\mu_k = \{\mu_{(k,1)}...\mu_{(k,T)}\}$ and spatial covariances $\Sigma_k = \{\Sigma_{(k,1)}...\Sigma_{(k,T)}\}$. We also maintain a per-cluster weighting vector $w \in [0,1]^K, \sum w = 1$ modeling the fraction of the population aligned to a cluster. This GM model can be viewed as a time series of Gaussian mixtures where the weight vector is conserved across time. Each agent maintains a GM model of its view of the global mobility D_i^g (we will omit the subscript when context allows), and also an agent-specific GM of their own trajectories D_i^l.

The local and global GM models are employed to predict densities at location l and time t interpreted as either a normalized agent count in the case of D^g or a probability of presence from the local model. This density is given by $P^D(l,t) = \sum_k w_k P^D(l,t|k)$, where D is D^l or D^g and $P(l,t|k)$ is the Gaussian density according to the mean and variance for cluster k at time t, $P(x|k) = (2\pi|\Sigma|)^{-0.5} \exp(0.5(x-\mu)'\Sigma^{-1}(x-\mu))$. A single expected location can also be calculated. Since these models represent trajectories through space-time, our prediction of agent locations can be precise. Specifically, an agent can infer its future location based on its history within the current day and the model from prior days. The best-match trajectory k is given as $\arg\max_k \prod_{\tau<t} P^{D^l}(l_\tau^i, \tau|k)$.

B. Mobility model updates.

Since we predict the mobility in an online manner, a key challenge is to update the models both upon arrival of new information as well as with changes of the underlying mobility patterns. To this end, we need to devise updates to cluster parameters based on new observations as well as procedures for merger/birth of clusters.

Recall that D model mobility as GM model trajectories which allow updates from single observations to incorporate new information on the fly. This process entails updates to the weight and distributional parameters of individual GM clusters according to the likelihood of new observation belonging to them. Updates of distributional parameters are equivalent to maintaining the mean and covariance of a weighted stream, for which we use a modified (for two dimensions and weights) version of Welford's algorithm [7], as follows:

$$\begin{cases} sp_{k,t} = sp_{k,t} + P(k|l_t) \\ \omega = \frac{P(k|l)}{sp_{k,t}} \\ \delta l = l - \mu_{k,t} \\ \mu_{k,t} = \mu_{k,t} + \omega \delta l \\ \Sigma_{k,t} = (1-\omega)\Sigma_{k,t} + \omega(\delta l)(l - \mu_{k,t})', \end{cases} \tag{1}$$

where $P(k|l) = \frac{P(l|k)w_k}{\sum_k P(l|k)w_k}$ via Bayes' theorem and $sp_{k,t}$ is the aggregate weight for each path cluster k as observed at time t. Cluster weights are then computed as $w_k = \frac{\sum_t sp_{k,t}}{\sum_{k,t} sp_{k,t}}$. Context-dependent explicit updates are described next:

1) **Updates upon a failed connection:** When an agent observes that there are no peers at the current location, "nearby" clusters are weighted down based on a single observation "elsewhere". Specifically, for clusters whose Mahalanobis distance $(l-\mu)'\Sigma^{-1}(l-\mu)$ exceeds a (γ_{-}) threshold on the χ^2_2 distribution [29], we update aggregate weights $sp_{k,t} = sp_{k,t} - P(k|l)$, and consequently w_k. The mean and covariance parameters are not updated based on this type of observation, as it simply boosts mass somewhere else as opposed to change a particular location. Any local aggregate weight is also bounded below by 0 to maintain interpretability and consistency. We set closeness parameter $\gamma_{-} = 0.05\%$ (other values performed similarly in experiments).

2) **Updates upon a successful connection:** Upon a connection, agents exchange unknown information as (agent, time, location) triples including the agent's location history and partial location histories of any contacted individuals within a memory window of M past days. Each of these new point observations is incorporated into D^g as described in Eq.1. New information is maintained and exchanged further. In addition to point-level data, information about a peer's private model is incorporated by adding clusters contained therein to the receiving agent's global model. This enriches the global cluster model with additional possible trajectories, if those differ from known behaviors. This exchange is done prior to incorporation of point data to make sure that information corresponding to a "new" cluster is properly incorporated.

There are two update types: i) new cluster creation when a new point does not fit into existing clusters and ii) cluster merging when a new cluster aligns well with an existing one.

2.1) New cluster creation. Changes in mobility or novel observations may not fit existing clusters (we quantify goodness of fit by the threshold γ_{-} defined above). In this case we create a new "constant" trajectory representing a stationary position, i.e., $\mu_{k,t}$ is the current location for all t and a default Σ (set to the identity in synthetic data).

2.2) Cluster merging. To maintain a sparse and informative model, clusters may need to be combined if i) they are sufficiently close or ii) the overall model grows too large, forcing merger of "closest" clusters (we set the maximum cluster count to 20). Cluster proximity is measured via time-aggregated Bhattacharyya distance $D_B = \sum_t \frac{1}{8}(\mu_{1,t}-\mu_{2,t})'\Sigma_t^{-1}(\mu_{1,t}-\mu_{2,t}) - \frac{1}{2}\ln(\frac{\det\Sigma_t}{\det\Sigma_{1,t}\det\Sigma_{2,t}})$, with $\Sigma_t = 0.5(\Sigma_{1,t}+\Sigma_{2,t})$. When D_B is below a predetermined threshold (or when the number of existing clusters exceeds the prescribed maximum), pairs of close clusters are merged. Upon merger, we create a new Gaussian cluster with the following parameters:

$$\begin{cases} \mu^* = & w_1\mu_1 + w_2\mu_2 \\ \Sigma^* = & w_1\Sigma_1 + w_2\Sigma_2 + w_1\mu_1'\mu_1 + w_2\mu_2'\mu_2 - \mu^{*\prime}\mu^*, \end{cases} \tag{2}$$

where w_1 and w_2 are the weights of the two clusters normalized by their sum, $w_1 + w_2 = 1$. Other quantities, including aggregate observed weights sp, are also combined.

3) **End-of-day updates for D^l.** The newly observed agent's daily trajectory x_i is employed to update the local mobility model D^{l_i}, similar to the single-observation updates: for each $x_{i,t}$ we update $\mu_{k,t}$ and $\Sigma_{k,t}$ of all clusters. If x_i is a poor fit to D^{l_i} (based on sum of the Mahalanobis distance compared to the appropriate χ^2_{2*T} statistic and closeness γ_{-}) the agent creates a new cluster $\mu_{K+1} = x_i$ with a default Σ_{K+1}.

Algorithm 1 *CORE* (Connection decision at time t)

```
Require: Time $t$, location $l_t$, budget $b_t$, local $D^l$ and global $D^g$ models,
Require: Exploration controls ($CORE$-Now, $CORE$-later) and rate $\epsilon$
 1: Compute prob. of future locations $P^{D^l}(l,\tau), \forall\tau \in [t+1,T], \forall l$
 2: for $\tau = 1 \ldots (T-t)$ do
 3:     $R_f[\tau] \leftarrow \sum_l P^{D^l}(l,t+\tau)D^g(l,t+\tau) + CI(\alpha) * \sigma(t+\tau)$
 4: end for
 5: Sort $R_f$ in descending order
 6: $R_t \leftarrow D^g(l_t,t)$
 7: DECISION $\leftarrow R_t >= R_f[b_t]$
 8: if $CORE$-Now and $rand < \epsilon$ then DECISION $\leftarrow True$
 9: if $CORE$-Later and $rand < \epsilon$ then DECISION $\leftarrow False$
10: if DECISION then
11:     Attempt Connection
12:     if Successful Connection then
13:         Absorb $D^l$ from peer into $D^g$
14:         Merge clusters in $D^g$ as necessary (IV.B.2.2)
15:         Update $D^g$ with new data from peer (IV.B.2)
16:     else
17:         Do a "failed" connection update for $l_t$ (IV.B.1)
18:     end if
19: end if
20: if $t \equiv T$ then
21:     End-of-day update of $D^l$ (IV.B.3)
22:     Decay $D^g$ - multiply all $sp_{k,t}$ by $\beta$
23: end if
```

C. Reinforcement learning (RL) for connection decisions

Non-stationarity, incomplete observations about peers and limited battery on smartphones all present challenges for employing traditional predictive methods. In addition, successful adaptation to changes hinges on collecting sufficient observations after behavioral shifts and requires connection attempts at low-likelihood (w.r.t. the current model) spatio-temporal locations. This trade-off between *exploration* and target optimization *exploitation*, is the purview of Reinforcement Learning (RL) algorithms and hence we model our agents' connection decision function as an RL process.

While the long-term goal of agents is to optimize total connections, the immediate *reward* for a decision is the success of the connection attempt. We quantify the expected reward via the probability of encountering other agents $P^{D^g}(l,t)$. An agent's *action* (the decision to connect) impacts the available budget and the total reward, as it reduces the budget for possible future connections. An effective decision function must balance this opportunity cost with current reward. Our decision is *contextual*, where context is inferred from the agent's mobility models and current location.

1) The *CORE* Algorithm. We outline the steps of $CORE$ in Alg. 1. At a given time t an agent makes a decision on whether to attempt a connection (Steps 1-9) and updates model parameters based on the outcome (Steps 10-20). The key idea behind making a decision is to compare the expected reward for an immediate connection R_t with those in future time steps R_f. Given a remaining budget of b_t connections, an attempt

Function	Parameters	Description
$CORE$	α	Confidence bound on R_f (Alg. 1, Step 3)
$CORE$-Now	α, ϵ	Early random exploration (Alg. 1, Step 8)
$CORE$-Later	α, ϵ	Early random wait (Alg. 1, Step 9)

TABLE II: Summary of alternative decision functions.

at time t is reward-optimal if R_t exceeds the expected reward of the b_t-th best future time point $R_f[b_t]$ (Step 7).

Estimating the reward R_f for future connection requires both estimates of the future spatio-temporal density of peers and the agent's own future locations (Step 3). Since future location predictions through D are uncertain, we employ a confidence interval estimate to account for anticipated variance in R_f. Specifically, we compute P^{D^l}-weighted mean and variance of D^g at t; $CI(\alpha)$ is the corresponding value from the standard normal distribution, indicating the confidence interval bound. Different choices of α or exploration strategies (Steps 8, 9) yield different families of decision functions.

The result of a connection attempt informs the agent's mobility model updates based on both the new observation about the presence/absence of peers at (l_t, t) as well as additional observations collected by the peer in previous exchanges (Steps 12-18). An important consideration in RL models is the learning rate for updating the reward function based on new data. For $CORE$, this is controlled by two memory parameters: (i) agent memory M which specifies the number of days to keep past observations from exchanges and (ii) the model persistence controlled by a decay parameter β, which multiplies cluster $sp_{k,t}$ weights at the end of the day (step 22), preserving cluster relative weights w_k. This parameter is inversely related to the traditional learning rate - a lower β means less weight on an existing model and therefore more on newly arriving information.

Complexity discussion: Model updates are the costliest steps in CORE. Since we store the inverse covariance matrix, point updates require exponentiation of a small matrix-vector product. Cluster mergers come at a higher cost which can be controlled via various parameters to limit energy usage. In terms of memory, storing all observations from a fixed past window requires at most $O(MTN)$ storage in an ideal scenario with all potential connections realized. The local and global models require $O(KT)$ storage for storing means, variances, and time-specific weights for a maximum of K clusters. Choosing a low (potentially heterogeneous) spatio-temporal resolution while maintaining high connectivity success can enable significant reductions of the required storage.

2) Decision Functions. The exploration switches ($CORE$-Now,$CORE$-Later), exploration rate ϵ and the future confidence interval $\alpha = 0.5$ allow for the configuration of several families of decision functions summarized in Tbl. II. Setting $\alpha = 0.5$ eliminates prediction uncertainty and effectively results in a *greedy* RL approach, while lower or higher α allow for lower/higher confidence in future predictions. To configure an ϵ-greedy RL approach, we allow exploration by taking an action regardless of state (Steps 8,9) with a rate of ϵ. We can prioritize early attempts (connect *Now*) or later attempts (connect *Later*) or inform when to explore based on statistics from past observations. Different values of α can be combined with ϵ for $CORE$-Now or $CORE$-Later.

Event	Power (mW)	STD	ΔT (h)	STD	% Battery	STD
Connect	1.1330e+3	58.01	0.0022	1.842e-4	0.0215	0.0011
Send	2.3406e+3	104.95	2.056e-5	3.020e-7	8.30e-10	4.02e-11

TABLE III: Power draw, duration and percentage battery drain per byte to *connect* and *send* data. Energy for receiving data is negligible (omitted).

D. Realistic budget based on battery drain

In deriving $CORE$ we made a simplifying assumption that the budget B is specified in terms of number of connection attempts and that all connection attempts and information exchanges have a fixed battery cost. This assumption might not hold for real-world exchanges. In addition battery expenditure (as %) would be a more user-friendly parameter on devices running $CORE$ as opposed to connection attempts. Hence, we next discuss how to augment our protocol to allow for measurement-based realistic cost in terms of battery drain.

We first measure the connection and exchange costs as %-age of battery drain for Android devices establishing a connection over Wi-Fi Direct. Then we discuss the necessary changes to our protocol to specify % battery as budget. Note that our choice of Android devices using Wi-Fi Direct is for demonstration and evaluation purposes (see §V Fig. 3). In real deployments, one can estimate device-specific costs and consider alternative (or even multiple) radio connectivity such as Bluetooth and multi-peer connectivity for Apple devices.

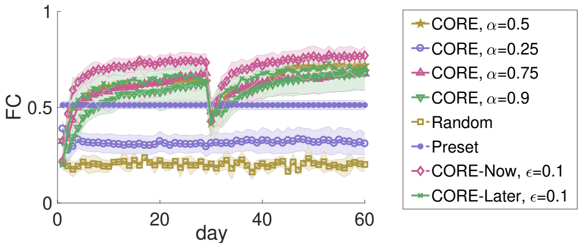
To estimate connection and exchange costs, we use one Motorola G(6) and one Samsung Galaxy 5 Duos smartphone running a simple Wi-Fi Direct app. The Samsung is connected to a Monsoon power monitor, bypassing the battery, so we can measure the energy consumption of the device. We measure the required power to i) *Connect* and ii) *Send* data to a peer. Tbl. III presents average and standard deviation over 10 runs for power draw, duration and % battery discharge per byte, which we use in our experimentation.

Modifying $CORE$ to admit variable connection costs consists of two components. First, we must compute actual exchange cost and decrease the budget by exact non-integer amounts, instead of simply in units, after any successful connection. Second, we need to adjust the anticipated number of remaining connections at any decision point by dividing the available power budget by an expected cost. This expected cost can be estimated using the same streaming mean computation as the trajectory cluster means described in the previous section. Each agent stores this expected connection cost C_{exp} locally, and updates it with the computed cost of every connection attempt. We can then modify the total daily budget B to be a battery percentage, and the remaining number of connections becomes $b_t = B/C_{exp}$.

V. Experimental Evaluation

A. Experimental setup

1) Synthetic mobility traces. To evaluate the impact of modeling decisions in $CORE$, we generate synthetic data



(a) FC over time

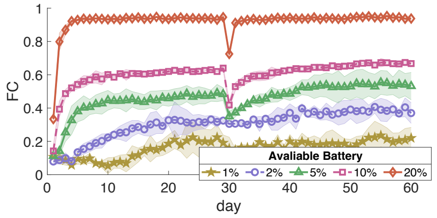
		$CORE$, (α)				$CORE$-Now	$CORE$-Later	Random	Preset
Parameters:		0.5	0.25	0.75	0.9	$\epsilon = 0.1, \alpha = 0.5$		-	-
T_{90}	Pre	7	1	12	12	6	7	1	1
	Post	12	1	7	11	6	6	2	1
FC_m	Pre	0.64	0.32	0.67	0.62	0.74	0.66	0.21	0.51
	Post	0.72	0.33	0.68	0.68	0.77	0.71	0.21	0.51

(b) Metrics before (Pre) and after (Post) the shift at day 30

Fig. 2: (a) Comparison of $CORE$'s quality (FC) with different decision functions and Random and Preset baselines ($B = 10$ for all methods).

from trajectory model with controllable agent co-location over time. Agent daily trajectories are sampled from probabilistic GM trajectories models similar to those we employ in $CORE$. We generate $k = 3$ global GM clusters with independent, manually-generated (μ_k, Σ_k), representing large-scale mobility patterns within a 50×50 grid. The 3 global proto-trajectories are designed to represent typical global daily behavior: (1) an early-day static location with large variance ($\approx 7I$) representing multiple "home" locations, (2) a mid-day location with small ($\approx I$) variance ("work place"), and (3) a return to home via a set of locations with mid-to-large variance. Global trajectories are designed to overlap at various points during the day. Each agent randomly draws two of the global clusters. A sample daily trajectory is then generated by drawing a sample x from one of the agent's clusters and adding user-specific variance controlled by a parameter σ_{self}, which we vary in Fig. 5(a). Overlap between agents can come from either a shared global cluster or from intersections between global paths. Unless otherwise specified, data is "hourly" (i.e. $T = 24$), we simulate 30 agents over a 100×100 grid, and $\sigma_{self} = 0$. To evaluate cold start learning and adaptation to changes, we also introduce a shift in the underlying mobility at day 30 where the "work place" location in two of the global clusters changes along with the timing of the simulated workday in those clusters. This affects the mobility patterns of agents which "subscribe" to these global clusters.

2) Real-world mobility traces. We employ the Yonsei Lifemap trace [10] from CRAWDAD and a trace from UberMedia, (ubermedia.com). Yonsei contains observed locations for nine users from Yonsei University in Korea with shared observation period of 63 days. To create uniform daily sampled location, we interpolate hourly locations based on the closest available user locations. We remove days in which users do not collocate. To enable a longer learning window, we randomly re-sample and "replay" full daily histories for all users, creating a 250-day evaluation trace. The UberMedia data contains daily mobility of phones visiting a downtown area in the US with state public offices. We select the subset of 50 devices with the most location updates spanning February through April of 2020. Locations are averaged over all available reads within an hour, and aligned to a $600m$ grid. Locations at hours with no observations are interpolated as in Yonsei.



	T_{90}		FC_m	
	Pre	Post	Pre	Post
1%	21	10	.21	.22
2%	17	10	.32	.39
5%	10	11	.48	.54
10%	7	7	.63	.67
20%	3	1	.94	.95

Fig. 3: (left) Performance of $CORE$ ($\alpha = 0.5$) over variable battery budget % and (right) corresponding metrics. Daily time points $T = 144$

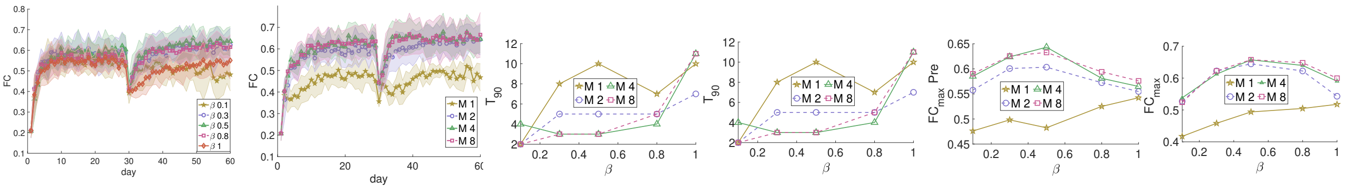
3) Evaluation metrics. Our primary metric is the *Fraction of ideal Connections (FC)* per agent, defined as the fraction of successful from all possible connections, based on actual co-locations of peers capped by the available budget. This metric is the larger of precision (true positives over all selections) and recall (selected positives over all positives) due to the budget bound on the denominator. The quantity $1 - FC$ is a daily measure of the regret in RL, i.e. the difference from an optimal agent. The maximum attained value of FC is denoted by FC$_m$. Due to the variability of this statistic, we present FC$_m$ as the maximum over five-day moving averages.

We can describe the *learning rate* in terms of FC's evolution over time. The metric T_{pct} is the number of days needed to achieve a fixed percentage *pct* of FC$_m$. To quantify $CORE$'s ability to learn from a "cold start" and after a shift in mobility patterns (generated at day 30 of our traces), we notate T_{pct} and FC_m before the shift as *Pre*-shift and after it as *Post*-shift.

B. Effect of the decision function, memory and budget

We first evaluate $CORE$ for varying key parameters. Unless stated otherwise, we set $\beta = 0.8$, $M = 2$ and $B = 10$ connections. Presented results are averages of five runs.

1) Effect of the decision function. Fig. 2 compares the performance of $CORE$'s variants employing decision functions from Sec. IV-C.2, and two baselines: *Random* which attempts $B = 10$ connections at random times; and *Preset* which attempts connections every two hours (starting hour 4 of the day). Quality differences among variants of $CORE$ are subtle but informative. In the basic $CORE$ decision function (no exploration), confidence bound of $\alpha = 75\%$ seems to be optimal compared to a simpler mean-comparison approach ($\alpha = 0.5$) or lower confidence ($\alpha = 0.25$) which performs close to Random. Increasing the confidence to $\alpha = 0.9$ is too "conservative" and leads to slow learning. Introducing exploration as opposed to greedy decisions improves performance noticeably. In particular, early exploration by $CORE$-Now outperforms all alternatives, while delayed exploration by $CORE$-Later performs on par with the best basic approach $CORE$, $\alpha = 0.5$. The Random and Preset baselines perform significantly worse than the best variants of $CORE$.



(a) Effect of β (b) Effect of M (c) T_{90} Pre (d) T_{90} Post (e) FC_m Pre (f) FC_m Post

Fig. 4: Effect of model decay β (a) and memory M (b) on the successful connections. Panels (c) and (d) present effects of both parameters on the learning rates (T_{90}) at the beginning of the experiment (Pre) and after a mid-stream behavioral shift (Post). Panels (e) and (f) show the effects of both parameters on maximal achieved connectivity FC$_m$ under Pre and Post regimes.

Notably $CORE$-Now reaches 90% of its peak FC_m in as few as 6 days both before (Pre) and (After) the shift in mobility at day 30. (Cols T_{90} in Fig. 2). Its performance peaks at over 74% (Pre) and 77% (Post) of the maximal possible connections given sufficient time to learn the underlying mobility model. This performance is aided by $CORE$-Now's stochastic exploration which unlike a fixed method does not get stuck into a sub-optimal decision regime when information about alternatives is limited. $CORE$-Later which has an implicit preference for attempts later in the day when initial attempts fail is also slow to learn after change points. Random and Preset's quality in comparison is limited as they do not model and learn the underlying mobility.

2) Connection budget. Another important question is: What is a sufficient battery budget for attaining high levels of FC? A greater budget allows for more information gain via exchanges among agents and our goal in the next experiment is to quantify the relationship between battery percentage dedicated to peer connections and the attained FC. We estimate the actual battery drain for each exchange (both connection and data transfer) using our Android-based cost model from Sec. IV-D. Fig. 3 demonstrates that higher budget enables not only higher connectivity, but also a greater learning rate, i.e. faster convergence to FC_m. In particular a budget of 20% leads to success in almost all connection opportunities, while 10% budget peaks at $FC_m = 0.67$. In settings of limited co-location of agents (fewer opportunities to connect) it makes sense to increase the budget initially to enable some exchanges to happen and bootstrap learning. Additionally, higher budgets allow for consistent performance while lower ones results in larger variance. $CORE$ is able to quickly recover from the mid-point shift at day thirty across battery budgets, but more so with higher budget allocations.

3) Memory and model decay. An important aspect of $CORE$ is its ability to adjust to evolving mobility, which is controlled by the amount of observations used for training (i.e. memory M) and the amount of "inertia" or weight given to a trained model (i.e. decay β). Fig. 4 shows the impact of these parameters, with an emphasis on the shift in agent mobility at day 30. While cold-start performance is similar for all β in Fig. 4(a) due to a lack of model to persist, the inertia of an outdated model with mobility change (day 30) shows that higher β takes longer to recover. Too low β means minimal learning is retained, which deteriorates the overall performance. Thus, β selection should balance the tradeoff between insufficient learning (low β) and outdated models (high β). Larger memory M is consistently beneficial according to Fig. 4(b). A large memory is also beneficial for learning based on both T_{90} (Figs. 4(c), 4(d)) and FC$_m$ (Figs. 4(e), 4(f)). These figures also demonstrate the effects of β, which serves to improve learning rate and particularly the recovery from a behavioral shift (Fig. 4(d)) via faster clearing of outdated models. FC$_m$ shows that an optimum β exists regardless of M, as in previous figures - greater model retention is only helpful up to a point, after which learning is inhibited.

C. Effect of the underlying mobility and population size

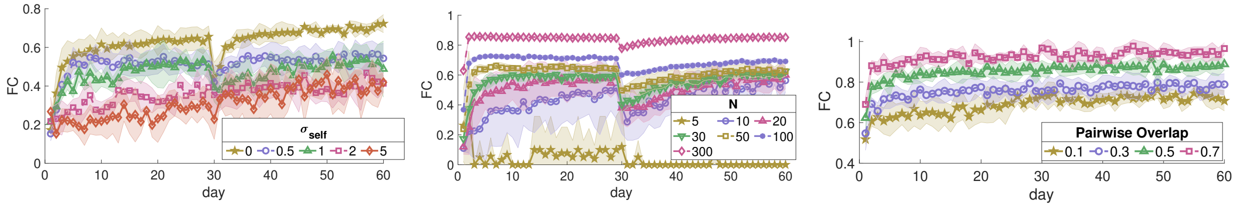
We next evaluate $CORE$'s performance for varying agent mobility dynamics and population size. Intuitively, more stable individual trajectories mean more reliable information from exchanges. Fig. 5(a), shows that increasing the variance of single-agent behavior from 0 to 5 leads to paths that are harder to learn and take longer to reach a high FC. The data shift at day 30 in our synthetic examples does not have an observable effect for larger variances, as the lower consistency of mobility leads is comparable to changes incurred by the shift.

In Fig. 5(b) we consider the impact of the number of agents N on the performance of $CORE$. With higher N the availability of partners, and thus, "mixing", increases. Additionally, as demonstrated before, more data availability leads to faster learning, which results in lower T_{90} for larger agent pools. Learning occurs among as few as 10 agents.

The global trajectories' interactions play a role in agent effectiveness. More intersection between clusters means more opportunity for agents from different clusters to encounter each other, leading to more learning. Fig. 5(c) explores this by varying the overlap of spatiotemporal locations, between pairs of global trajectories. The trajectories for this plot are randomly generated and are therefore more chaotic. Less patterned behavior leads to higher difficulty in learning, however an upward trend is still visible (in fact, $T_{90} = 1$ for all settings here). Variability in connectivity with global trajectory overlap indicates that agents do not connect only with those on the same global trajectory (the number of those is, on average, fixed across settings).

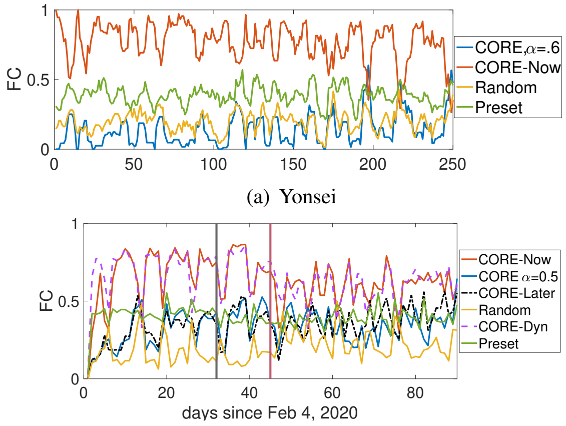
D. Performance on real-world traces

We first evaluate $CORE$'s behavior on the Yonsei mobility trace (Fig. 6(a)). As we observed in synthetic traces, effective learning hinges on mobility consistency and large populations



(a) Personal variance σ_{self} (b) Population (c) Overlap

Fig. 5: Effect of (a) variability in personal trajectory, (b) number of agents, and (c) spatio-temporal intersection of global trajectories.



(a) Yonsei

(b) UberMedia, Feb-Mar, 2020

Fig. 6: Performance comparison on real-world traces from (a) Yonsei and (b) UberMedia. B=10, T=24 for both datasets. In (b), a black line at Mar 7 and red at Mar 20 mark pandemic events.

for mixing. The Yonsei trace with only 9 users and large variance in individual mobility presents a particularly challenging case even after re-sampling of full days. Nevertheless, $CORE$ exhibits promising behavior in this data. Fig. 6 compares the performance of two variants of $CORE$ and two baselines with an available budget of $B = 10$ connections (out of $T = 24$ time points). An immediate takeaway is the importance of coordination: *Random* connection decisions perform poorly, even compared to attempts that are timed arbitrarily but synchronously across agents (*Preset*).

Consistent with observations on synthetic traces, $CORE$-Now with $\epsilon = 0.1$ performs better among the two variants of $CORE$. Random early exploration allows for faster recovery upon changes in mobility as accumulated information from peers allows all agents to target advantageous times. Short-term learning and recovery is visible even with consistently ”near-perfect” performance: after dips in FC due to mobility changes, $CORE$-Now recovers to nearly FC=1. The worst performer on Yonsei is the basic decision function: $CORE$, $\alpha = 0.6$. By default, $CORE$'s decision function is conservative with its agent budget. Limited early connections, combined with high variance of individual agents renders basic $CORE$'s conservative lack of exploration disadvantageous.

Results on UberMedia are presented in Fig. 6(b). For reasons similar to those presented in Yonsei, random and preset connection schemes underperform. Likewise, performance for $CORE$, α=0.5 is weak, due to its conservative strategy coupled with a lack of explicit exploration leading to missed connection opportunities. Once again we observe a good performance from the early exploration $CORE$-Now, with clear learning trends over the period as agents accumulate observations. $CORE$-Later performs worse, $CORE$-Now, but still exhibits a clear upward trend over the period of analysis.

To further investigate the impact of exploration in a challenging setting, we introduce an additional learning approach: $CORE$-Dyn. This method is similar to $CORE$-Now, however ϵ varies depending on the observation history as $\epsilon_t = \min(0.4, s_t/f_t * 0.4)$, where s_t and f_t are successful and failed connection attempts at time t, respectively, from previous days. In this way, the agent further “explores” promising *time* steps while avoiding times of past failed connections, independent of the location. $CORE$-Dyn learns faster than $CORE$-Now initially and their performance equalizes after a period of time.

These data straddle the beginning of the COVID-19 pandemic. In NY, a state of emergency was declared on March 7 and a stay-at-home order on March 22. $CORE$-Now and $CORE$-Dyn experience a mild decline in performance, yet all $CORE$ approaches show stabilization and beginnings of adaptation to the new regime by the end of the analysis period.

Further refinement of the decision function as well as modeling parameters are likely to allow for further improvement, though a detailed examination on this particular dataset is outside the scope of this work. As a simple direction, we can modify the randomness of the $CORE$-Now approach over time, reducing the ϵ parameter to target exploitation of learned patterns after early exploration has yielded meaningful information (i.e. time-evolving exploration). Another possible modification is inspired by a visible periodicity of the performance in Fig. 6(b): the dips in performance every 7 days correspond to weekends when agents are unlikely to visit their office locations. Modifying the predictive model to explicitly account for different regimes (possibly based on attempts early in the day) may yield better performance on these off-days.

VI. Conclusion

We introduced an adaptive and distributed protocol, named $CORE$, with the objective to maximize peer-to-peer connectivity in WANETs based on reinforcement learning. Our protocol addresses an important challenge in ad hoc networks of smaprtphones, namely, a predictive and energy-aware data link layer based on modeling human mobility using partial knowledge. $CORE$ models individual and global mobility based on partial historical location observations and incorporates predictions of these models into a reinforcement learning framework capable of on-the-fly decision making regarding when and where to scan for peers. We evaluated our model in both simulated and real-world mobility traces and demonstrated that agents are able to materialize 95% of the maximum possible connection opportunities using at most 20% of the phone battery for discovery and exchange. $CORE$ can serve as a foundation for a plethora of distributed applications such as disaster response, activist coordination and information access in disconnected areas.

The $CORE$ approaches presented in this work yield promising results but they also present a variety of natural extensions for future work. In particular, we can consider varying the decision function to more intelligently capitalize on the need to explore in variable settings. This can be a $CORE$-Now implementation with ϵ that varies based on time, potentially through tracking the number of (successful) attempts at that point. Additionally, there may be value in adjusting the reward function altogether to account for different potential connection targets. For the information-dissemination application described herein, it is beneficial to connect to as wide a variety of other agents as possible; encoding this concept of variety in the reward is likely to yield a more effective system in that regard.

VII. Acknowledgements

This work is supported by NSF S&CC grant CMMI-1831547 and NSF CAREER grant CNS-1845858.

References

[1] A. Al-Akkad, C. Raffelsberger, A. Boden, L. Ramirez, A. Zimmermann, and S. Augustin. Tweeting'when online is off'? opportunistically creating mobile ad-hoc networks in response to disrupted infrastructure. In *IEEE ISCRAM*, 2014.

[2] A. Al-Akkad, L. Ramirez, A. Boden, D. Randall, and A. Zimmermann. Help beacons: Design and evaluation of an ad-hoc lightweight sos system for smartphones. In *ACM SIGCHI Conference*, 2014.

[3] F. Álvarez, L. Almon, P. Lieser, T. Meuser, Y. Dylla, B. Richerzhagen, M. Hollick, and R. Steinmetz. Conducting a large-scale field test of a smartphone-based communication network for emergency response. In *ACM CHANTS'18*, 2018.

[4] J. Bian, D. Tian, Y. Tang, and D. Tao. A survey on trajectory clustering analysis. *CoRR*, abs/1802.06971, 2018.

[5] J. Bian, D. Tian, Y. Tang, and D. Tao. A survey on trajectory clustering analysis. *arXiv preprint arXiv:1802.06971*, 2018.

[6] N. Carrara, E. Leurent, R. Laroche, T. Urvoy, O.-A. Maillard, and O. Pietquin. Budgeted reinforcement learning in continuous state space. In *Advances in Neural Information Processing Systems*, 2019.

[7] T. F. Chan, G. H. Golub, and R. J. Leveque. Algorithms for computing the sample variance: Analysis and recommendations. *The American Statistician*, 37(3):242–247, 1983.

[8] H.-H. Chang, H. Song, Y. Yi, J. Zhang, H. He, and L. Liu. Distributive dynamic spectrum access through deep reinforcement learning: A reservoir computing-based approach. *IEEE IoT Journal*, 2018.

[9] L. Chen, J. Xu, S. Ren, and P. Zhou. Spatio–temporal edge service placement: A bandit learning approach. *IEEE Transactions on Wireless Communications*, 17(12):8388–8401, 2018.

[10] Y. Chon, E. Talipov, H. Shin, and H. Cha. CRAWDAD dataset yonsei/lifemap (v. 2012-01-03). Downloaded from https://crawdad.org/yonsei/lifemap/20120103/mobility, Jan. 2012. traceset: mobility.

[11] N. Ding, D. Wagner, X. Chen, A. Pathak, Y. C. Hu, and A. Rice. Characterizing and modeling the impact of wireless signal strength on smartphone battery drain. *In Proc. of ACM SIGMETRICS Review*, 2013.

[12] Ó. R. Helgason, E. A. Yavuz, S. T. Kouyoumdjieva, L. Pajevic, and G. Karlsson. A mobile peer-to-peer system for opportunistic content-centric networking. In *ACM SIGCOMM Workshops*, 2010.

[13] S. Hoteit, S. Secci, S. Sobolevsky, C. Ratti, and G. Pujolle. Estimating human trajectories and hotspots through mobile phone data. *Computer Networks*, 64:296–307, 2014.

[14] A. Ippisch, S. Sati, and K. Graffi. Optimal replication based on optimal path hops for opportunistic networks. In *IEEE AINA*, 2018.

[15] I. Jindal, Z. T. Qin, X. Chen, M. Nokleby, and J. Ye. Optimizing taxi carpool policies via reinforcement learning and spatio-temporal mining. In *IEEE BigData2018*, 2018.

[16] R. Kahn. The organization of computer resources into a packet radio network. *IEEE Transactions on communications*, 25(1):169–178, 1977.

[17] J.-K. Lee, K.-M. Lee, and J. Lim. Distributed dynamic slot assignment scheme for fast broadcast transmission in tactical ad hoc networks. In *In Proc. IEEE MILCOM*, 2012.

[18] A. Léon and L. Denoyer. Options discovery with budgeted reinforcement learning. *arXiv preprint arXiv:1611.06824*, 2016.

[19] Y. Li, Y. Zheng, and Q. Yang. Dynamic bike reposition: A spatio-temporal reinforcement learning approach. In *In Proc. SIGKDD*, 2018.

[20] Y. Liu, A. E. Bashar, F. Li, Y. Wang, and K. Liu. Multi-copy data dissemination with probabilistic delay constraint in mobile opportunistic device-to-device networks. In *IEEE WoWMoM*. IEEE, 2016.

[21] Y. Liu, H. Wu, Y. Xia, Y. Wang, F. Li, and P. Yang. Optimal online data dissemination for resource constrained mobile opportunistic networks. *IEEE Transactions on Vehicular Technology*, 66(6):5301–5315, 2016.

[22] H. Lu, J. Li, Z. Dong, and Y. Ji. CRDMAC: an effective circular rtr directional mac protocol for wireless ad hoc networks. In *IEEE MSN'11*.

[23] X. Lu, E. Wetter, N. Bharti, A. J. Tatem, and L. Bengtsson. Approaching the limit of predictability in human mobility. *Sci. reports*, 3:2923, 2013.

[24] C. Maitland and R. Bharania. Balancing security and other requirements in hastily formed networks: The case of the syrian refugee response. *Available at SSRN 2944147*, 2017.

[25] D. J. Malan, T. Fulford-Jones, M. Welsh, and S. Moulton. Codeblue: An ad hoc sensor network infrastructure for emergency medical care. In *IEEE BSN2004*, 2004.

[26] M. Mongiovi, A. K. Singh, X. Yan, B. Zong, and K. Psounis. Efficient multicasting for delay tolerant networks using graph indexing. In *IEEE INFOCOM*, 2012.

[27] B. Mutsvairo and S. T. G. Harris. Rethinking mobile media tactics in protests: A comparative case study of hong kong and malawi. In *Mobile Media, Political Participation, and Civic Activism in Asia*. 2016.

[28] E. S. Nadimi, R. N. Jørgensen, V. Blanes-Vidal, and S. Christensen. Monitoring and classifying animal behavior using zigbee-based mobile ad hoc wireless sensor networks and artificial neural networks. *Computers and Electronics in Agriculture*, 82:44–54, 2012.

[29] R. C. Pinto and P. M. Engel. A fast incremental gaussian mixture model. *PloS one*, 10(10):e0139931–e0139931, 2015.

[30] C. Shi, V. Lakafosis, M. H. Ammar, and E. W. Zegura. Serendipity: Enabling remote computing among intermittently connected mobile devices. In *ACM MobiHoc'12*, 2012.

[31] C. Song, Z. Qu, N. Blumm, and A.-L. Barabási. Limits of predictability in human mobility. *Science*, 327(5968):1018–1021, 2010.

[32] L. Stabellini and J. Zander. Interference aware self-organization for wireless sensor networks: A reinforcement learning approach. CASE'08.

[33] X. Tang, E. Eftelioglu, D. Oliver, and S. Shekhar. Significant linear hotspot discovery. *IEEE Transactions on Big Data*, 3(2):140–153, 2017.

[34] X. Tang, J. Gupta, and S. Shekhar. Linear hotspot discovery on all simple paths: A summary of results. In *ACM SIGSPATIAL*, New York, NY, USA, 2019. Association for Computing Machinery.

[35] L. Tran-Thanh, A. Chapman, A. Rogers, and N. R. Jennings. Knapsack based optimal policies for budget-limited multi-armed bandits. *arXiv preprint arXiv:1204.1909*, 2012.

[36] Q. Wang and J. E. Taylor. Patterns and limitations of urban human mobility resilience under the influence of multiple types of natural disaster. *PLoS one*, 11(1):e0147299, 2016.

[37] S. Wang, M. Liu, X. Cheng, Z. Li, J. Huang, and B. Chen. Opportunistic routing in intermittently connected mobile p2p networks. *In Proc. of IEEE Journal*, 2013.

[38] Z. Wang, H.-X. Li, and C. Chen. Reinforcement learning-based optimal sensor placement for spatiotemporal modeling. *IEEE transactions on cybernetics*, 50(6):2861–2871, 2019.

[39] Y. Xia, T. Qin, W. Ma, N. Yu, and T.-Y. Liu. Budgeted multi-armed bandits with multiple plays. In *IJCAI*, pages 2210–2216, 2016.

[40] Y. Yang, J. Cai, H. Yang, J. Zhang, and X. Zhao. Tad: A trajectory clustering algorithm based on spatial-temporal density analysis. *Expert Systems with Applications*, 139:112846, 2020.

[41] M. Zheleva, A. Paul, D. L. Johnson, and E. Belding. Kwiizya: Local cellular network services in remote areas. In *ACM MobiSys13*, 2013.

[42] Y. Zheng. Trajectory data mining: An overview. *ACM Trans. Intell. Syst. Technol.*, 6(3), May 2015.