Opportunistic Peer-to-Peer Mobile Cloud Computing at the Tactical Edge

Wei Gao
Department of Electrical Engineering and Computer Science
The University of Tennessee, Knoxville

***Abstract*—Mobile computing capabilities of the hand-held mobile devices are important for warfighters at the tactical edge to efficiently process their in-situ sensory data about the surrounding environment, so as to maintain situational awareness. Mobile cloud computing is generally applied to augment such capabilities of mobile devices, but has been limited to the interaction between the back-end cloud infrastructure, which availability, however, could be limited due to the dynamic network characteristics at the tactical edge. In this paper, we develop a novel theoretical framework to exploit the potential of peer mobile devices on mobile cloud computing, when they opportunistically move into the communication range of each other. Our proposed framework aims to adaptively configure the computational workload allocations among mobile devices, by considering both the energy consumption and timeliness of computational task execution. Probabilistic methods are adopted to quantitatively estimate the opportunistic network transmission delay and ensure that the computational results could be delivered back to the task initiator on time. Extensive trace-driven simulation results show that our proposed framework could significantly improve the mobile cloud computing performance, while reducing the energy consumption of such computing, compared to the existing schemes.**

I. Introduction

Hand-held mobile computing devices are nowadays crucial for warfighters in the practical Disconnected, Intermittent, and Limited-bandwidth (DIL) environments at the tactical edge [30], [27] to maintain situational awareness to their surrounding environments, by processing a large variety of contextual sensory data at real-time. For example, the background gunfire sound could be processed at real-time for sniper detection [33], and a photo of a susceptible object needs to be analyzed immediately to exclude possible hazard of Improvised Explosive Devices (IEDs). Such computing workload that could be computationally intensive, in some cases, may exceed the local computational capabilities of the warfighters' handheld mobile devices, and hence experience degraded computing performance on excessive response delay. These performance degradation and delay increase would seriously hinder the warfighters' rapid situational response to sudden events and adaptive operations at the tactical edge.

A straightforward solution to the aforementioned challenge is Mobile Cloud Computing (MCC), which offloads the computation workloads from local mobile devices to the remote cloud, and is supported by various techniques such as code migration and [6], [21] and Virtual Machine (VM) synthesis [3], [14]. Such workload offloading, however, is challenging to be realized in the DIL environments at the tactical edge where it is generally difficult to maintain the reliable long-haul communication links back to the remote cloud or tactically deployed cloudlets. Various military research efforts have been made to effectively maintain such network connectivity at the tactical edge, such as the Army's Warfighter Information Network - Tactical (WIN-T) [2] and DARPA's Content-Based Mobile Edge Networking (CBMEN) program [1], but they are generally based on the traditional Mobile Ad-Hoc Networks (MANETs) paradigm and unrealistically assume end-to-end wireless connectivity between warfighters [31]. Tactical cloud systems, such as the Army's Distributed Common Ground System (DCGS-A) and the Navy's Tactical Cloud (NTC), have also been deployed, but focus on ensuring universal and interoperable data storage and access. The low-echelon warfighters' needs of in-situ distributed data processing and analysis are generally ignored, but are critical to ensure the situational awareness at the tactical edge.

Instead, the major focus of this paper is to fully unleash the unexploited potential of collaborative mobile cloud computing capabilities among peer mobile devices. Being different from the existing work which assumes end-to-end wireless connectivity between warfighters [31], we consider that these warfighters at the tactical edge are only intermittently connected when they opportunistically *contact* each other, i.e., move into the communication range of each other's short-range radios. Such network paradigm, which has also been depicted as Delay/Disruption Tolerant Networks [9], is appropriate to represent the actual mobile networking scenarios at the tactical edge, due to the unpredictable environmental uncertainty.

Built on such network modeling, we develop a novel theoretical framework to adaptively configure the computational workload allocations among mobile devices, which are heterogeneous in their local computational capabilities and wireless networking conditions. Our framework takes both energy consumption and timeliness of such computational workload into account, to ensure that these workloads are migrated to and executed at the most appropriate network locations, so that they would be completed in time with the minimum amount of energy consumption in computation and data transmission. Such workload migration among peer mobile devices hence improves the efficiency of network resource utilization, enabling more prompt situational response to unpredictable events at the tactical edge.

More specifically, we have made the following detailed contributions:

- Our framework analytically ensures that the migration

This work was supported in part by the DOD Grant W911NF-14-1-0262.

of computational workloads among mobile nodes reduces the energy consumption of workload execution, and further improves the efficiency of network resource utilization by allowing recursive workload migration.
- Our framework also takes the time for delivering the computational results back to the workload initiator into consideration, so as to ensure timely workload completion over a large network scope.
- We also notice the locality of computation and operational data at the tactical edge, and allows multiple mobile nodes to coordinate with each other during workload execution for better efficiency of network resource utilization.

The rest of this paper is organized as follows. Section II briefly reviews the existing work in mobile cloud computing. Section III presents our models on the mobile network scenarios and computing systems, and also highlights our basic idea of the peer-to-peer mobile cloud computing framework. The details of our proposed framework are presented in Section IV and Section V. Section VI presents the performance evaluation results. Section VII concludes the paper.

II. Related Work

Mobile cloud computing (MCC) has been extensively studied. It integrates cloud computing into the mobile environment [7] and allows mobile users to efficiently utilize the cloud resources. Cloudlets, which are local resource-rich servers providing prompt cloud access to nearby mobile users, have been suggested to avoid the wireless transmission latency between smartphones and the remote cloud [26]. Other designs adopt various cloud computing techniques such as virtualization [16] and Service-Oriented Architecture (SOA) [32].

Workload offloading, as the key technical challenge in MCC, focuses on *how* to offload and *what* to offload. Research has been done to support efficient remote application execution. MAUI [6] relies on developers to specify the application partitioning by annotating remoteable methods. Later solutions improve the efficiency and reliability of workload offloading through synthesis and migration of VMs [4]. CloneCloud [3] creates an augmented clone of the local application on the cloud, and ThinkAir [21] enforces on-demand VM creation and resource allocation. Other schemes further improve the offloading generality by supporting multi-threaded [14] and interactive applications [23].

Appropriate decisions of application partitioning are the prerequisite to efficient workload offloading and the answer of "what to offload". Such decisions are based on the profiling data about application execution and system context, such as the CPU usage, energy consumption, and network latency, either through offline benchmark testing [3] or online application profilers [20], [21]. Odessa [23] assumes linear speedup over consecutive frames in a face recognition application. ThinkAir [21] defines multiple static offloading policies, each of which focuses on a sole aspect of system performance. The efficiency of offloading decisions in practice, however, are left unexamined and questionable. Analytical offloading framework, on the other hand, has been studied [24], [13], [6], [10], and they generally assume the method invocations to be represented by a stationary calling graph. Offloading decisions are then formulated as a graph cut problem [24], [13], integer optimization [6], or fuzzy logic decision process [10], assuming static execution patterns of user applications.

Cloud computing among peer mobile devices, on the other hand, has also been studied in intermittently connected mobile environments [29], [28]. In Serendipity [29], each computing task being performed at mobile devices is modeled as a Directed Acyclic Graph (DAG), such that the graph vertices are programs and the directed graph edges represent data flows between two programs. A graph vertex, further, is considered as a PNP-block consisting of a pre-process, a post-process, and a number of parallel tasks. These tasks are then allocated to mobile nodes upon opportunistic contacts based on various empirical heuristics, according to the different availability of network contact information. However, Serendipity only takes the task completion time into account when allocating computing tasks, and delivers the computational results back to the task initiator without any delay guarantee. Any expired results are simply discarded. The factors of contact prediction and multi-node coordination are also ignored. In contrast, our proposed framework for peer-to-peer mobile cloud computing quantitatively evaluates the probability of timely delivery of the computational results in a probabilistic manner, and also allows multiple mobile nodes undertaking similar computational workloads to adaptively coordinate with each other.

III. Overview

A. Network model

Opportunistic contacts among mobile devices are described by the network contact graph (NCG) $G(V, E)$, where the contact process between nodes $i, j \in V$ is modeled as an edge $e_{ij} \in E$, and e_{ij} only exists at time t if i and j have contacted before t. We assume that node contacts are symmetric, i.e., i contacts j whenever j contacts i, and G is therefore undirected. In the rest of this paper, we call a pair of nodes i, j as *contacted neighbors* if there exists $e_{ij} \in E$, and call the node set $\{j|e_{ij} \in E\} \subseteq V$ as the *contacted neighborhood* of node i.

The characteristics of an edge $e_{ij} \in E$ are mainly determined by the properties of inter-contact time (ICT) distribution among mobile nodes, which could vary according to the specific network scenarios and have been suggested by the existing research studies to follow either exponential [35], power-law [19], or log-normal [5] distributions. Based on these studies, we consider that any node in the network is able to probabilistically predict the future contacts between nodes i and j based on the knowledge about their contact pattern in the past. More specifically, such prediction is given in form of $p_{ij}(t)$ that indicates the probability that i and j will contact again within time t in the future. Precise computation of $p_{ij}(t)$ has been extensively studied by the existing work [22], [12], [11].

TABLE I
NOTATION SUMMARY

Notation	Explanation
C_i	The computational capability of node i
T_i	The computing task initiated at node i
w_i	Computational workload of T_i
d_i	Operational dataset of T_i with size s_i
D_i	Completion deadline of T_i
t_c	Current time
E_C^i	Node i's energy consumption for the local execution of T_i
$E_T^{(ij)}(s_i)$	The energy consumption of data transmission for workload migration of T_i from node i to j
$p_{ij}(t)$	Probability that node i and j contact within time t in future

B. System Model

In this paper, we focus on investigating the most efficient computing strategy for executing a specific computing task $T_i = \{w_i, d_i, D_i\}$ initiated by a node i at time 0, where w_i indicates the amount of computational workload of T_i, d_i is the operational dataset of T_i with the size s_i, and D_i is the completion deadline of T_i. Different computing tasks are handled separately. The problem of scheduling multiple computing tasks being executed at the same node is orthogonal to the main focus of this paper, and could be effectively addressed by the existing work [29]. On the other hand, identical computing tasks at different nodes due to data locality will be handled in this paper through cooperative coordination among mobile nodes.

We consider that each node i has a local computing capability C_i that indicates the amount of computational workload that i can undertake in unit time. As a result, the execution time of the task T_i at node i is w_i/C_i. Without loss of generality, we assume that the energy consumption of a specific mobile node due to computation is proportional to the execution time of the corresponding computing task. The longer it takes to execute the computing task locally, the more energy is consumed. Such rate of energy consumption over time, however, depends on the specific system model and computing capability, and would be heterogeneous over different nodes. Node i's energy consumption due to the local execution of T_i is hence $E_C^i = k_c^i \cdot w_i/C_i$, which could be precisely estimated locally before task execution by adopting various models for mobile system energy consumption [18], [34].

To remotely execute a computing task T_i at another node j, T_i's operational dataset d_i needs to transmitted from node i to node j when i and j contact each other. The energy consumption of such data transmission depends on the specific conditions of wireless channel quality and bandwidth between i and j during data transmission. In particular, existing work shows that such energy consumption does not linearly increase with the amount of data being transmitted [6]. We denote $E_T^{(ij)}(s)$ as the amount of energy being consumed by transmitting a data item with size s between node i and j.

The notations being used throughout the rest of this paper are summarized in Table I.

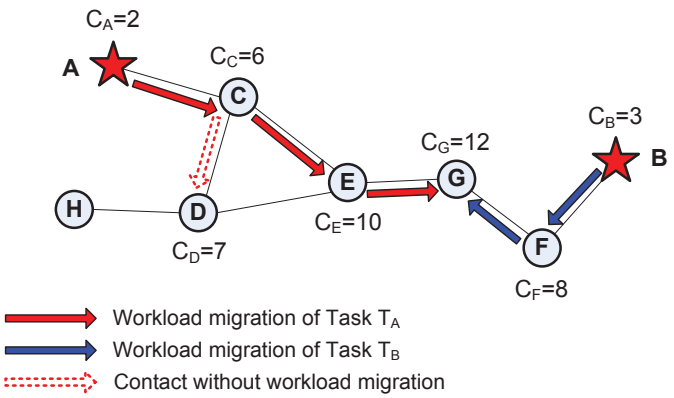


Fig. 1. The big picture of opportunistic peer-to-peer mobile cloud computing

C. The Big Picture

Our basic idea of ensuring the efficiency of peer-to-peer mobile cloud computing is to develop a distributed framework which takes both energy consumption, task execution time, and multi-node coordination into account in a unified and probabilistic manner. The big picture of our proposed mobile cloud computing framework is illustrated in Figure 1, where the computational capabilities of individual mobile nodes are listed, and each edge between two mobile nodes represents the contact pattern between the two nodes. As shown in Figure 1, when node A contacts node C which has a much higher computational capability than A, A migrates the computational workload of its computing task T_A to C, so that T_A could be completed with much shorter time and less energy consumption. Such saving of computational energy consumption exceeds the data transmission cost between A and C, so as to increase the overall efficiency of network resource utilization. Afterwards, C acts as the "sub-initiator" of T_A and recursively migrates the workload of T_A to E. In contrast, C decides not to migrate T_A to D when they contact, even if $C_D > C_C$. The main reason for this decision is the low contact frequency between C and D that reduces the likelihood of timely delivery of the computational results of T_A back to A in time. More details of quantitatively calculating such delivery likelihood and making migration decisions accordingly will be presented in Section IV.

Moreover, Figure 1 also demonstrates that another computing task T_B is initiated at node B simultaneously, and is later migrated to node G with higher computational capability. In particular, due to the locality of environmental contexts and operational dataset, the operations of T_A and T_B may partially overlap with each other. As a result, when E and G contact each other, they have the chance to migrate the workload of one task to another, and hence execute the two tasks at the same location. Such coordination avoids repetitive execution of identical computational operations at different nodes, so as to eliminate energy consumption on unnecessary computations. The major challenge, however, is to appropriately make such coordinating decisions to ensure that the deadlines of both tasks could be met. We will present details of such multi-node coordination in Section V.

IV. Opportunistic Workload Migration

In this section, we develop our probabilistic framework for efficient decisions of workload migration among peer mobile devices, when they opportunistically contact each other. We take the characteristics of both network data transmission and energy consumption into account, so as to migrate the computational workloads to the most appropriate network locations and ensure that 1) the results of remote task execution could be timely returned back to the task initiator before the task expires, and 2) the amount of network energy consumption saved by remote execution is maximized.

A. Basic Approach

We first consider the basic scenario: when node i, the initiator of a computing task T_i, contacts another node j, i decides whether to migrate the execution of T_i from i to j. Without loss of generality, we assume that $C_i < C_j$, otherwise it is trivially unnecessary for migrating T_i for remote execution. The major challenge for such decision of workload migration, however, is two-fold. On one hand, such workload migration should reduce the cumulative energy consumption of the execution of T_i, which stems from both the computation of task execution and wireless transmission of the operational dataset of T_i for such execution. On the other hand, i needs to determine whether j is the best choice for workload migration among all of its contacted neighbors, i.e., whether it is better to migrate T_i to j instantaneously, or to wait for contacting with another node with higher computational capability.

According to our system model described in Section III-B, when $C_i < C_j$, the execution of T_i on node j consumes less energy than that on node i due to the shorter execution time, and such difference in energy consumption can be computed as $E_C^i - E_C^j$. Therefore, when the execution of T_i is migrated from i to j, the amount of energy being saved is

$$S_C^{(ij)} = E_T^{(ij)}(s_i) - (E_C^i - E_C^j), \tag{1}$$

i.e., the amount of energy being saved from computation of T_i on j should exceed the energy consumption of transmitting d_i from i to j.

Moreover, for node i to make the appropriate migration decision between j and another contacted neighbor k, we take the estimations of their future contacts into consideration, so as to make sure that the computational results of T_i could be returned back to i in time. More specifically, assuming that T_i is initiated by i at time 0 and the ICT between i and j is generally much longer than the workload execution time, the probability for such timely delivery of computational results from j to i is simply $p_{ij}(D_i - t_c)$, i.e., the computational result of T_i is delivered back to i when the next time j contacts i. Comparatively, such probability for another contacted neighbor k of i is

$$p_{ik}^{(2)}(D_i - t_c) = \int_{t_c}^{D_i} (f_{ik}(t) \otimes f_{ik}(t))dt, \tag{2}$$

where $f_{ik}(t)$ is the Probability Density Function (PDF) of the ICT distribution between node i and k, and $\otimes$ indicates convolution of functions. Eq. (2) hence measures the probability that k contacts i twice before T_i expires. i is then able to migrate T_i to k during the first contact, and k return the computational results back to i when they contact again. By combining Eq. (1) and Eq. (2), we conclude that i only migrates T_i to k instead of j if

$$S_C^{(ik)} \cdot p_{ik}^{(2)}(D_i - t_c) > S_C^{(ij)} \cdot p_{ij}(D_i - t_c). \tag{3}$$

In general, each time when i contacts another node j, it will perform the comparison specified by Eq. (3) for each of its contacted neighbors, and hence selects the most appropriate node for workload migration.

B. Recursive Workload Migration

The decision of workload migration made by a task initiator i is only made based on i's local knowledge about the computational capabilities and contact patterns of other nodes within its contacted neighborhood, and hence may not be optimal from the global perspective of the network. In this section, we aim to further improve the efficiency of such workload migration, by allowing the intermediate executers to further migrate their local workloads recursively, when they opportunistically contact other nodes.

Generally speaking, when an intermediate executer j, which is currently undertaking a computational task T_i initiated by node i, contacts another node k, j decides whether to recursively migrate T_i to k following the similar process as described in Section IV-A. The major challenge of such recursive migration, however, is that the computational results of T_i need to be sent back to the original task initiator i in time, and the estimation of such data delivery time is more challenging due to the larger network scope being involved into such migration. To precisely estimate such data delivery time across multiple hops on the NCG, we formulate the multi-hop data transmission characteristics among intermittently connected mobile nodes in the form of an opportunistic path defined as follows.

***Definition** 1: Opportunistic path*
A n-hop opportunistic path $P_{AB} = (V_P, E_P)$ between two nodes A and B consists of a node set $V_P = \{A, N_1, N_2, ..., N_{n-1}, B\}$ and an edge sequence $E_P = \{e_1, e_2, ..., e_n\}$ such that $V_P \subset V$ and $E_P \subset E$.

Further, we define the weight of an opportunistic path between nodes A and B as the probability $p_{AB}(T)$ that data could be forwarded from A to B along P_{AB} within time T. It is easy to see that the computation of $p_{AB}(T)$ mainly depends on the ICT distributions of the node pairs on the opportunistic path. More specifically, letting the random variable X_i with a PDF $p_i(x)$ indicate the ICT between node N_i and N_{i+1} on the opportunistic path, the total time for transmitting data from A to B along P_{AB} is $Y = \sum_{i=1}^{n} X_i$, and the PDF $p_Y(x)$ is calculated by convolutions on $p_i(x)$ as

$$p_Y(x) = p_1(x) \otimes p_2(x) ... \otimes p_n(x).$$

The corresponding weight of the opportunistic path is then computed as

$$p_{AB}(T) = \int_0^T p_Y(x)dx. \quad (4)$$

Eq. (4) shows that the calculation of the opportunistic path weight is a generalization of Eq. (2), and varies according to the different models of ICT distributions. For example, when each X_i is assumed to be exponentially distributed with the parameter λ_i, the opportunistic path weight could be expressed as [25]

$$p_{AB}(T) = \sum_{i=1}^{n} C_i^{(n)} \cdot (1 - e^{-\lambda_i T}),$$

where $C_i^{(n)} = \prod_{j=1, j\neq i}^{n} \frac{\lambda_j}{\lambda_j - \lambda_i}$.

In practice, the information of the opportunistic path from the current task executer to the original task initiator i is recursively maintained by the intermediate executers, each time when the task T_i is migrated. As a result, when j makes the decision of workload migration on its contacted neighbor k, it will substitute the quantity $p_{ik}(D_i - t_c)$, which is specified in Eq. (4), into Eq. (3), so as to ensure that the computational result of T_i could be timely returned back to i from k.

V. Multi-Node Coordination

In practical application scenarios at the tactical edge, both the workload and operational dataset of computational tasks being initiated and executed at mobile nodes could exhibit significant locality, due to the dynamic squad formation and tactical missions in the theater. As a result, the computational task initiated by a mobile node could be highly similar to these tasks initiated by other nodes nearby. In this section, we present techniques which aim at efficient coordination among the executers of computational tasks with noticeable similarity, so as to further improve the efficiency of network resource utilization by avoiding repetitive computational overhead on the same workload.

A. Coordinated Execution of Similar Tasks

Our basic idea of such multi-node coordination, as illustrated by Figure 1, is that the executers of two similar computational tasks exchange the task information when they contact each other, so as to identify the overlapping part of their computational tasks. Such overlapping part could stem from either the similarity of operational dataset, or similar pre- and post-processing on such data. Afterwards, the decision of workload migration, which is similar to that described in Section IV, will be made at the task executer with less computational capability, to decide whether to migrate the overlapping workload to another executer with higher computational capability. In practice, such workload migration is able to be systematically realized by existing application partition techniques [6], [14], which enable remote execution for a partition of user applications at the level of either methods or threads. As a result, such overlapping workload will only be executed once for both computational tasks, so as to reduce the overall energy consumption for peer-to-peer computing. The execution results are then returned back to the other executer when they contact again in the future. For example in Figure 1, the computational results will be returned to E when E and G contact again. E is then able to accomplish the local execution of T_A and return the result back to A.

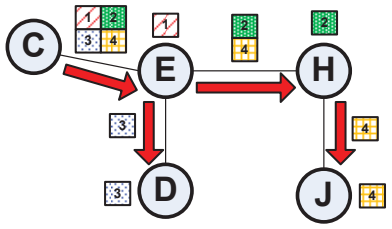


Fig. 2. The big picture of opportunistic peer-to-peer mobile cloud computing

B. Concurrent Task Executions

In cases that the task completion deadline is approaching or the probability of timely data delivery computed in Section IV is not sufficiently high, we also propose to intentionally employ multiple executers to undertake the computational task concurrently, so as to ensure that the task could be completed in time.

Such concurrent task execution is illustrated in Figure 2. Each time when an executer of a computational task contacts another node and decides to migrate its local workload to the new executer, instead of migrating the whole computational workload as described in Section IV, it adaptively partitions such workload, which is then executed simultaneously at both executers. As shown in Figure 2, such partitions are recursively conducted among multiple mobile nodes, according to the specific node contact patterns and computational capabilities. The subsequent concurrent executions at multiple nodes could hence expedite the task completion.

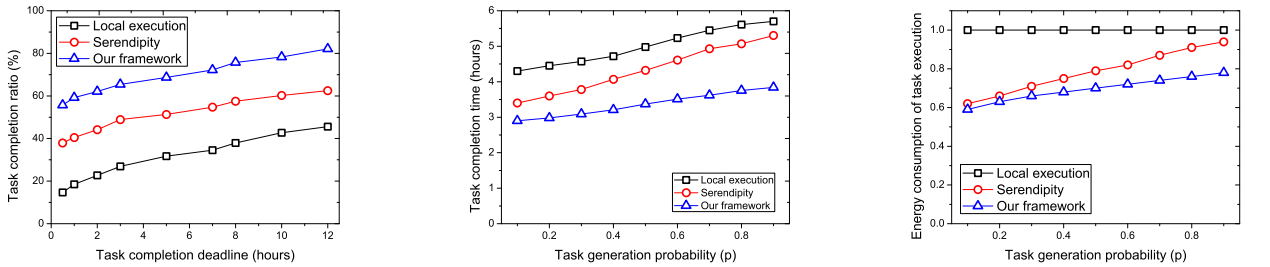
This concurrency also enables flexible balancing between the peer-to-peer computing performance and overhead. The more mobile nodes are involved into the execution of a specific computing task, the faster that this task could be completed, at the cost of higher data transmission overhead among multiple mobile nodes. In practice, such tradeoff needs to be adaptively adjusted according to the specific task completion deadline and network contact patterns. We refer the development of such adjustment algorithm and related formal analysis as our future work.

VI. Performance Evaluation

In this section, we evaluate the performance of our proposed framework for opportunistic peer-to-peer mobile cloud computing. The performance of our proposed framework is first evaluated against the baseline of local workload execution, and is then compared with the existing scheme of peer-to-peer mobile cloud computing, i.e., Serendipity [29].

A. Simulation Setup

Our experiments are performed over various realistic DIL network traces. These traces record contacts among users



(a) Task completion ratio (b) Task completion time (c) Energy consumption of task execution

Fig. 3. Performance of our probabilistic framework on opportunistic peer-to-peer mobile cloud computing

carrying hand-held mobile devices at various networking environments, including the university campus (MIT Reality [8]) and conference site (Infocom [17]). These devices periodically detect their peers via their Bluetooth interfaces, and a contact is then recorded when two devices move close to each other. As summarized in Table II, the two traces differ in their scale, contact density, and contact durations.

Without loss of generality, we assume that each mobile node in the network has a specific level of computational capability that is normally distributed within the range of $[0, 1]$. Correspondingly, each node has a probability p to start a new computing task every hour. It randomly generates the task's computational workload also in the range of $[0, 1]$, and randomly generates the task completion deadline in the range $[0.5{\cdot}T_{avg}, 1.5{\cdot}T_{avg}]$ where T_{avg} is the average deadline among all the tasks. The quantity of p is varied to simulate different levels of computational workload throughout the network. We assume that the wireless channel bandwidth is 1 Mbps, and the sizes of the operational datasets of computational tasks are uniformly distributed in $[10Mb, 50Mb]$. The PowerTutor model [15] is then applied to estimate the energy consumption of computation and data transmission.

The following metrics are used in our experiments:

- **Task completion ratio**, which is the percentage of computational tasks being completed before they expire.
- **Task completion time**, which is the average time being taken to complete a computational task.
- **Energy consumption of task execution**, which is the average amount of energy being consumed by executing each computational task. Such energy consumption is normalized against the baseline of local execution.

TABLE II
TRACE SUMMARY

Trace	MIT Reality	Infocom
Number of devices	97	78
Number of internal contacts	114,046	182,951
Duration (days)	246	4
Contact detection period (secs)	120	120
Pairwise contact freq. (per day)	4.6	7.52
Average contact duration (hours)	0.57	0.142

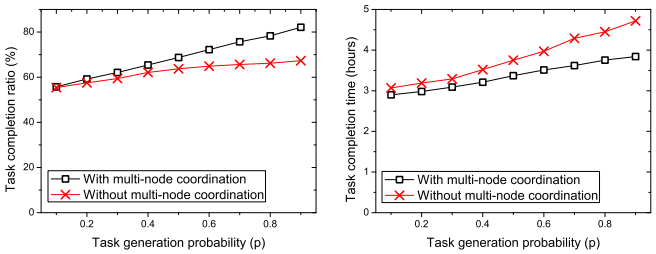
B. Performance of Opportunistic Peer-to-Peer Mobile Cloud Computing

We first evaluate the performance of our proposed probabilistic framework on improving the performance and energy efficiency of opportunistic peer-to-peer mobile cloud computing. The experimental results on the MIT Reality trace are shown in Figure 3. Generally speaking, since our proposed framework takes both the elapsed time of task execution and transmission of the tasks' operational datasets into consideration, and probabilistically quantifies the likelihood of timely data delivery between the remote task executers and local task initiators, it is able to ensure that tasks are executed at appropriate network locations and returned back to the task initiators on time. As shown in Figure 3(a), the task completion ratio of our framework is more than 40% higher than that of local execution, and 20% higher than that of Serendipity when the task completion deadline changes from 1 hour to 12 hours. Similarly, Figure 3(b) shows that when the task completion deadline is uniformly set at 6 hours, our framework is able to significantly reduce the task completion time. Such reduction is particularly noticeable when the task generation probability (p) is higher than 0.6 and the subsequent computational workload in the network is intensive. Figure 3(b) shows that our framework performs more than 30% better than Serendipity in such cases.

On the other hand, Figure 3(c) also shows that our proposed framework is able to dramatically reduce the energy consumption of computational task executions, which includes the energy consumed by both computations and data transmissions. The major reason for such reduction is that our framework quantitatively takes the measurements of energy consumption into consideration when making workload migration decisions. As shown in Figure 3(c), the amount of energy being saved by our framework increases along with the task generation probability, such that more energy would be saved when the overall computational workload in the network becomes more intensive. When $p > 0.6$, the energy consumption of our framework is around 20% less than that of Serendipity.

C. Effectiveness of Multi-Node Coordination

Moreover, we also experimentally evaluate the effectiveness of our approach to multi-node coordination proposed in Section V. The experiment results based on the Infocom trace are shown in Figure 4. Generally speaking, Figure 4 shows that our proposed approach to multi-node coordination is able to significantly improve the efficiency of remote workload execution. In particular, when the amount of computational workload in the network increases, different computational



(a) Task completion ratio (b) Task completion time

Fig. 4. Effectiveness of multi-node coordination

tasks are more likely to overlap with each other, and hence multi-node coordination leads to more increase of the task completion ratio and reduction of the task completion time. More specifically, such difference is smaller than 5% when $p \leq 0.3$, but would be up to 30% when $p > 0.8$.

VII. Conclusions

In this paper, we present an analytical framework enabling opportunistic peer-to-peer mobile cloud computing among mobile devices at the tactical edge, which is featured by the Disconnected, Intermittent, and Limited-bandwidth (DIL) network environments. Our proposed framework ensures appropriate workload migration by quantitatively measuring the energy consumption and data transmission delay of computational workloads simultaneously. As a result, we ensure that workload migration is able to not only reduce the energy consumption of workload execution, but also to deliver the computational results back to the task initiator on time. Extensive simulation results show that our proposed framework could significantly improve the task completion ratio and time, as well as reducing the power consumption of task executions.

References

[1] DARPA content-based mobile edge networking (CBMEN).

[2] M. Acriche, C. Holsinger, A. Staikos, J. Dimarogonas, and R. Sonalkar. Multi-dimensional, assured, robust communications for an on-the-move network. In *Proceedings of the Military Communications Conference (MILCOM)*, pages 2378–2383, 2005.

[3] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti. Clonecloud: Elastic execution between mobile device and cloud. In *Proceedings of the 6th Conference on Computer Systems*, pages 301–314, 2011.

[4] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield. Live migration of virtual machines. In *Proceedings of USENIX NSDI*, pages 273–286, 2005.

[5] V. Conan, J. Leguay, and T. Friedman. Characterizing pairwise inter-contact patterns in delay tolerant networks. In *Proceedings of the 1st Int'l Conf on Autonomic Computing and Communication Systems*, 2007.

[6] E. Cuervo, A. Balasubramanian, D.-k. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl. Maui: Making smartphones last longer with code offload. In *Proceedings of ACM MobiSys*, pages 49–62, 2010.

[7] H. T. Dinh, C. Lee, D. Niyato, and P. Wang. A survey of mobile cloud computing: architecture, applications, and approaches. *Wireless Communications and Mobile Computing*, 2011.

[8] N. Eagle and A. Pentland. Reality Mining: Sensing Complex Social Systems. *Personal and Ubiquitous Computing*, 10(4):255–268, 2006.

[9] K. Fall. A Delay-Tolerant Network Architecture for Challenged Internets. *Proc. SIGCOMM*, pages 27–34, 2003.

[10] H. R. Flores Macario and S. Srirama. Adaptive code offloading for mobile cloud applications: Exploiting fuzzy sets and evidence-based learning. In *Proceeding of the 4th ACM MCS workshop*, 2013.

[11] W. Gao, G. Cao, T. La Porta, and J. Han. On exploiting transient social contact patterns for data forwarding in delay tolerant networks. *IEEE Transactions on Mobile Computing*, 12(1):151–165, 2013.

[12] W. Gao, Q. Li, B. Zhao, and G. Cao. Social-aware multicast in disruption-tolerant networks. *IEEE/ACM Transactions on Networking*, 20(5):1553–1566, 2012.

[13] I. Giurgiu, O. Riva, D. Juric, I. Krivulev, and G. Alonso. Calling the cloud: Enabling mobile phones as interfaces to cloud applications. In *Proceedings of the 10th International Middleware Conference*. 2009.

[14] M. S. Gordon, D. A. Jamshidi, S. Mahlke, Z. M. Mao, and X. Chen. Comet: Code offload by migrating execution transparently. In *Proceedings of USENIX OSDI*, pages 93–106, 2012.

[15] J. Huang, Q. Xu, B. Tiwana, Z. Mao, M. Zhang, and P. Bahl. Anatomizing Application Performance Differences on Smartphones. In *Proceedings of ACM MobiSys*, pages 165–178. ACM, 2010.

[16] G. Huerta-Canepa and D. Lee. A virtual cloud computing provider for mobile devices. In *Proceedings of the 1st ACM MCS Workshop*, 2010.

[17] P. Hui, J. Crowcroft, and E. Yoneki. Bubble rap: social-based forwarding in delay tolerant networks. *Proceedings the ACM MobiHoc*, 2008.

[18] A. Kansal and F. Zhao. Fine-grained energy profiling for power-aware application design. *ACM SIGMETRICS Performance Evaluation Review*, 36(2):26–31, 2008.

[19] T. Karagiannis, J.-Y. Boudec, and M. Vojnovic. Power law and exponential decay of inter contact times between mobile devices. *Proceedings of the 13th Annual International Conference on Mobile Computing and Networking (MobiCom)*, pages 183–194, 2007.

[20] R. Kemp, N. Palmer, T. Kielmann, and H. Bal. Cuckoo: A computation offloading framework for smartphones. In *Mobile Computing, Applications, and Services*, pages 59–79. Springer, 2012.

[21] S. Kosta, A. Aucinas, P. Hui, R. Mortier, and X. Zhang. Thinkair: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading. In *Proceedings of INFOCOM*, 2012.

[22] S. C. Nelson, M. Bakht, and R. Kravets. Encounter-based routing in dtns. In *Proceedings of IEEE INFOCOM*, pages 846–854, 2009.

[23] M.-R. Ra, A. Sheth, L. Mummert, P. Pillai, D. Wetherall, and R. Govindan. Odessa: Enabling interactive perception applications on mobile devices. In *Proceedings of MobiSys*, pages 43–56, 2011.

[24] J. S. Rellermeyer, O. Riva, and G. Alonso. Alfredo: An architecture for flexible interaction with electronic devices. In *Proceedings of the 10th International Middleware Conference*, pages 22–41, 2008.

[25] S. M. Ross. *Introduction to probability models*. Academic Press, 2006.

[26] M. Satyanarayanan. Mobile computing: The next decade. *ACM SIGMOBILE Mobile Computing and Communications Review*, 15(2):2–10, 2011.

[27] K. Scott, T. Refaei, N. Trivedi, J. Trinh, and J. P. Macker. Robust communications for disconnected, intermittent, low-bandwidth (dil) environments. In *Proceedings of the Military Communications Conference*, pages 1009–1014, 2011.

[28] C. Shi, M. H. Ammar, E. W. Zegura, and M. Naik. Computing in cirrus clouds: the challenge of intermittent connectivity. In *Proceedings of the 1st ACM workshop on Mobile cloud computing*, pages 23–28, 2012.

[29] C. Shi, V. Lakafosis, M. H. Ammar, and E. W. Zegura. Serendipity: enabling remote computing among intermittently connected mobile devices. In *Proceedings of ACM MobiHoc*, pages 145–154. ACM, 2012.

[30] J. Sonnenberg. Disconnected, intermittent, limited (dil) communications management technical pattern. *Network Centric Operations Industry Consortium*, 2009.

[31] S. Taneja and A. Kush. A survey of routing protocols in mobile ad hoc networks. *International Journal of Innovation, Management and Technology*, 1(3):2010–0248, 2010.

[32] W.-T. Tsai, X. Sun, and J. Balasooriya. Service-oriented cloud computing architecture. In *Proceedings of the 7th International Conference on Information Technology*, pages 684–689, 2010.

[33] P. Volgyesi, G. Balogh, A. Nadas, C. B. Nash, and A. Ledeczi. Shooter localization and weapon classification with soldier-wearable networked sensors. In *Proceedings of ACM MobiSys*, pages 113–126, 2007.

[34] L. Zhang, B. Tiwana, Z. Qian, Z. Wang, R. P. Dick, Z. M. Mao, and L. Yang. Accurate online power estimation and automatic battery behavior based power model generation for smartphones. In *Proceedings of the 8th international conference on hardware/software co-design and system synthesis*, pages 105–114, 2010.

[35] H. Zhu, L. Fu, G. Xue, Y. Zhu, M. Li, and L. M. Ni. Recognizing Exponential Inter-Contact Time in VANETs. In *Proceedings of INFOCOM*, 2010.