Service Compositions in Challenged Mobile Environments Under Spatiotemporal Constraints

Daisuke Kasamatsu
Fujitsu Laboratories Ltd.
Kanagawa, Japan
kasamatsu.daisu@jp.fujitsu.com

Mohan Kumar
Rochester Institute of Technology
New York, USA
mjkvcs@rit.edu

Peizhao Hu
Rochester Institute of Technology
New York, USA
ph@cs.rit.edu

***Abstract*—Opportunistic network created among mobile devices in challenged environments can be effectively exploited to provide application services. However, data and services may be subject to space and time constraints in challenged environments where it is critical to complete application services within given spatiotemporal limits. This paper discusses an analytical framework that takes into account human mobility traces and provides quantitative measures of the spatiotemporal requirements for service sharing and composition in challenged opportunistic environments. The analytical results provide estimates on feasibility of service sharing and service compositions for various mobility models. To validate the framework, we conduct simulation experiments using multiple human mobility and synthesized datasets. In these experiments, we analyze service composition feasibility, service completion rate and time for resource utilization.**

1. Introduction

Cloud computing allows resource-constrained mobile devices to offload computation-intensive tasks (e.g., face recognition) over last-mile connectivities (WiFi or cellular networks). In challenged networks however, access to the cloud may be slow or intermittent due to lack of WiFi or cellular networks. In situations where connectivity is intermittent due to coverage dead-spot or insufficient bandwidth, exploitation of available resources in the neighborhood is a preferred alternative. Cirrus [1] and Serendipity [2] are examples of instantaneous cloud formed directly among neighboring devices. In this paper, we exploit opportunistic encountering of mobile devices for service sharing and service composition. Modern mobile devices, including wearable devices, offer one or more services that can be combined with other services offered by encountering devices. To ensure efficiency of computations and data validity, the service compositions need to consider the spatiotemporal constraints. This is difficult since node contacts are opportunistic and data transmissions are best-effort in opportunistic networks. Even though service compositions are possible, the results could be dated if the compositions of services rely on devices that have infrequent encountering for data transmission. This paper discusses an analytical framework to provide quantitative measures of the feasibility of support service compositions in the presence of spatiotemporal constraints.

We use this analytical framework to estimate service sharing parameters for service composition, where a rich set of distributed services can be composed to provide mobile users with a multitude of application-level services. First, we develop spatiotemporal feasibility graphs (STFG) to capture spatiotemporal service sharing in opportunistic environments. STFG captures available service compositions that satisfy service completion time and geographic distance requirements. Then, we develop an analytical framework based on STFG to estimate success of service compositions and expected computing resources. In particular, the analytical framework comprises three functions: (i) estimating service completion rate, (ii) optimizing parameter setting, and (iii) estimating resource utilization time. To demonstrate applicability of the analytical framework, we extensively perform trace-driven simulations applied to a wide range of empirical datasets from real-world traces and synthetic models of human walk mobility. Our simulation results verify that the analytical framework can make the above types of estimations in any opportunistic environment. The estimations are validated with projected datasets to examine their tolerance against future trajectory change. Three main contributions of this paper are summarized as, (i) *Spatiotemporal feasibility graphs (STFG)* - A graph model to capture spatiotemporal service sharing in opportunistic environments; (ii) *Analytical framework for service sharing* - Analytic functions to estimate service sharing parameters based on STFG; and (iii) *Extensive trace-driven simulations* - Applicability of the framework is demonstrated with a wide range of empirical datasets.

The novelty of this paper lies in the development of generic analytical framework to provide accurate guidance for mobile users in terms of space and time. The framework would be useful for opportunistic service sharing in any mobility scenario and also enable effective service composition in opportunistic environments.

2. Related Work

Murray et al. [3] introduced crowd computing, where opportunistic networks can be used to spread computational tasks and collect results. They estimated computational capacity in such environments and also developed a task-farming algorithm to exchange tasks between mobile nodes. They analyzed real-world contact traces to show that exploiting social structures boosts up successful task completions. Passarella et al. [4] proposed a service provisioning model for fault-tolerance. The model measures the expected service completion time and then determines the optimal number of parallel executions, in the presence of multiple resource providers for a given service request. They showed that replicated parallel executions reduce service completion time. Ferrari et al. [5] have proposed a metric of expected resource availability to capture the topological impact of resources. Their analysis with real-world contact traces showed that availability correlates well with the inter-node contact probability. Li et al. [6] introduced mobile cloudlets. Their trace-based analysis demonstrated that intermittent connections have little negative effect on optimal performance when a task is delay-tolerant. Sadiq et al. [7] proposed a service composition algorithm to select a service sequence based on a service load and encounter-based timer to measure relative distance to other nodes.

Shi et al. [2] have proposed Serendipity to enable a mobile user to utilize remote computational resources available in its mobile environment. They designed a task allocation algorithm to disseminate tasks among mobile devices by accounting for the available connectivity. Using the Emulab emulator with real-world contact traces and synthetic mobility models, they demonstrated the potential to speed up computation as well as save energy for the user. Mitbaa et al. [8], [9], [10] have proposed a power balancing algorithm for computation offloading. The algorithm schedules computational tasks among a set of mobile devices so that no device has depleted its battery as long as possible. Measuring the performance of power and computation between two android phones via Bluetooth/WiFi Direct, they demonstrated that the algorithm extends network lifetime. Wang et al. [11] have formulated an opportunistic offloading problem to determine the amount of computation offloaded onto other devices. They developed an optimization method with the aid of statistic property of contact rates. Their trace-based simulation study showed that the method achieves high service completion rate. However, the focus of above work is on computation offloading for computation speedup but not on service sharing for successful service utilizations, especially for service composition.

3. Preliminaries

3.1. Problem Statement

In opportunistic computing, each mobile device has dual roles — provider or seeker. A seeker utilizes services on encountered providers by exploiting direct contact opportunities between the two. Service sharing includes sending of service requests from a seeker to providers and then the seeker receiving results from providers. In certain application contexts, service requests could lose their time and/or space validities. For example, if a request goes too far away from a seeker, the seeker cannot receive the results from providers in a timely manner. Thus, it is important to complete service requests within spatiotemporally valid limits. To handle spatiotemporal limits, we consider time to live (TTL) and space to live (STL) for each service request. As a key metric, service completion rate is defined as the number of successful services within TTL and STL to the total number of service requests.

In this paper, we estimate service sharing parameters for the following queries:

- *Query 1.* Given TTL and STL, what is the estimated service completion rate?
- *Query 2.* What are the optimal TTL and STL values to achieve a given service completion rate?
- *Query 3.* Given a time window, is it possible to successfully complete the request?

In spatiotemporal service sharing in opportunistic environments, a seeker makes Query 1 to know service completion rate expectancy. Subsequently, in order to avoid setting too long or too short TTL/STL, the seeker makes Query 2. The seeker makes Query 3 to check the success likelihood of remote service execution.

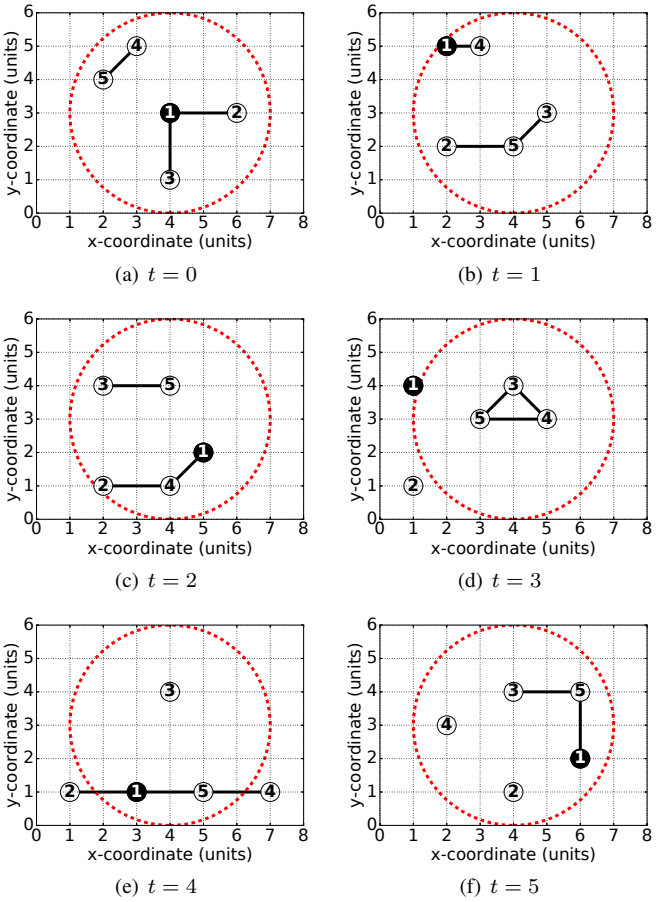
3.2. Opportunistic Networking Model

The position of mobile devices and their connectivity in a snapshot can be modeled as a graph. A collection of these graphs, which capture the mobility of devices and evolution of their connectivities over a time window, are call time-varying graphs [12].

Time-Varying Graphs (TVG). A graph snapshot is described as $G_t = (V_t, E_t, \zeta)$, where t is time instance. A set of nodes at time t is denoted by V_t, and $E_t \subseteq V_t \times V_t$ is the set of edges between nodes at time t. A latency function, denoted by ζ, indicates edge traversal time (i.e., transmission time) taken to cross a given edge.

Journey. A journey from a source node src to a destination node dst, denoted by $J_{src,dst}$, is described as a sequence of tuples $\langle e_{u,v}, t\rangle$, which represents a contact between nodes u and v at time t.

Fig. 1 illustrates an example of mobility scenario. Node n_1 has a contact with n_2 at $t = 0$, and then n_2 has a contact with n_5 at $t = 1$. If the contact between these pairs of nodes remains available until data is transferred successfully, then there is an opportunistic communication link from n_1 to n_5 over the period of two time ticks. In this case, the communication from n_1 to n_5 is described as $J_{1,5} = \{\langle e_{1,2}, 0\rangle, \langle e_{2,5}, 1\rangle\}$.



(a) $t = 0$ (b) $t = 1$ (c) $t = 2$ (d) $t = 3$ (e) $t = 4$ (f) $t = 5$

Figure 1. An example of time-varying graphs at six time slices: edges represent pair-wise contacts between mobile nodes; node 1 generates a request at $t = 0$; and dotted circles correspond to the spatial-validity area of request.

3.3. Service Computing Model

A service scenario can be modeled using service graphs [7], where multiple services can be composed based on their input and output requirements.

Service Graph (SG). A service graph can be described as $G_s = (V_s, E_s)$, where V_s is the set of data and E_s is the set of services. $d_i \in V_s$ represents data. $\langle s_{i,j}, v\rangle \in E_s$ corresponds to a service $s_{i,j}$, where i and j are input/output data types to/from the service supplied by a provider v, for all $i < j$ to avoid cyclic compositions. A directed edge from one vertex to another exists if a service $\langle s_{i,j}, v\rangle$ is available for input/output data d_i and d_j.

Service Path. A path from a start data d_{sta} to an end data d_{end}, denoted by $P_{d_{sta},d_{end}}$, is described as a sequence of services $\langle s_{i,j}, v\rangle$, representing possible concatenation of services.

An example of service scenario is depicted in Fig. 2. Suppose $s_{1,2}$ and $s_{2,3}$ represent text-translation and text-to-speech services. d_3 is voice data, whereas d_1 and d_2 are text data in different languages, respectively. Then, $s_{1,3}$ represents a composite service of both translation and text-to-speech. In this case, P_{d_1,d_3} has two possible service paths: $\{\langle s_{1,3}, n_2, n_4\rangle\}$ and $\{\langle s_{1,2}, n_3\rangle, \langle s_{2,3}, n_5\rangle\}$, where n_i is a provider of the corresponding services.

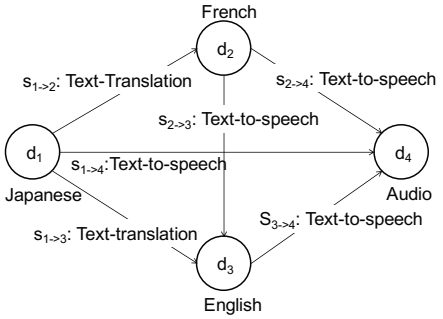


Figure 2. An example service graph: nodes represent data; and edges represent services as a tuple $\langle seeker, provider\rangle$

4. Spatiotemporal Feasibility graphs

4.1. Spatiotemporal Service Sharing

Let δ be the TTL, that specifies a temporal-validity duration of request by limiting the maximum delay from departure time t_s. Let γ be the STL, which stipulates a spatial-validity area of request with the radius γ from a source position by restricting the maximum distance from the location of request generated. Spatiotemporal service sharing can be regarded as a round-trip journey between a seeker and providers such that temporal and spatial lengths are within TTL and STL.

Using an example of mobility and service scenarios shown in Figs. 1 and 2, spatiotemporal service sharing is exemplified as follows. A seeker n_1 makes a service request for data d_3 at time $t_s = 0$. Execution time is one unit for each service while edge traversal time is one unit for each edge.[1] With $TTL = 5$ units, $STL = 3$ units, the source node n_1 at $(4, 3)$ in Fig. 1 want to find a service composition to convert data from d_1 to d_3; such that, constructing a service path P_{d_1,d_3}. Two possible service paths are derived from SG in Fig. 2: $\{\langle s_{1,3}, n_2, n_4\rangle\}$ and $\{\langle s_{1,2}, n_3\rangle, \langle s_{2,3}, n_5\rangle\}$. The second path satisfies both TTL and STL, whereas the first path exceeds STL.

4.2. Definition of STFG

A spatiotemporal feasibility graph can be derived by combining the corresponding TVG and SG.

Temporal Length of Journey. Temporal length, L_t, is the difference between arrival time at a destination node and the corresponding departure time t_s at the source node. When there are multiple journeys from a source node to a destination node at time t_s, the $earliest$ journey is given by

1. Execution time and edge traversal time can be arbitrary for each node and edge over time, respectively.

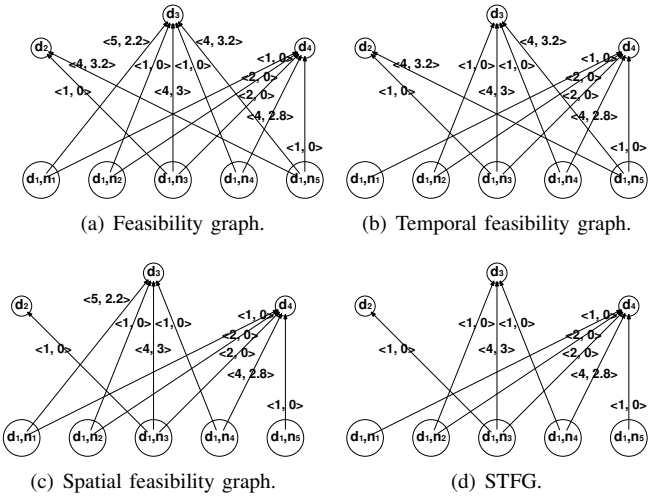


Figure 3. Feasibility graphs for the example scenario in Figs. 1 and 2. The tuple $\langle L_t, L_s \rangle$ on edges indicates temporal/spatial length; TTL=5; and STL =3.

the earliest arrival time at the destination node. In a round-trip journey, a seeker corresponds to a source/destination node, whereas providers correspond to intermediate nodes. When a seeker generates a service request for d_{end} at t_s, L_t is the time needed to complete the request, $L_t = \sum_{i=1}^{k} \{T_{ul_i} + T_{c_i}\} + T_{dl}$, where k is the length of service composition given by the number of services in a service path. T_{ul} is the time required to upload an input data to a provider involved in the service path. T_{dl} is the time required to download a final output data from the last provider to the seeker. These upload/download times are given by the summation of waiting time for the needed contact and edge traversal time for the transferred data. The edge traversal time is given by dividing the size of input/output data by the data transmission rate. T_c is the service execution time at a provider. The execution time is given by dividing the workload of service by the computation speed at a provider.

Spatial Length of Journey. Let source position be the location where a seeker generates a service request. The farthest point on the $earliest$ journey is given by the maximum distance from a source position to each node on the journey. Spatial length, L_s, is the farthest distance from a source position to the farthest point.

Feasibility graphs for the example scenario in Figs. 1 and 2 are illustrated in Fig. 3. To obtain these graphs, service requests are generated by every source node at $t_s = 0$ for a set of data excluding a start data: $d_{end} = \{d_i \in V_s \mid d_i \neq d_{sta}\}$. Fig. 3(a) depicts a feasibility graph. Fig. 3(b) is a temporal feasibility graph, where temporal length of journeys does not exceed TTL. Fig. 3(c) depicts a spatial feasibility graph, where spatial length of journeys does not exceed STL. Fig. 3(d) is a spatiotemporal feasibility graph (STFG), where temporal and spatial lengths of journeys do not exceed TTL and STL, respectively. Each edge corresponds to a pair of start and end data that can be obtained by available service composition within TTL and STL.

Spatiotemporal Feasibility Graphs (STFG). Given departure time, TTL, and STL, let $G_R^{\delta\gamma} = (V_R^{\delta\gamma}, E_R^{\delta\gamma})$ be STFG, where δ and γ represent TTL and STL. $V_R^{\delta\gamma}$ is a set of end data d_{end} and start data $\langle d_{sta}, n_{src} \rangle$, which represents a start data d_{sta} at a seeker n_{src}. A directional edge $e_{\langle d_{sta}, n_{src} \rangle, d_{end}} \in E_R^{\delta\gamma}$ corresponds to the $earliest$ journey from a start data d_{sta} at a seeker n_{src} to an end data d_{end} at departure time t_s such that for each journey, the temporal length is less than or equal to TTL, and the spatial length is less than or equal to STL. Let E_R^{inv} be the set of invalid edges whose temporal length is greater than TTL, or the spatial length is greater than STL. For the feasibility graph in Fig. 3(a), the set of invalid edges is described as $E_R^{inv} = \{e_{\langle d_1, n_1 \rangle, d_3}, e_{\langle d_1, n_5 \rangle, d_2}, e_{\langle d_1, n_5 \rangle, d_3}\}$. STFG in Fig. 3(d) is obtained by removing the set of invalid edges from the feasibility graph. STRG in Fig. 3(d) is the intersection of two graphs: the temporal feasibility graph in Fig. 3(b) and the spatial feasibility graph in Fig. 3(c).

Algorithm 1 $journey\_search(G, G_s, d_{sta}, d_{end}, src, t_s)$

Input: TVG $G = \{V, E\}$, SG $G_s = (V_s, E_s)$, start/end data $d_{sta}, d_{end} \in V_s$, a seeker src, and departure time t_s
Output: An edge $e \in E_R$, which gives spatiotemporal length for a round-trip journey

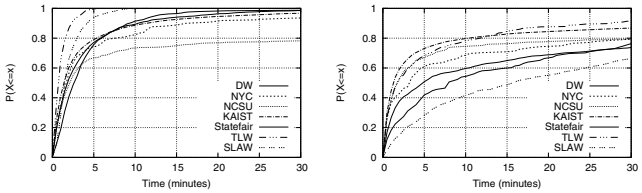
$L_t \leftarrow \infty$; $L_s \leftarrow 0$
$P_s \leftarrow service\_paths(G_s, d_{sta}, d_{end})$
for $path \in P_s$ **do**
  $t \leftarrow t_s$; $u \leftarrow src$; $d \leftarrow d_{sta}$; $dist \leftarrow 0$
  for $\langle s_{i,j}, v \rangle \in path$ **do**
    if $d \neq d_{end}$ **then** ▷ uploading
      if $u = v$ **then** ▷ local execution
        $t \leftarrow t + T_c$
      else ▷ remote execution
        $T_{ul}, D_{ul} \leftarrow earliest\_contact(G, e_{u,v}, t)$
        $t \leftarrow t + T_{ul} + T_c$
        $dist \leftarrow max(D_{ul}, dist)$
        $u \leftarrow v$
      $d \leftarrow j$
    if $d = d_{end}$ **then** ▷ downloading
      if $u \neq src$ **then**
        $T_{dl}, D_{dl} \leftarrow earliest\_contact(G, e_{u,src}, t)$
        $t \leftarrow t + T_{dl}$; $dist \leftarrow max(D_{dl}, dist)$
      if $t - t_s < L_t \vee (t - t_s = L_t \wedge dist < L_s)$ **then**
        $L_t \leftarrow t - t_s$; $L_s \leftarrow dist$
return $e \leftarrow (d_{sta}, src, d_{end}, L_t, L_s)$

Algorithm 1 details how to find a round-trip journey with spatiotemporal length. After deriving a feasibility graph from a set of edge data with spatiotemporal length, STFG is constructed by removing the set of invalid edges for any TTL and STL.

Table 1. DATASET INFO

dataset	# nodes	area size ($km \times km$)	duration (minutes)	degree	velocity (m/s)
DW	41	15×18	857	0.55	0.72
NYC	39	32×20	1360	0.67	1.06
NCSU	35	15×10	1302	1.11	0.49
KAIST	92	33×25	1400	5.08	0.58
Statefair	19	1×1	207	1.26	0.45
TLW	19	1×1	207	0.28	1.20
SLAW	19	1×1	207	0.72	0.76

- degree (velocity) is the average degree (velocity) for each node



(a) Contact time. (b) Inter-contact time.

Figure 4. Characteristics of all node pairs.

5. Analytical Framework for Service Sharing

We scope our simulations to the common scenario of mobile pedestrian networks, where mobile devices are carried by pedestrians.

Empirical Mobility Traces of Human Walks. We use the following mobility traces obtained from real-world experiments, where participants carry GPS devices [13]. The mobility traces were observed, between September 2006 - January 2007, from five different sites: Disney World theme park in Orlando (DW), New York city (NYC), North Carolina State University campus (NCSU), KAIST campus in Korea (KAIST), and North Carolina state fair (Statefair). The participants walked most of times within the radius of ten kilometers in these locations while they may also occasionally use transportation such as bus, trolley, cars, or subway trains. These traces provide us with detailed coordinates of mobile nodes at every 30-second interval.

Synthetic Mobility Models of Human Walks. We use the mobility models of Truncated Levy Walk (TLW) [14] and Self-similar Least Action Walk (SLAW) [15]. TLW generates mobility patterns for each single node, explicitly taking parameters such as distance to move, direction to follow, the amount of time to take pause, and speed of movement. Most nodes move around homogeneously in an area. SLAW produces heterogeneity of mobility patterns in an area, utilizing self-similarity in geographical dispersion of a set of locations visited by humans. Some of the locations are very common for most of people and some others are unique for each other.

Table 1 summarizes the basic information of datasets. Parameters of TLW and SLAW—such as the number of nodes, area size, and trace duration—are set to the comparable values to Statefair. Assuming that all nodes are equipped with WiFi Direct, transmission range is set to 75 meters.

For each analytic function, estimation is validated with modified datasets to examine how tolerance the estimates are with regard to future trajectory change.

Service sharing parameters are estimated based on historical mobility traces. In the future, mobility patterns could be repeated in the same mobility scenario, whereas node trajectories would not be exactly same. For example, a student's trajectory from one class room to another may vary from day to day depending on traffic and events happening in the hallways. To incorporate the trajectory change into estimation validation, original datasets are modified as follows. In the original datasets, each trace file corresponds to a node and has the following data format: time instance, x-coordinate, and y-coordinate. For every time instance, node coordinates are displaced to any point within the circle determined by the original position and a displacement parameter. The displacement parameter is set to 50, 100, 150, or 200 in meters. The percentage of node changes is set to 5, 10, 15, or 20, indicating how many nodes are modified for the original dataset.

5.1. Opportunistic Networking Properties

In opportunistic environments, mobile devices get connected with each other intermittently. Data are transferred by exploiting direct contact opportunities between two devices. Hence, the environments can be characterized by combined individual mobility and pairwise-contact patterns. Contact and inter-contact times are measured as well-known properties in opportunistic networks [16], [17].

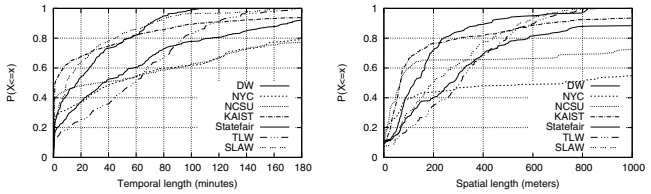
Contact Time. Contact time, also called link lifetime or link duration, is the time interval during which a given pair of nodes are in each other's transmission range. The value is useful to estimate the possible amount of transferrable data in each connection opportunity.

Inter-Contact Time. Inter-contact time is the time interval elapsed between two successive contact periods for a given pair of nodes. The value is helpful to predict the likelihood that a node will be encountered again in a given time period.

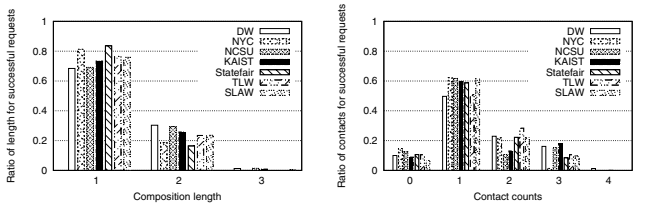
Fig. 4 shows the cumulative distributed function (CDF) of aggregate contact and inter-contact times for all pairs of nodes. We concentrate on the time range between zero to 30 minutes to clearly see the difference among mobility scenarios. Inter-contact time has more significant variation among the scenarios than contact time. In most cases, i.e., more than 50%, contact time is less than three minutes, whereas inter-contact time is less than 17 minutes. At the extreme cases, 20% of contact times for NCSU are more than 30 minutes while 30% of inter-contact times for SLAW are more than 30 minutes.

5.2. Service Scenarios

Each mobile device has heterogeneous services. A service may be offered by more than one device. For service



(a) Temporal length. (b) Spatial length.



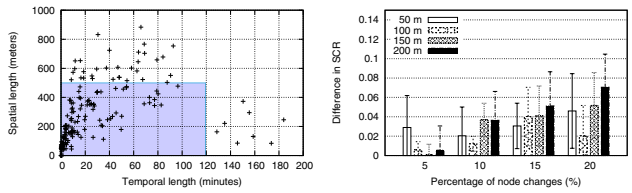
(c) Composition length. (d) Contact counts.

Figure 5. Successful service requests during time window zero to one hour.

deployment, the number of input/output data types is four. The number of unique services is 6. Each service is provided by 10% of the nodes. Let us suppose that $s_{1,2}$, $s_{2,3}$, and $s_{3,4}$ are single services such as text translation, text to speech, and time stretch (i.e., speaking speed), respectively. Other services are composite services for the above services. The average link delay is set to one second.The average execution time for $s_{1,2}$, $s_{2,3}$, $s_{3,4}$, $s_{1,3}$, $s_{1,4}$, and $s_{2,4}$ are 10, 15, 5, 25, 30, and 20 in seconds, respectively. The service execution time follows normal distribution to incorporate randomness. The distribution is determined by the average execution time and its standard deviation, which is given by 5% of average execution time. Service requests are generated as follows. First, twenty time instances are selected randomly within a time window from zero to one in hours. For each time instance, service requests are generated by random seekers for random end data d_i, for all $i \geq 2$. For every request, start data is d_1. The total number of requests for DW, NYC, NCSU, KAIST, Statefair, TLW, and SLAW are 400, 400, 363, 926, 193, 182, and 197, respectively.

5.3. Estimating Service Completion Rate

In Fig. 5(a), when TTL is three hours, the NYC trace exhibits a service completion success probability of 0.75, whereas the corresponding probability is 1.0 for Statefair, TLW, and SLAW. This is because the inter-contact time for NYC is very long, whereas Statefair, TLW, and SLAW have shorter inter-contact time. When STL is one kilometer, NYC is less than 0.6 while Statefair, TLW, and SLAW are 1.0 as shown in Fig. 5(b). NYC is a large area size while Statefair, TLW, and SLAW are small as seen in Table 1. For successful service requests, service composition length and contact counts are shown in Fig. 5. Composition length is given by the number of services in a service path, whereas contact counts are given by the number of contacts



(a) (b)

Figure 6. (a) The spatiotemporal property of nodes in SLAW, where each dot represents an available service composition and a rectangle area corresponds to the spatiotemporal frame with $TTL = 120min$ and $STL = 500m$. (b) Estimated SCR.

exploited to complete a given request. Composition length comprises at least one edge: $(s_{1,d_{end}}, v)$, where $s_{1,d_{end}}$ is a service and v is its provider. Composition length of one implies that functionality needed for a given request is provided by one service. Composition length of more than one implies composition of two or more services. 20–30% of requests are achieved by service composition, depending on the mobility scenarios. For exploiting contacts, most of contact counts are one, whereas some are zero, two, three, or four as shown in Fig. 5(d). A contact count of one implies that a service request is completed during a single contact, including service execution and input/output data forwarding. A contact count of two implies use of two providers whose services are composed, and so on. Zero contacts imply that the seeker has the service needed, resulting in its local execution. More than half of the services are achieved by exploiting just one contact.

To facilitate response to user queries, we develop an analytic function for opportunistic service sharing. The following function relates to Query 1: given TTL and STL, determine the estimated service completion rate (SCR). The function can be broadly defined as $SCR = f(t_w, TTL, STL)$, where t_w is the time window.

Spatiotemporal property of journey is captured by Algorithm 1 for building STFG. The algorithm finds successful service compositions with spatiotemporal lengths at any time window in any mobility scenario. Fig. 6(a) depicts an example of spatiotemporal property in SLAW. Each dot represents a successful service composition, which can complete a service request at the estimated temporal and spatial lengths. If the request is not completed, it is not plotted on the graph. Let $\delta \times \gamma$ STF be the spatiotemporal frame (STF), which is a rectangle area bounding the spatiotemporal length in Fig. 6(a). For each service composition within STF, temporal (spatial) length is less than or equal to TTL (STL). Service completion rate is estimated as q_1/q_2, where q_1 is the number of successful service compositions within $\delta \times \gamma$ STF and q_2 is the total number of service requests.

Fig. 6(b) shows an example of validation for estimated service completion rate (SCR). The difference in SCR values is calculated by subtracting SCR values for the projected dataset from SCR values for the original dataset. The results are the average of 10 independent simulation runs, shown

with 95% confidence intervals. The difference value tends to increase with increase in the percentage of node changes. However, the maximum difference is 0.12, which is not so large and would be acceptable for estimating service completion rate. For other datasets, the difference value is less than 0.1.

5.4. Optimizing Parameter Setting

This analytic function relates to Query 2: determine optimal TTL and STL values to achieve a given service completion rate. The function optimizes the parameter setting at the time window t_w.

In the function, the spatiotemporal frame (STF) is optimized such that the distance, denoted by $dist(O, P)$, is minimized as the shortest vector from the origin (O) to the opposite corner of $\delta \times \gamma$ STF at coordinates of the point (P). $dist(O, P)$ is calculated by,

$$dist(O, P) = \sqrt{(\frac{TTL}{L_t^{max}})^2 + (\frac{STL}{L_s^{max}})^2} \quad (1)$$

where temporal and spatial lengths are normalized with their corresponding maxima, i.e., L_t^{max} and L_s^{max}.

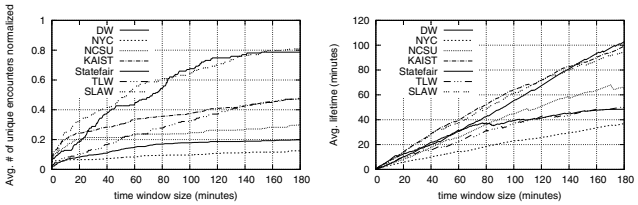
Algorithm 2 $parameter\_optimization(E_R^{\delta\gamma}, SCR^D)$

Input: a set of edges $E_R^{\delta\gamma}$, desired service completion rate SCR^D
Output: a tuple SV to be optimized by the shortest vector

```
1: SV ← (∞, ∞, ∞)
2: L_t^max, L_s^max ← max(E_R^δγ)
3: L ← min(L_t^max, L_s^max)
4: for k = 1 to L do
5:     if SCR(k, k) ≥ SCR^D then
6:         update_parameters(k, k)
7:         break
8:     for i = k to L_s^max do
9:         if SCR(k, i) ≥ SCR^D then
10:            update_parameters(k, i)
11:    for j = k to L_t^max do
12:        if SCR(j, k) ≥ SCR^D then
13:            update_parameters(j, k)
14: return SV
15: procedure update_parameters(TTL, STL)
16:    if dist(O, P) < SV[2] then
17:        SV ← (TTL, STL, dist(O, P))
```

Algorithm 2 details how to optimize parameter setting. Given a set of edges obtained by Algorithm 1 and a desired service completion rate, the algorithm returns a pair TTL–STL optimized by the shortest vector.

In the same way as the previous section, we examined the validation for optimizing parameter setting. The desired SCR is set to 0.4 at the time window zero to one in hours. The parameter setting optimized for the original dataset is applied to the projected datasets. For every dataset, the difference in SCR values is less than 0.12.



(a) Number of unique encounters. (b) Lifetime.

Figure 7. Resource size (normalized by the total number of nodes) and lifetime according to the length of time window.

5.5. Estimating Resource Utilization Time

In opportunistic environments, service resources are dynamically formed by providers that a seeker encounters over time. Service requesting, computing, and retrieving can be performed after the first contact between a seeker and a provider and before their last contact. As computational properties [6], resource size and lifetime are measured to estimate how many and how long such resources are available in the environment.

Resource Size. The size of potential resources is given by the number of nodes that a seeker encounters within a time window. The value is useful to estimate the number of potential resources for computing.

Resource Lifetime. A provider's resource lifetime is given by the time duration from its first contact to its last contact with a seeker. The resource lifetime should be within a given time window. In the worst case, a seeker can utilize a provider's whole lifetime to execute services. For example, a provider receives a service request at its first contact; then disconnects and computes the request off-line; and sends back the result to the seeker during its last contact. This value is helpful to predict the possible time duration for resource utilization.

Fig. 7 shows the resource size and lifetime. The average number of unique encounters is shown in Fig. 7(a), normalized by the total number of nodes in the network to compare the mobility scenarios. A large increase rate implies that a seeker frequently encounters resource providers and then utilizes a large pool of potential resources. In contrast, a small increase rate indicates that a seeker rarely meets providers and then exploits only a mall portion of resources. At 60 minutes, a seeker in SLAW can encounter more than half the available providers (i.e., 10 nodes), whereas a seeker in NYC encounters less than 10% of providers (i.e., 4 nodes). This is because SLAW has short inter-contact times compared to NYC. In Fig. 7(b), the average lifetime increases linearly as the window size increases. Both resource size and lifetime increase with the window size, whereas their growth rates depend on mobility scenario.

Based on the above observations, we develop the following analytic function that relates to Query 3: given a time window, estimate the possibility of successfully completing the request. The function estimates the possible resource utilization time within a certain time window. If the longest

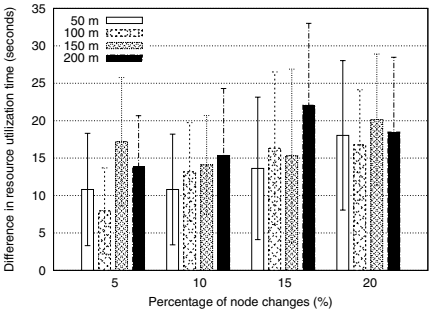


Figure 8. Estimated resource utilization time in TLW.

execution time needed by a service request is less than the resource utilization time, the whole service is likely to be executed; otherwise, the service would not be completed.

Let RS be the resource size and LT be the lifetime in a given time window. A seeker can utilize a resource supplied by its provider, where the maximum time duration is given by LT. The resource provider encounters different nodes, where the number of unique encounters is given by RS. The resource can be shared with such encounters in the time window. Thus, the possible resource utilization time is estimated as $\frac{RS}{LT}$. This is the average time duration that each seeker can utilize a resource for computing.

An example of validation for the estimated resource utilization time in TLW is shown in Fig. 8. The difference in the resource utilization time is calculated by subtracting the time for the projected dataset from the time for the original dataset. The results are the average of 10 independent simulation runs, shown with 95% confidence intervals. The difference value increases as the percentage of node changes and distance of displacement increase. The maximum difference is 33 seconds. For other datasets, the difference value is less than 30 seconds.

6. Conclusion

Opportunistic computing leveraging nearby mobile devices is expected to facilitate a number of emerging mobile applications in the near future. In this paper, we provide an analytical framework to estimate service sharing parameters with a goal to identify favorable characteristics of typical opportunistic environments for effective service composition. Spatiotemporal feasibility graphs are developed to depict spatiotemporal service sharing in opportunistic environments. The validity of the analytical framework is investigated for a wide range of empirical datasets. The proposed framework would be useful for opportunistic service sharing in any mobility scenario and enable effective service composition in opportunistic environments.

In future work we will develop mechanisms to estimate service completion rates and resource utilization times. Furthermore, a systemic overview of the analytical framework will be developed for application scenarios.

Acknowledgments

This work was completed while the first author of the paper was a post-doctoral fellow at the Rochester Institute of Technology, Rochester NY, USA.

References

[1] C. Shi, M. H. Ammar, E. W. Zegura, and M. Naik, "Computing in cirrus clouds: The challenge of intermittent connectivity," in *Proc. ACM Workshop on Mobile Cloud Computing (MCC)*, 2012, pp. 23–28. [Online]. Available: http://doi.acm.org/10.1145/2342509.2342515

[2] C. Shi, V. Lakafosis, M. H. Ammar, and E. W. Zegura, "Serendipity: Enabling remote computing among intermittently connected mobile devices," in *Proc. ACM MobiHoc*, 2012, pp. 145–154. [Online]. Available: http://doi.acm.org/10.1145/2248371.2248394

[3] D. G. Murray, E. Yoneki, J. Crowcroft, and S. Hand, "The case for crowd computing," in *Proc. ACM SIGCOMM Workshop on Networking, Systems, and Applications on Mobile Handhelds (MobiHeld)*, 2010, pp. 39–44. [Online]. Available: http://doi.acm.org/10.1145/1851322.1851334

[4] A. Passarella, M. Kumar, M. Conti, and E. Borgia, "Minimum-delay service provisioning in opportunistic networks," vol. 22, no. 8, pp. 1267–1275, Aug 2011.

[5] A. Ferrari, D. Puccinelli, and S. Giordano, "Characterization of the impact of resource availability on opportunistic computing," in *Proc. ACM Workshop on Mobile Cloud Computing (MCC)*, 2012, pp. 35–40. [Online]. Available: http://doi.acm.org/10.1145/2342509.2342517

[6] Y. Li and W. Wang, "Can mobile cloudlets support mobile applications?" in *Proc. IEEE INFOCOM*, April 2014, pp. 1060–1068.

[7] U. Sadiq, M. Kumar, A. Passarella, and M. Conti, "Service composition in opportunistic networks: A load and mobility aware solution," vol. 64, no. 8, pp. 2308–2322, Aug 2015.

[8] A. Mtibaa, A. Fahim, K. A. Harras, and M. H. Ammar, "Towards resource sharing in mobile device clouds: Power balancing across mobile devices," *SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 4, pp. 51–56, Aug. 2013. [Online]. Available: http://doi.acm.org/10.1145/2534169.2491276

[9] A. Fahim, A. Mtibaa, and K. A. Harras, "Making the case for computational offloading in mobile device clouds," in *Proc. ACM MobiCom*, 2013, pp. 203–205. [Online]. Available: http://doi.acm.org/10.1145/2500423.2504576

[10] A. Mtibaa, K. Harras, and A. Fahim, "Towards computational offloading in mobile device clouds," in *Proc. IEEE Int. Conf. Cloud Computing Technology and Science (CloudCom)*, vol. 1, Dec 2013, pp. 331–338.

[11] C. Wang, Y. Li, and D. Jin, "Mobility-assisted opportunistic computation offloading," vol. 18, no. 10, pp. 1779–1782, Oct 2014.

[12] A. Casteigts, P. Flocchini, B. Mans, and N. Santoro, "Measuring temporal lags in delay-tolerant networks," vol. 63, no. 2, pp. 397–410, Feb 2014.

[13] I. Rhee, M. Shin, S. Hong, K. Lee, S. Kim, and S. Chong, "CRAWDAD data set ncsu/mobilitymodels (v. 2009-07-23)," Downloaded from http://crawdad.org/ncsu/mobilitymodels/, Jul. 2009.

[14] I. Rhee, M. Shin, S. Hong, K. Lee, S. J. Kim, and S. Chong, "On the levy-walk nature of human mobility," vol. 19, no. 3, pp. 630–643, June 2011.

[15] K. Lee, S. Hong, S. J. Kim, I. Rhee, and S. Chong, "SLAW: Self-similar least-action human walk," vol. 20, no. 2, pp. 515–529, April 2012.

[16] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, "Impact of human mobility on opportunistic forwarding algorithms," vol. 6, no. 6, pp. 606–620, June 2007.

[17] T. Karagiannis, J.-Y. Le Boudec, and M. Vojnović, "Power law and exponential decay of intercontact times between mobile devices," vol. 9, no. 10, pp. 1377–1390, Oct 2010.