COSC: Paths with Combined Optimal Stability and Capacity in Opportunistic Networks

Shiraz Qayyum
The MathWorks Inc.
3 Apple Hill Dr.
Natick, MA, USA
Email: shiraz.qayyum@mathworks.com

Peizhao Hu
Golisano College of Computing and Information Sciences
Rochester Institute of Technology
Email: ph@cs.rit.edu

Mohan Kumar
Golisano College of Computing and Information Sciences
Rochester Institute of Technology
Email: mjkvcs@rit.edu

***Abstract*—Opportunistic networks are characterized by the dynamic connectivity created when mobile devices encounter each other, as they are within close proximity. During these transient opportunities, devices are typically within one-hop wireless range of their neighbors. Opportunistic networks are an effective way, in terms of bandwidth and battery consumption to distribute large volume content among peers. Many existing proposals consider opportunistic networks as a best-effort content delivery approach, which limits their applications. We exploit characteristics of human mobility to derive an effective data forwarding scheme that achieves Combined Optimal Stability and Capacity (COSC) for opportunistic networks. COSC includes a path selection algorithm to maximize the utility of link capacity and stability. We validate theoretical findings with rigorous simulation studies using synthetic and real-world mobility traces. When compared with other approaches, COSC shows significant improvement due to the consideration of link capacity and stability.**

I. Introduction

A recent Cisco [17] report claimed 18 exabytes of mobile traffic in the cellular network in 2013, an increase of 81% compared to that in 2012. For years, cellular network providers have been looking into alternative solutions to off-load mobile data traffic. Femtocells, WiFi hotspots and opportunistic networking are among the popular options [19]. Although opportunistic networking is considered as a potential solution given the rapid adoption of mobile devices, there are a number of challenges related to its dynamic property in connectivity [6]. Typical *store-and-forward* solutions consider opportunistic networking as a way of best-effort content delivery, should an end-to-end path disappear. Several novel approaches proposed in the literature exploit *inter-contact times* and node mobilities to ensure packet delivery [18]. However, there is a need for further investigations to reduce delays and overheads in opportunistic networks (ONs).

We explore the idea of exploiting human mobility patterns to develop an effective algorithm for data forwarding in opportunistic networks. As users move around with their mobile devices, clusters of opportunistic networks are formed periodically among different social encounters. Users within these clusters can utilize the transient connectivity to distribute bandwidth intensive contents (prefetched video) among peers. Research works on human behavior and social connectivity have discovered that *humans follow a high degree of spatial and temporal regularity in their movements* [9] [24]. Moreover, studies by Karut et al., have shown that proximal users exhibit similar interests and proximity forges common interests. Keller et al., also present scenario wherein collocated users are interested in a same video [13] [12]. In this paper, we propose a novel scheme that achieves combined optimal stability and capacity (COSC)[1] in opportunistic networks. COSC exploits repeating trajectories embedded in the mobility histories of participating nodes. While our previous work [21] estimates contact volume that can be transferred during an *opportunistic contact*, the work presented in this paper estimates capacity of an *opportunistic path.*

A time varying graph (TVG) [4] is an instant snapshot of clusters of connected components in ONs. A TVG represents the connectivities between nodes over a series of discrete time intervals. In other words, TVG captures the pair-wise connectivity of any two nodes at different timestamps. If we relate this to the human mobility model, we can map devices' connectivities reoccurrence (as their users encounter each other) to a TVG over a period of time. At the start, mobile devices can retrieve the initial mobility graph, which is relevant to them, from a server. At run time, when users are moving around, their devices encounter each other. Information about such contacts at runtime can be used to verify and update initial mobility graphs. When a node needs to forward data, it searches for matching paths (containing its immediate neighbors) that lead to the desired destination. It is assumed that human mobility follows repeating trajectories, this predictive path lookup can reduce significant amount of overhead messages and result in packet delivery improvement. In COSC, each edge (or link) weight between two nodes of the TVG corresponds to the contact volume [21] when the two nodes come into contact. The effective capacity of a path from a source to a destination is the capacity of the bottleneck link. In addition to maximizing path capacity, it is also desirable to minimize the degradation in packet delivery due to path changes over a TVG. The utility of a path is a function of the path capacity and its stability. In effect, we solve an optimization problem that compares alternative paths

[1]The work was completed as part of the doctoral degree of the first author at the Rochester Institute of Technology

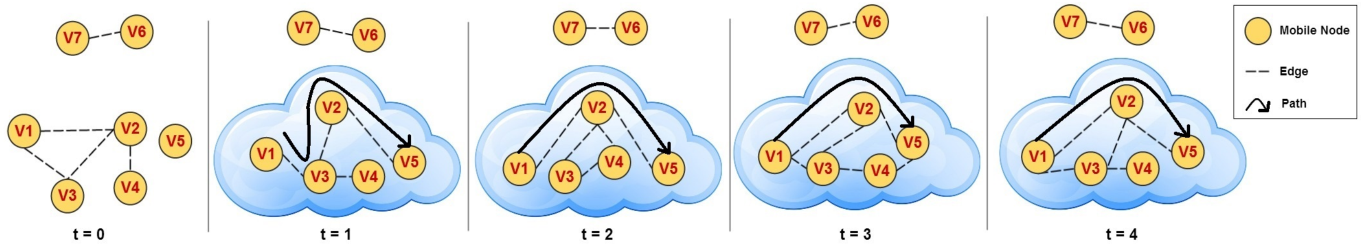


Figure 1: A TVG representation of an opportunistic network where one connected cluster $C_1, ...C_4$ comprising $\{v_1, ..., v_5\}$ is highlighted, with $a = 1$ and $b = 4$ and $a, b \in \tau[0, 4]$.. An example of a path set is also depicted, with a single changeover between $t = 1$ and $t = 2$.

in terms of their utilities over a series of snapshots of a TVG.

This paper makes the following contributions:

1) Utilizes mobility histories to identify suitable path for effective data transfer;
2) Proposes a path-aware adaptive solution;
3) Utilizes capacity and stability metrics to evaluate available paths; and
4) Performance evaluation with synthetic and real-world mobility traces to demonstrate the feasibility of solution.

II. System Design

In this section, we discuss three aspects of our system design: (i) a graph theoretic model, (ii) connected clusters within the TVGs, and (iii) synthetic mobility models that allow more rigor performance evaluation.

A. Graph theoretic model

Dynamic environment of ONs can be viewed as edges appearing and disappearing in a time varying graph (TVG). Recently, Casteigts et al., have presented a unified model of time varying graphs [4]. We modify their general definition in this subsection to model ONs. Let the set of total nodes in the environment be V and let $E \subseteq V \times V$ represent the set of edges. Events, such as, inclusion and exclusion of edges happen over time $\tau \subseteq T$, where $T = \mathbb{N}$, i.e., discretized temporal domain. This work assumes the starting value of τ to be zero. ρ is the presence function such that $\rho : E \times \tau \rightarrow \{0, 1\}$. It represents presence of an edge at a given time - presence (absence) of an edge is represented by '1' (0). Therefore a TVG is represented as a tuple $G = (V, E, \tau, \rho)$. Figure 1 shows an example of a TVG, with seven nodes and τ in the range $[0, 4]$.

Edge computation in presence function: When two nodes v_x, $v_y \in V$ come within each others' transmission range D, they can form a communication link (edge in a TVG), i.e., when the predicate $\left\| pos(v_x) - pos(v_y) \right\| \leq D$ is true, where $pos(.)$ represents two dimensional position vector of a node. Therefore, the presence function ρ evaluates the aforementioned predicate at times $t \in \tau$ to determine if an edge is present between v_x and v_y in a TVG. The radio transmission link may be established using either Bluetooth or Wi-Fi Direct, which have their own characteristic D [1]. However, the experimental evaluation uses communication ranges of 100 and 150 meters, which are consistent with the ns-2 default values.

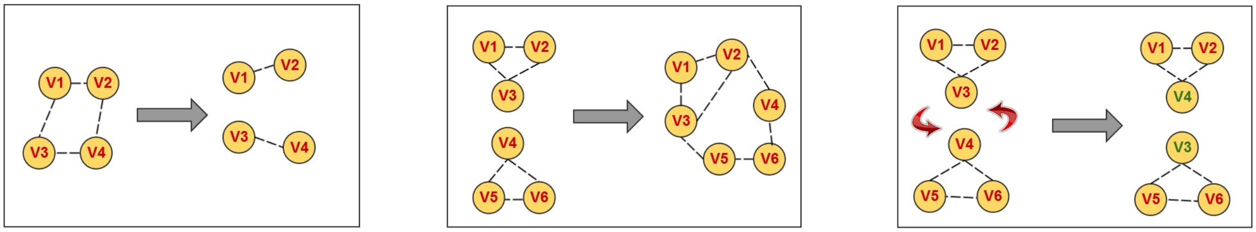
B. Time varying connected clusters

Zooming in on any one snapshot of a TVG ($t \in \tau$) reveals a single graph, that may or may not be connected. Though disconnectedness is not uncommon in snapshots derived from opportunistic networks, it is however easy to find connected clusters of nodes in it. Moreover, environments that have high node density are much more likely to contain such connected clusters, e.g., at a busy train station or a bus ride, wherein the nodes may be interested in sharing content such as a prefetched video.

A cluster is essentially a subgraph in its parent TVG. A cluster retains its connectivity over a period of time, despite changes in the set of edges [15]. The lifetime of any given cluster is marked by starting and ending times $a, b \in \tau$. Consider one such cluster comprising a subset of mobile nodes $V' \subseteq V$. At any one starting point $a \in \tau$, there is an edge set E_a among these nodes. As the nodes move in time, the set of edges changes from E_a to E_{a+1}, then to E_{a+2} and so forth to an edge set E_b. The graph representing such a connected, time varying cluster comprising vertices V', but with varying edges is represented by $C_i(V', E_i)$ for $i = a, a+1, ..., b$. Figure 1 highlights one such cluster for $a = 1$ and $b = 4$.

The time stamp $b \in \tau$ and the edge set E_b marks the time after which a cluster comprising $V' \in V$ set of nodes fails to stay connected. Essentially, this happens whenever the connected set V' changes. The following three scenarios elucidate this event:

1) A cluster breaks down into smaller clusters, thus ending its lifetime and creating new ones as shown in Figure 2a.
2) Two or more clusters merge together to form a bigger cluster as shown in Figure 2b. This case ends the lifetime of multiple clusters to start a single new cluster.
3) A cluster may lose and gain equal number of nodes till the next timestamp as shown in Figure 2c. Though such



(a) A cluster breaking into two; marks the end of one cluster and creation of two.

(b) Two clusters merging into one bigger cluster; ends two clusters and starts a single new cluster.

(c) The set of nodes changes, though the size is maintained; ends two clusters and starts two new clusters.

Figure 2: The three scenarios that mark the lifetime of a cluster

transformation preserves the number of nodes in any given cluster, it still changes the identity of the original set of connected nodes.

C. Synthetic Mobility Models

Human mobility patterns are key to understanding information flow in ONs. Fortunately, a number of human mobility traces have been collected in environments like, campuses, conferences, state fairs etc., that can be used in simulations pertaining to opportunistic networks [16] [11]. To bring more rigor to performance evaluation, synthetic mobility models are considered as well. Using traces generated with synthetic mobility models allow us to study the proposed solution in a variety of other scenarios.

There have been numerous efforts in determining the underlying common and stationary features of human mobility [5] [24]. It has been shown that the inter contact times in real-world mobility traces follow a power law distribution [5] as opposed to previously assumed exponential distribution. Moreover, it is also shown that various other statistical factors in human mobility, such as, flight lengths, pause times and fractal way-points follow power law distribution as well. Hence, to make the simulation results descriptive of real-world phenomena, we use self similar action walk (SLAW) as one of the synthetic mobility models [14] to generate movement scenarios. We have also used Home-cell Community-based Mobility Model (HCMM) as another synthetic mobility model [2] for trace generation. HCMM combines three main properties of human motion: 1) human movement is governed by social interactions; 2) users visit a few locations where majority of their time is spent; and 3) users prefer shorter paths over longer ones.

III. Path Selection

In this section, we deal with the design choices in order to find paths that enhance the information flow between a source and destination pair over a series of time stamps. Let P_i denote a multi hop (can be direct as well) path between two nodes of interest, in the time varying cluster C_i, for $i = a, a+1, a+2, ..., b-1, b$. The collection of these paths is also termed as a path set in this paper. In order to find the most effective path set between a pair of connected nodes, we consider path capacity and path stability. Our objectives include:

1) Maximizing capacity between connected nodes; and
2) Maintaining a stable path.

COSC finds clusters in ONs and employs a utility function that considers path capacity and path stability to determine a path. A polynomial time, dynamic programming algorithm efficiently computes such paths in time varying connected clusters by maximizing the utility function.

A. Link capacities

When two mobile nodes v_x and v_y make contact for a duration of t^0_{xy}, the amount of data can be transferred depends on their velocity vectors [21]. Maximum amount of data that can be transferred between a pair of opportunistically meeting nodes v_x and v_y is given by $K(v_x, v_y)$.

In wireless communications, the received signal power S_{rec} at the receiver varies inversely with the distance d from the sender. The relationship is concretely represented as $S_{rec} \propto 1/d^{\psi}$, where the exponent ψ depends on the environment in which the nodes operate. It is also known that the transmission throughput depends on S_{rec}, making the throughput a function of instantaneous distance [8]. Our model [21] accounts for the changing distance and hence variable throughput during the contact period. It is interesting to note that an empirically obtained function of throughput against distance can be transformed into a time dependent function, by expressing the distance in terms of time for the moving nodes. If the time dependent throughput between v_i and v_j is denoted by $R(t)$ then the following equation couples it with the contact volume and contact duration:

$$K(v_x, v_y) = \int_0^{t^0_{ij}} R(t)\, dt. \quad (1)$$

The model in this paper uses the contact volume $K(v_x, v_y)$ as the edge weight for the single hop connection between v_x and $v_y \in V'$.

1) Contact volumes on a path: Let e^i_k denote a single edge on a path P_i in an edge set E_i for cluster C_i. The two nodes forming the direct edge e^i_k, are represented by v_k, $v_{k+1} \in V'$. Then the edge weight is given by

$$w(e^i_k) = K^i(v_k, v_{k+1}). \quad (2)$$

The superscript i in the contact volume definition, represents value in the i^{th} time stamp. Hence the total effective capacity of a path $\xi(P_i)$ is the minimum of all the edge weights that make up that path,

$$\xi(P_i) = \min w(e_k^i), \quad \forall\ e_k^i \in P_i \tag{3}$$

As discussed at the start of this section, in order to improve the data transfers, it is desirable to look for paths that have a higher capacity. $\xi(P_i)$ is the precise mathematical quantity that our scheme tries to enhance.

B. Stability of paths

At the start of Section III, we define P_i as a path in C_i for $i = a, a{+}1, a{+}2, ..., b{-}1, b$, then let $\psi(P_a, P_{a+1}, ..., P_{b-1}, P_b)$ represent the total number of changes, i.e., the points at which the identity of the path switches from the one in the previous instance of the subgraph. Formally, it is the number of indices i $(a \leq i \leq b-1)$ for which $P_i \neq P_{i+1}$. For example, considering the highlighted cluster in Figure 1, a chosen path $P_1 \in C_1$ from v_1 to v_5 is $(v_1 \rightarrow v_3 \rightarrow v_2 \rightarrow v_5)$, which is different from $P_2 = P_3 = P_4$ = $(v_1 \rightarrow v_2 \rightarrow v_5)$. Therefore, $\psi(P_1, P_2, P_3, P_4) = \psi(P_1, P_2, P_2, P_2) = 1$, as there is only one effective changeover in the path set.

In order to bring stability in the chosen path, it is desirable that it remains constant for as long as possible, despite the network dynamics. There are a number of reasons for such a desideratum:

1) *Overheads due to routing table updates:* A path switch requires exchange of messages to update routing tables, as nodes in a cluster, maintain routing information. This may also result in path oscillations [10].
2) *Data quality:* A switch to a new path entails, stalling the flow in the previous path and shifting the flow to the new path. This results in overheads, such as flushing buffers, and closing sockets, readers and writers, leading to poor data quality.
3) *Trust:* Protocols often need to establish trust whenever, new nodes are used for exchanging data, a costly procedure.

Therefore, in order to avoid the aforementioned penalties, a network designer should try to find paths that do not switch very often, i.e., minimize $\psi(.)$.

C. Finding a path set

Earlier, we identified two objectives for choosing paths in a connected cluster and modeled them quantitatively. We put them together using the following utility function,

$$utility(P_a, ..., P_b) = \sum_{i=a}^{b} \xi(P_i) - \theta \times \psi(P_a, ..., P_b) \tag{4}$$

$utility(P_a, ..., P_b)$ gives the total utility of the paths that are selected during the lifetime of a time varying cluster. Note that, the first term $\sum_{i=a}^{b} \xi(P_i)$ on the right side of Eq. 4 is the total sum of effective capacities all the paths, where each effective capacity of a path is the contact volume of the bottleneck edge making up the path. The second term gives the cost associated with path instability. The tuning constant θ can be used to weigh the second term according to network properties and application requirements. The unit of θ is MB and it is a quantity that signifies the amount of capacity that is equivalent to a single change over in the path set. A higher value of this quantity would suggest that every change over in the path set will incur a high penalty. The overall objective of this scheme is to maximize the utility given in by the Eq. 4.

It is possible to find connected components in linear time in a given snapshot of a time varying graph, i.e., at any $t \in \tau$ by using breadth first search or depth first search. Where the common technique of an outer loop can be used to cover all nodes in V.

In order to compute a path set of maximum utility for a time varying cluster, it is helpful to look at the available options for adding the final P_b. The following are the two options:

1) Choose P_{b-1} as the final path, which will add $\xi(P_{b-1})$ to the total utility, but will avoid the change penalty θ.
2) Choose a new path, that exists in C_b. It is natural to look for a path with maximum effective capacity, call it $P_{best} = P_b$. This path adds $\xi(P_b) - \theta$ to the final utility.

It is desirable to choose a path P_{b-1} in C_{b-1} that is also present in C_b, i.e., $P_b = P_{b-1}$. This will avoid the change penalty θ. The effect of C_b on the earlier part of the solution can be anticipated using dynamic programming.

Let $Opt(i)$ denote the solution to the subproblems for the clusters $C_a, ..., C_i$. To compute $Opt(b)$, i.e., a solution for the complete time varying connected cluster at hand, one should look for the last changeover that occurred in the path. Let the last changeover be between C_i and C_{i+1}. This means we have a path P_{i+1} in all the clusters $C_i, ..., C_b$. Therefore, the edges of P_{i+1} are present in all of those clusters. Let $common(C_i, ...C_j)$ represent a graph that is an intersection of all the clusters $C_i, ...C_j$ and let the best path in such a common graph be $P_{best}(i, j)$ for $a \leq i \leq j \leq b$. Then, if the last changeover occurred between the indices i and $i + 1$, the recurrence relation for the optimal utility can be expressed as $Opt(b) = Opt(i) + (b - i) \times \xi(P_{best}(i + 1, b)) - \theta$. However, there is a special case that may exist for finding a single path for the entire duration of a cluster that incurs no changes i.e., $\psi(.) = 0$. In this case $Opt(b) = (b - a + 1) \times \xi(P_{best}(a, b))$. Hence the final recurrence that gives maximum utility, set of paths and guarantees the minimum number of packets transferred in a cluster is as follows,

$$\begin{aligned} Opt(b) = \max\{&(b - a + 1) \times \xi(P_{best}(a, b)), \\ &\max_{a \leq i \leq b} (Opt(i) + (b - i) \times \xi(P_{best}(i + 1, b)) - \theta)\} \end{aligned}$$

The algorithm presented above first computes for each pair i, j, the $common(C_i, ...C_j)$ and $P_{best}(i, j)$ values for $a \leq i \leq j \leq b$. There are $O((b - a)^2)$ such pairs and the complexity to compute each subgraph is $O(|V|^2 (b - a)^2)$, where $|V|$ is the number of nodes in a cluster. The factor of $|V|^2$ arises because of the maximum number of possible edges in a cluster. A simple linear search is employed to compute the best path in each graph in linear time. Therefore the total running time of the algorithm is $O(|V|^2 (b - a)^3)$. The algorithm can be

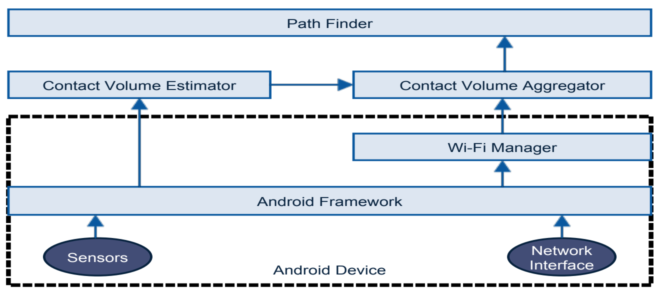


Figure 3: System Architecture

speeded up by computing the graphs $common(C_i, ...C_j)$ and $P_{best}(i, j)$ for a fixed value of i in order of $j = i, ..., b$. Then the total polynomial running time of the proposed algorithm is reduced to $O(|V|^2 (b-a)^2)$.

IV. System Architecture and Implementation

In order to implement a software agent that may execute our scheme and compute a path set of maximum utility, we present a system architecture developed for Android devices. Figure 3 depicts the proposed modular design consisting of two major components.

We make use of Android system services to get instances of Wi-Fi P2P manager and other sensor controllers. The Wi-Fi P2P manager is responsible for discovering and setting up connection to nearby neighboring devices to exchange control information, contact volume estimates and data streams. Sensors such as accelerometer, gyroscope and GPS feed information to the contact volume estimator. All of the sensor data is made available through system calls executed through the Android framework.

A. Contact volume estimator and aggregator

The contact volume estimator receives sensor data such as device's accelerometer readings, along with GPS information to estimate the contact volume between neighboring devices. The basic idea hinges on the fact that, two mobile devices have a high contact volume if the contact duration is long. The estimator predicts the contact volume based on the data rate profile as well as the device's estimated contact duration inferred from its velocity and initial position vectors.

The device's own contact volume estimation is compared against the one received from neighboring devices and thus aggregated using a pessimistic approach, i.e, the minimum of the two contact volumes is chosen as a final estimate for further computation. It is possible for the two devices to predict slightly different contact volumes, as it is not a symmetric measure [21]. Formally a device v_x computes contact volumes $K(v_x, v_i)\ \ \forall v_i \neq v_x$, where $v_i \in V'$ is an immediate neighbor of v_x. Similarly, the neighboring devices estimate their contact volumes for device v_x, i.e., $K(v_i, v_x)$ and send this control information to v_x. Subsequently v_x chooses $\min(K(v_x, v_i), K(v_i, v_x))$ for all neighboring v_i. Also, note that the neighboring devices of v_x are responsible for sending contact volume information pertaining nodes that are not in direct connection with v_x.

B. Path finder

The path finder module is responsible for carrying out all the necessary tasks to compute a maximum utility path set based on the mobility history of the users comprising the opportunistic network. Figure 4 shows a flow chart that describes all the important processes carried out by the path finder module. Initially the device downloads a relevant mobility history data trace and preprocesses it. The preprocessing step ensures cleansing of the data and turns it into a usable form. For example, mobility history data traces can be in the form of waypoints, location coordinates or mere device to device contacts. As all of these forms of data have associated time stamps, preprocessing helps in building a TVG.

The path finder periodically acquires information about its direct one hop neighbors through a simple device discovery process. Knowledge of direct one hop neighbors at runtime helps in partially verifying the TVG and saves unnecessary computation.

The node can verify whether its one hop neighbors depicted in the TVG are also its one hop neighbors at runtime, thereby preventing computation of false paths early on in the path finding process. If the one hop neighbors do not match with the ones in the TVG, those nodes can be removed along with their edges before the path computation. Moreover, the history is updated to reflect a more recent view of the network.

A high level application, that is not part of the path finder module provides the destination node and tuning parameter θ to appropriately penalize the change overs in a path set. The path finder then computes a path set based on the processed TVG, to maximize the utility for the given destination node. Though repetitive human movement patterns favor this approach and guarantee a high hit rate, there is still a chance, that the computed paths based on mobility history may be nonexistent at runtime. As humans sometimes do deviate from their usual movement patterns, COSC verifies once more whether the computed path set is pertinent. If it is, then the computed path set is deemed fit and subsequently used for data transfers. On the contrary, our scheme adapts and falls back to epidemic routing with anti-packets [31], wherein, the source *infects* its immediate neighbors with the data packets; the neighbors then send the data packets to their next hop neighbors. This process of data packet infection continues till the destination receives the data, at which point, it sends back an anti-packet as an acknowledgment. All the nodes that are earlier infected, use the anti-packets to purge the corresponding original data packets from their buffers. Finally, COSC has a corrective mechanism and in the case when epidemic routing is invoked, it learns the new movement patterns and updates the original movement history.

V. Evaluation

This section evaluates the performance of COSC. First, the distribution of connected time varying clusters is presented.

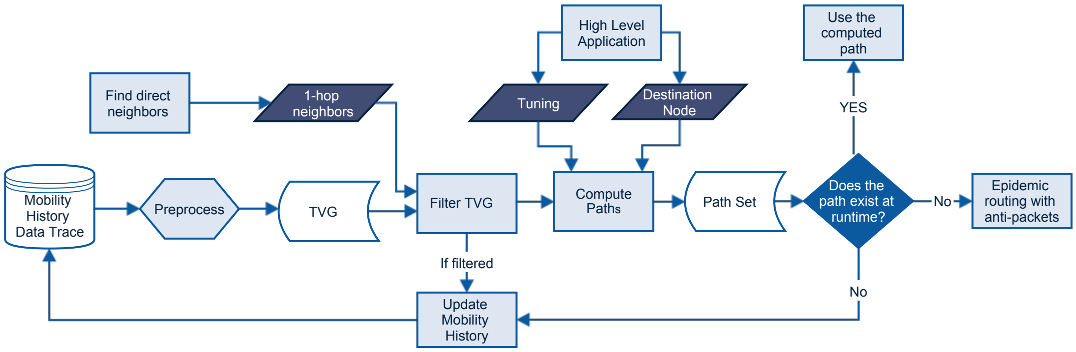


Figure 4: Path finder's flowchart. The tuning parameter is provided by a high level application. The path finder adaptively switches between using maximum utility path set and epidemic routing with anti-packets.

We then evaluate the total information flow that is possible in opportunistic networks. Finally, based on information flow results, we compare COSC with shortest multi hop paths and investigate the improvement achieved by COSC for file transfer failures.

A. Simulation setup

For the purpose of this simulation, both real-world and synthetic mobility traces are considered. The real-world mobility traces were contributed by two groups of volunteers who visited a State Fair and Disney Land in Orlando respectively [22]. The positions of the individuals were logged at 30 second intervals for a duration of approximately three and twelve hours. Therefore the positions of all the users at each time instance (every 30 seconds) is referred as a snapshot. There are a total 19 and 41 log files for State Fair and Disney Land traces respectively, where the i^{th} log file represents the position of that user in two dimensional space for the entire duration of the collected trace. In order to simulate a user-based opportunistic network, we assume, that all log files represent users carrying mobile devices that are capable of making connections with each other if they come within the radio communication link.

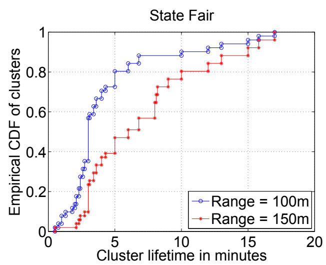
Apart from the real-world mobility traces, we use Self-Similar Least Action Walk (SLAW) mobility model for generating additional scenarios that captures the statistical human mobility characteristics. We span the simulation area to a $500m \times 500m$ two dimensional space, and make use of both slow and fast moving nodes (10 each). The details of this synthetic mobility trace are provided in Table I. Furthermore, we have also used the HCMM mobility model to test our scheme with 20 nodes [2]. The details of parameters used to setup the HCMM trace are presented in Table II We have used 100 m and 150 m as the two radio communication ranges. The specific values are used to depict ranges of Wi-Fi Direct [1], which is a prevalent technology in the smart phones and mobile devices of today.

B. Connectivity of time varying clusters

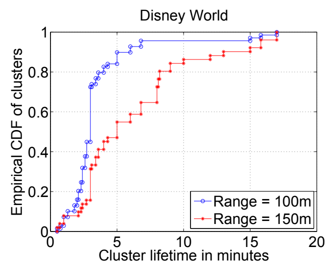
In order to find suitable path sets for enhanced information flow in a number of applications, there is a strong need to find connected components in several human mobility scenarios. With the aforementioned setup in Section V-A, we ran depth first search on each of the snapshots of the time varying graphs to first find the individual connected components in linear time. Then in order to find clusters that adhere to the definition provided in this paper, i.e., follow the lifetime rules of a cluster, each of them was backtracked till the identity of the nodes did not change. As the traces have been generated with 30 $secs$ time intervals, we did not include clusters, that just remained connected for a single snapshot of a time varying graph. Figure 5 shows the empirical CDF plots obtained for State Fair, Disney World, SLAW and HCMM mobility traces. The plots, depict the CDF of the life time of various clusters. It is observed that, for a communication range of $100m$, the real-world traces contained most of the clusters that remained alive for approximately 3 $minutes$. As approximately 65% of the clusters disintegrated after this time interval. However, the SLAW mobility trace shows a much more transient behavior, where approximately 80% of the clusters remained connected for less than 3 $minutes$. We believe it is because of the inclusion of 10 fast moving nodes, which are highly dynamic and thus, disrupt the connectivity among other connected nodes. Note that, the definition of time varying connected cluster presented in this work is strict, as even an inclusion of a single node to an already connected cluster, starts the lifetime of a new cluster and ends that of an earlier one. Data trace based on HCMM showed some clusters that were alive for up to 30 minutes. This is probably true because of the cell based communities present in the trace. For a communication range

Table I: SLAW mobility trace used for simulation

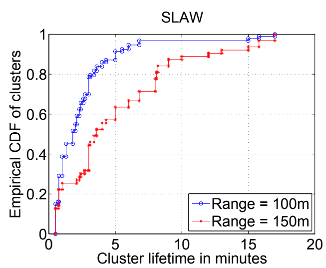
Description	Slow	Fast
Exponent of step length distribution	1.6	
Exponent of pause time distribution	0.6	
Hurst Parameter	0.75	
Velocity of node	1m/s	5m/s
Minimum step length	5m	25m
Minimum pause duration	30s	7.5s
Maximum pause duration	600s	60s
Simulation area	500×500m	



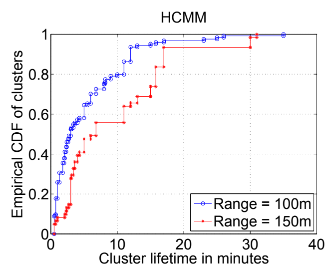
(a) State Fair - Longer communication range produces clusters with longer lifetimes.



(b) Disney Land



(c) SLAW - Short lived clusters are most prevalent



(d) HCMM - Clusters living for over 30 mins found as well

Figure 5: Empirical CDF of the lifetime of clusters in different models

of $150\ m$, the clusters are observed to remain working for longer time intervals. However, in the SLAW mobility trace, we do not see the effect as pronounced as in the real-world or HCMM traces as 60% of the clusters remained alive for less than $5\ minutes$.

Table II: HCMM mobility trace used for simulation

Parameter	Value
Velocity of node	min: 1m/s, max: 5m/s
Nodes	20
Groups	5
Rewiring probability	0.2
Travelers	5
Grid	4×4
Simulation area	500×500m

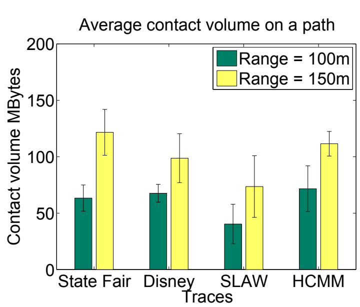
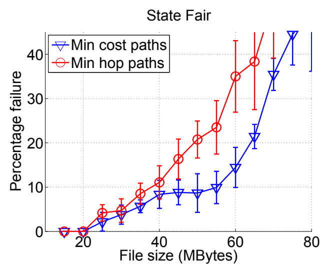


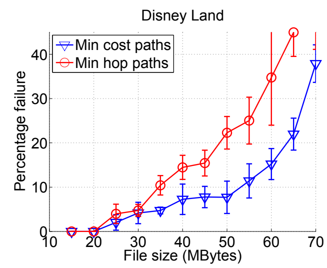
Figure 6: Average contact volume on a path set between arbitrarily chosen nodes in clusters. Longer communication range opens up the opportunity for transferring more data.

C. *Possible data transfer*

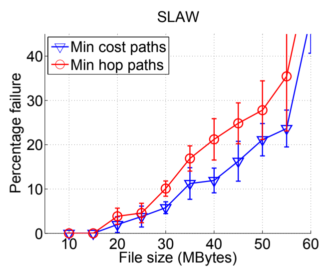
To quantify the possible data transfers between nodes in a cluster, we randomly select a pair of nodes whenever a cluster is formed. Based on COSC, we select a path set for that particular pair and evaluate the contact volume of each path. The contact volume of a single path is chosen to be the one that makes up the weakest link between the two end nodes. Its value is valid for a single snapshot of the cluster at hand. Finally, the total contact volume for the entire path set is computed by summing individual minimum contact volumes. For example, back in Figure 1 we found a path set $\{P_1, P_2, P_3, P_4\}$. Suppose the contact volumes of the weakest links in each of these paths are $10, 20, 20$ and $10\ MB$ respectively, where each path is valid for $30\ secs$ interval, then the total contact volume will be the sum of these quantities, i.e. $60\ MB$. Note that, though the paths P_2, P_3, P_4 essentially represent the same path, the contact volume of the weakest link maybe different, even when the contact durations are for $30\ secs$. The reason for that was investigated in our previous work [21], where it is shown that the contact volume not only depends on contact durations, but also on instantaneous distance among communicating nodes. The simulation is repeated 1000 times, and the averages are plotted in Figure 6, for three mobility traces. The error bars represent the standard deviation. It is observed that an average of approximately $125\ MB$ can flow at best in a State Fair setting with $150\ m$ radio link. However, SLAW shows the minimum average contact volumes on a path set, with $45\ MB$ at a $100\ m$ range. HCMM showed the least percentage change in the contact volume of a path, when the radio range is



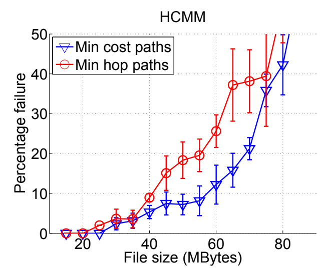
(a) State Fair - The difference in the file transfer failure rates are most pronounced around file sizes close to the contact volume, i.e., 55 MB



(b) Disney Land - The maximum difference in failure rates is seen around 60 MB files



(c) SLAW - Failure rates climb up fairly quickly after the possible contact volumes. Failure rates of min cost path are 40% less at best.



(d) HCMM - Failure rate is lesser compared to other traces at 70 MB file size.

Figure 7: Percentage of file transfer failure rates at communication range of 100 m. A comparison of COSC against shortest paths computed based on number of hops

increased from 100 m to 150 m.

D. File transfer failure rates

In order to compare the maximum utility path set, obtained using the methodology described in this paper, against the well known shortest multi hop path set, we use file transfer failure rates as a measure. The shortest multi hop path set is formally defined as a path set in a time varying connected cluster, where all link costs are assumed to be positive and equal to 1. Thus a shortest multi hop path set, contains the shortest paths in terms of hop count for each snapshot of a cluster. For the purposes of this study, we ran the simulation a 1000 times and the average file transfer failure rates were plotted against file sizes. The error bars represent the standard deviation. Failure rate is defined as the percent of file transfer failures. In other words, failure rate is given by number of *(failed transfers) / (total number of attempted transfers)*. The range of file sizes is chosen based on results shown in Figure 6, i.e., file sizes were picked in the proximity of the maximum possible amount of data transferrable for each of the three traces at 100 m range. The rationale for doing so is explained below. For example, if huge file sizes are used in comparison to the found data limits, it would invariably lead to failed file transfers. On the other hand, using file sizes that are very small, will let them through almost all the time. Therefore, both the extremes do not help in revealing any interesting information about our methodology. Figure 7 depicts the comparison of failure rates between COSC against the scheme that uses shortest multi hop path. Though we compared the two schemes for both 150 m and 100 m communication ranges, the plots are drawn only for 100 m, because higher ranges reveal similar trends, but around larger file sizes. For real-world mobility traces, it is observed that the file transfer failure rate of COSC is approximately 60% *less* than that of shortest multi hop for file sizes around 55 MB. Whereas, the advantage in SLAW mobility trace is almost 40% at best. Moreover, HCMM shows the least failure rate for 70 MB file sizes, where the other three environments have a 30% or more failure rate.

VI. Related Work

One of the earliest works on data transfers in opportunistic networks falls under the category of routing. There are schemes such as single copy [26] and multiple copy [25] that try to improve message delivery ratio while making a trade-off between end-to-end delays. Sadiq et al., presented a method that utilizes nodes' diffusion and proximity to improve both delivery ratio and delays [23]. The fastest method to perform routing in intermittently connected mobile networks is epidemic routing, i.e., to flood the message throughout the network [28]. However, this has obvious drawbacks, such as overflowing buffers and significant battery consumption on resource constrained mobile devices. Though routing in essence is related to our work, it generally deals with messages that are short and do not require considerable contact volume.

Data dissemination has been studied for conventional Mobile Ad Hoc Networks (MANETs) [30] In general, these systems assume that network paths are rather stable and in some cases generate significant amount of traffic just to maintain knowledge of other nodes' caches. Therefore, they are not suitable for user-based opportunistic networks. Conti et al., define how a semantic representation of information can be used to determine relevance of information to be exchanged in ONs [7]. Pietilanen et al., approach the problem of data dissemination from a social networking point of view [20].

Xuan et al. [29] have proposed algorithms for calculating shortest (in number of hops), fastest and foremost (i.e., earliest arrival) paths in TVGs. Casteigts et al. [3], study deterministic computations to broadcast data where edges in a TVG do disappear, but appear infinitely often. Unlike COSC, these algorithms do not take into account the costs associated with the links. Link capacity can also be estimated by using other approaches, e.g., SINR profile lookup [27].

VII. Conclusions

This paper presents a methodology that exploits mobility histories for identifying paths for effective data transfers in user-based opportunistic networks. The mobility history data traces are preprocessed to produce time varying graphs. COSC is adaptive and is able to recover when the computed paths differ from the ones that are available at runtime. To the best of our knowledge, this is the first scheme to estimate capacity over stable paths in opportunistic networks. We present a complete and thorough definition of time varying connected clusters and define scenarios for their lifetimes. In order to find source-destination paths that maximize contact volume and are stable, we utilize capacity and stability metrics. We then present a polynomial time algorithm to solve the resultant utility function. The simulation results show that using COSC, nodes can transfer from $45\ MB$ to $120\ MB$ of information in an opportunistic environment. Furthermore, a reduction of up to 60% in file transfer failure rate is seen in comparison to shortest multi hop path scheme. COSC has applications in several real life situations including video sharing, participatory sensing and crowd sourcing. In future, we would like to investigate advanced methods for learning and updating mobility histories. How many failures (computed path set differing from reality) should be necessary to permanently change the mobility history? Does time also play a role?

References

[1] Wi-Fi Direct. http://developer.android.com/guide/topics/connectivity/wifip2p.html/, 2014. [Online; accessed 12-May-2014].

[2] C. Boldrini and A. Passarella. Hcmm: Modelling spatial and temporal properties of human mobility driven by users' social relationships. *Computer Communications*, 33(9):1056–1074, 2010.

[3] A. Casteigts, P. Flocchini, B. Mans, and N. Santoro. Deterministic computations in time-varying graphs: Broadcasting under unstructured mobility. In *Theoretical Computer Science*, pages 111–124. Springer, 2010.

[4] A. Casteigts, P. Flocchini, W. Quattrociocchi, and N. Santoro. Time-varying graphs and dynamic networks. In *Ad-hoc, Mobile, and Wireless Networks*, pages 346–359. Springer, 2011.

[5] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott. Impact of human mobility on opportunistic forwarding algorithms. *Mobile Computing, IEEE Transactions on*, 6(6):606–620, 2007.

[6] M. Conti and M. Kumar. Opportunities in opportunistic computing. *Computer*, 43(1):42–50, 2010.

[7] M. Conti, M. Mordacchini, A. Passarella, and L. Rozanova. A semantic-based algorithm for data dissemination in opportunistic networks. In *7th Intl. Workshop on Self-Organizing Systems (IWSOS)*, 2013.

[8] J. D. Gibson. *Mobile communications handbook*, volume 45. CRC press, 2012.

[9] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi. Understanding individual human mobility patterns. *Nature*, 453(7196):779–782, 2008.

[10] T. G. Griffin, F. B. Shepherd, and G. Wilfong. The stable paths problem and interdomain routing. *IEEE/ACM Transactions on Networking (ToN)*, 10(2):232–243, 2002.

[11] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot. Pocket switched networks and human mobility in conference environments. In *SIGCOMM 2005*, pages 244–251. ACM, 2005.

[12] L. Keller, A. Le, B. Cici, H. Seferoglu, C. Fragouli, and A. Markopoulou. Microcast: cooperative video streaming on smartphones. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*, pages 57–70. ACM, 2012.

[13] R. Kraut, C. Egido, and J. Galegher. Patterns of contact and communication in scientific research collaboration. In *Proceedings of the 1988 ACM conference on Computer-supported cooperative work*, pages 1–12. ACM, 1988.

[14] K. Lee, S. Hong, S. J. Kim, I. Rhee, and S. Chong. Slaw: A new mobility model for human walks. In *INFOCOM 2009, IEEE*, pages 855–863. IEEE, 2009.

[15] M. Musolesi and C. Mascolo. A community based mobility model for ad hoc network research. In *Multi-hop ad hoc networks: from theory to reality*, pages 31–38. ACM, 2006.

[16] K. Nahrstedt and L. Vu. CRAWDAD data set uiuc/uim (v. 2012-01-24). Downloaded from http://crawdad.org/uiuc/uim/, Jan. 2012.

[17] C. V. Networking. Global mobile data traffic forecast update, 2012'2017. *Cisco white paper*, 2013.

[18] A. Passarella and M. Conti. Characterising aggregate inter-contact times in heterogeneous opportunistic networks. In *NETWORKING 2011*, pages 301–313. Springer, 2011.

[19] L. Pelusi, A. Passarella, and M. Conti. Opportunistic networking: data forwarding in disconnected mobile ad hoc networks. *Communications Magazine, IEEE*, 44(11):134–141, 2006.

[20] A.-K. Pietilänen and C. Diot. Dissemination in opportunistic social networks: the role of temporal communities. In *13th Mobile Ad Hoc Networking and Computing*, pages 165–174. ACM, 2012.

[21] S. Qayyum, M. Shahriar, M. Kumar, and S. K. Das. Pcv: Predicting contact volume for reliable and efficient data transfers in opportunistic networks. In *LCN, 2013,* pages 801–809. IEEE, 2013.

[22] 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/, July 2009.

[23] U. Sadiq and M. Kumar. Proximol: Proximity and mobility estimation for efficient forwarding in opportunistic networks. In *(MASS), 2011 IEEE 8th Intl. Conference on*, pages 312–321. IEEE, 2011.

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

[25] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Single-copy routing in intermittently connected mobile networks. In *IEEE SECON 2004.*, pages 235–244. IEEE, 2004.

[26] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Efficient routing in intermittently connected mobile networks: The single-copy case. *IEEE/ACM Transactions on Networking (TON)*, 16(1):63–76, 2008.

[27] W. L. Tan, P. Hu, and M. Portmann. Experimental evaluation of measurement-based sinr interference models. In *(WoWMoM), 2012*, pages 1–9. IEEE, 2012.

[28] A. Vahdat, D. Becker, et al. Epidemic routing for partially connected ad hoc networks. Technical report, Duke University, CS-200006.

[29] B. B. Xuan, A. Ferreira, and A. Jarry. Computing shortest, fastest, and foremost journeys in dynamic networks. *Intl. Journal of Foundations of Computer Science*, 14(02):267–285, 2003.

[30] L. Yin and G. Cao. Supporting cooperative caching in ad hoc networks. *Mobile Computing, IEEE Transactions on*, 5(1):77–89, 2006.

[31] X. Zhang, G. Neglia, J. Kurose, and D. Towsley. Performance modeling of epidemic routing. *Computer Networks*, 51(10):2867–2891, 2007.