Raven: Energy Aware QoS Control for DRNs

Harsha Chenji†, Lidia Smith‡, Radu Stoleru†, Evdokia Nikolova†
† Department of Computer Science and Engineering, Texas A&M University
‡ Department of Mathematics and Engineering, Blinn College
{cjh, stoleru}@cse.tamu.edu, lidia.smith@blinn.edu, nikolova@tamu.edu

***Abstract*—Disaster Response Networks (DRNs) are disruption tolerant networks designed to deliver mission critical data during disaster recovery, while operating with limited energy resources. While Quality of Service is desired, it is difficult to offer guarantees because of the unpredictable nature of mobility in such DRNs. The variance of the packet delivery delay (PDV, more commonly called jitter), an important QoS metric which in DRNs is measured in tens of minutes instead of milliseconds, has not been sufficiently addressed in recent research. Smartphones used by first responders generate large data workloads, causing the PDV to further degrade. Reducing packet replication at these workloads will lower energy consumption, but reduces the packet delivery ratio (PDR). The complex interplay between these QoS metrics remains unclear, making their control difficult. We present Raven, a routing protocol for DRNs that offers control over QoS, especially the PDV. Stochastic graph theory which deals with probabilistic edge weights having a mean and variance is used to model mobility in the disaster area. A stochastic version of the K-Shortest Paths algorithm routes data over multiple paths simultaneously. Raven has been thoroughly evaluated in simulation using realistic settings. The dynamics between performance and energy consumption is analyzed mathematically, and its control is demonstrated.**

***Keywords*—Delay tolerant networking; Quality of service**

I. Introduction

Natural disasters cause loss of power and communication infrastructure in the affected area, and make the recovery process challenging. The Federal Emergency Management Agency in the U.S.A describes Urban Search and Rescue (USAR) as a support function that deals with collapsed structures in urban areas. Since USAR first responders have always communicated using traditional means such as paper and paint on walls, recent research [1] has looked at providing them with new sensing modalities like low power wireless sensors and smart phones. DRNs use concepts from delay tolerant networking (DTN) research to build a networking infrastructure that integrates these modalities and allows responders to share data. These DTNs are expected to handle heterogeneous data with different QoS requirements: sensor data can be measured in kilobytes, while smartphones generate multi-gigabyte videos.

It is difficult for DTNs to provide hard QoS guarantees primarily because mobility in the area is inherently unpredictable and random. This unpredictability makes it very difficult to accurately estimate the node inter-contact time, which is the primary component of the end-to-end packet delivery delay [2], and remains an open problem. Limited buffers and bandwidth contribute to the complexity of estimating the PDR and energy consumption for a given workload. Recent DTN research has looked at algorithms which improve, but not guarantee, traditional QoS metrics like the average packet delivery delay (PDD) and PDR - but largely ignores the PDV. In traditional networks, the PDD/PDV is typically measured in milliseconds - but in DTNs it can range from tens of minutes (trace based experiments in [3]) to even hours. This means that some packets may have a delivery delay much higher than the average delay. Since sensed data from the field is used to make decisions at the Emergency Operations Center (EOC), a large PDV becomes problematic since some critical data may arrive very late.

Designing a DTN system for disaster recovery now poses several challenges: *Is it possible to control the PDV in a DTN? If so, what are the implications on the PDD, PDR & energy consumption?* Recent research [4][5][6] has shown that packet replication, used to combat uncertain mobility through redundancy, reduces PDD but results in higher PDR & energy consumption. Forwarding based techniques have lesser overhead but need more a priori information (which is difficult to obtain in an opportunistic DTN) to outperform replication based protocols. Analysis of the PDD/PDR/energy in DTNs have not included the PDV to the best of our knowledge. In research related to finance and traffic engineering, decision making in the presence of uncertainty is called risk-aversion. A risk-averse user will prefer a strategy whose reward has lower variance (i.e., more predictable) but a higher mean (i.e., lower reward). It is this concept of risk-aversion that we wish to make available to a first responder who is using an opportunistic DTN during disaster recovery (reward here is the PDD and the PDV represents uncertainty). Unfortunately, there is neither a DTN framework that jointly analyzes PDD, PDV, PDR & energy, nor an algorithm that is able to control the quantities simultaneously.

In this paper we present a DTN routing framework that provides its users with the ability to control QoS: PDD & PDV via risk-aversion, PDR & energy consumption via replication. Raven (Risk AVersE routing in dtNs) models mobility in the disaster area using the Post Disaster Mobility (PDM) model [7]. A "stochastic multigraph", where multiple edges with probabilistic weights is possible between vertices, represents a mathematical abstraction of the PDM scenario. Important geographical locations in a disaster called Centers are mapped to vertices, and DTN data mules (called Mobile Agents), to edges. The risk associated with each path between source and destination is calculated using a mean-risk model. Source routing is performed by selecting the least risk paths between the two vertices, using a K-Safest Paths algorithm. In order to route data between USAR responders around a Center, a forwarding decision is made wherein a packet is passed to the responder who represents the least risk in reaching the destination. Our contributions are as follows: 1) the first paper to apply the concepts of risk-aversion and QoS to DTNs, 2) a distributed protocol which controls risk-aversion as well as replication simultaneously, and 3) analysis on the coexistence of risk-aversion & replication.

978-1-4799-0428-0/13/$31.00

II. Motivation and Related Work

First, we define our mobility model which is based on the Post Disaster Mobility (PDM) model [7]. A simple example scenario is constructed, and the corresponding stochastic multigraph is calculated. The concept of least risk path is introduced as a generalization of the shortest path when both the mean and variance of the delay are considered. Finally, related state of art research is discussed.

A. Mobility Model

The PDM model uses two main components to functionally model the interaction between survivors and rescue workers: "mobile agents" (MAs) and "Centers" which are fixed areas in the city and represent important areas such as the EOC. Each MA moves from one point to another on a predefined set of roads. It is assumed that a networking device is present at each Center as well as on each MA. There are multiple Centers C, of which two are special and compulsory: the EOC and the Triage. The categories of MAs are: USARs, Volunteers, Supply Vehicles, Ambulances and Patrol Cars. Each category has its own min and max speeds, and an agent belonging to that category chooses a speed uniformly between min and max at random, for each leg of travel.

In the Volunteer Movement Model, each volunteer is placed at a randomly assigned home center $c_H \in C$ initially. Next, every volunteer individually chooses a random point within the entire map with 90% probability or chooses c_H with 10% probability, and travels to it along the shortest path. The process is repeated upon reaching the point. In the Supply Vehicle Movement Model, each SV is placed at a randomly assigned home center $c_H \in C$ initially. Then each SV individually chooses a center $c_d \in C$ at random and travels to it along the shortest path. The process is repeated upon reaching c_d. In the Patrol Car Movement Model, each car has a predefined list of centers $\{c_1, c_2, \ldots, c_n\} \in C \times C \times \cdots \times C$, and is placed at c_1 initially. Next, it travels to c_2 along the shortest path, and the process is repeated by choosing the next center in the list. In the Ambulance Movement Model, each ambulance is always assigned to the Triage initially. Next, each ambulance chooses a Center (including the Triage) at random to travel to, following which the ambulance always returns to the Triage. The process is repeated, resulting in a series of alternating Centers and Triages.

We add the "Urban Search and Rescue Worker (USAR)" category to the above list. USARs operate in a area of fixed radius around a Center. In the USAR Movement Model, each USAR member is placed at a predefined home center $c_H \in C$ initially. Then, every USAR agent individually randomly chooses a point uniformly within radius r of its c_H, and travels to it along the shortest path possible in the map. After reaching the point, the process is repeated. USARs need not visit c_H compulsorily. An example PDM scenario is depicted in Figure 1a. Following a disaster, the EOC has been setup, a collapsed building (RUBBLE) has been identified for search and rescue operations and a medical TRIAGE area has been setup resulting in three Centers. For simplicity only three categories of MAs are shown: ambulances, supply vehicles and USARs. USARs move around the Triage and Rubble in an area of fixed radius shown by the dotted line.

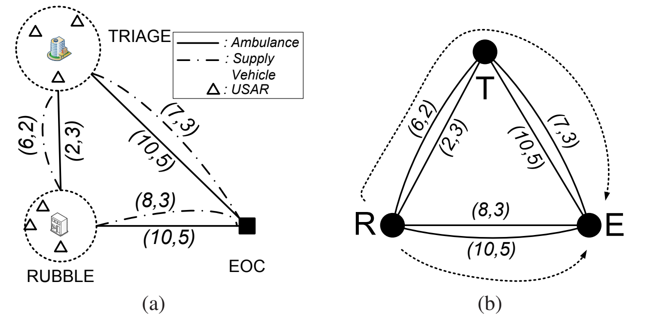


Fig. 1: (a) A simple scenario with 3 Centers and 3 Mobile Agents: ambulances, supply vehicles and USARs. Numbers next to a path indicate the (mean,variance) of the travel delay in minutes along that path, for the category of Mobile Agent represented by the line type (solid vs dashed). (b) Stochastic multigraph for this scenario. Vertices R, T and E correspond to the Rubble, Triage and EOC centers respectively. The edges represent Mobile Agents and have the same weights.

	Path	Leg1	Leg2	PDD	PDV	P(X<6)%	P(X<8)%
1	R→E	SV	-	8	3	25.2493	50 ⋆
2	R→T→E	Amb	SV	9	6	30.8538 ⋆	43.3816
3	R→E	Amb	-	10	5	21.1855	34.4578
4	R→T→E	Amb	Amb	12	6	15.8655	25.2493
5	R→T→E	SV	SV	13	5	8.0757	15.8655
6	R→T→E	SV	Amb	16	7	7.6564	12.6549

TABLE I: Enumerating the mean and variance of the delivery delay along multiple paths for routing data from R to E. All values are in minutes. SV stands for supply vehicle, Amb stands for ambulance. Path 2 is optimal for a deadline of 6 minutes, while path 1 is optimal for a deadline of 8 minutes. What is optimal for a deadline of 6 minutes is not necessarily optimal for a deadline of 8 minutes.

B. Risk as an Alternate Path Optimality Metric

Using the above example scenario, we now illustrate how the shortest path according to the PDD metric changes when its variance (PDV) is taken into account. This example motivates the need for an alternate path optimality metric that incorporates the second moment, the PDV metric in this case, in addition to the first moment or the PDD. Suppose that data is routed from one Center to another, following a "route by area" paradigm. The travel delay is a major contributor to the delivery delay. Minor contributors like queuing delay are neglected. The travel time between Centers is different for each type of Mobile Agent (MA) - and has a mean and a variance because the speed has a min and max, and this distribution is unique because of the different movement models. The scenario in Figure 1a is represented using a stochastic multigraph in Figure 1b. A stochastic multigraph in this context is a graph whose vertices represent Centers and an edge represents a MA category that visits the two incident Centers. Assuming that the travel time distributions are Gaussian, the delay distribution of a path is the sum of the distributions of its constituent edges. The PDD and PDV for multiple paths from Rubble to EOC in the above scenario (Figure 1b) is shown in Table I. When the data has a deadline of 8 minutes, path 1 is optimal since it has the highest probability that the delay will be less than 8 minutes. However, when the deadline changes to 6 minutes, path 2 is optimal. Determining the best (w.r.t delay) path in such a scenario becomes problematic. A combination of the mean and variance, called risk (Section IIIC), can be used as a path optimality metric. The "least risk path" will then be optimal.

C. Related Work

In this section prior research conducted in the fields of delay tolerant networking and stochastic optimization is discussed. The algorithms developed and used in Raven are placed in this context, while the similarities and differences between Raven and other routing frameworks are presented.

Delay Tolerant Networking: A DTN routing protocol can be classified based on its approach towards five major criteria: 1) Redundancy: forwarding/limited replication/unlimited replication, 2) Contact Bandwidths: limited/unlimited, 3) Storage Resources: limited/unlimited, 4) Knowledge of Mobility: none/partial/learned/complete/oracle and 5) Target Metric: mean delivery delay/ energy efficiency/other. Raven can either forward or replicate messages, assumes limited contact bandwidths, assumes unlimited storage, possesses partial knowledge of mobility and targets the multiple metrics simultaneously. Based on the taxonomy in [3], Raven can be placed in a new "P6" category for protocols which assume unlimited buffers and limited contact bandwidth. The primary motivation for assuming unlimited storage capacity is the falling price of storage cost (0.07 USD/GB in 2009; 0.03 USD/GB today), as well as the relatively low contact bandwidth (tens of MBs) compared to disk space (a few TBs). Recent COTS WiFi routers like the Netgear WNDR3700v2 have a USB port, which means that several terabytes of storage can be provisioned by connecting a USB drive. This is practically unlimited in comparison to the average compressed HD video or photograph taken with a smartphone camera which is about 1GB and 4MB respectively. The assumption of limited contact bandwidth is justified by the experiments in [8]. A short overview of recent DTN routing protocols in the context of the five criteria follows.

Epidemic [6] routing performs unlimited replication and relays a copy of the message to every node possible, assumes unlimited resources and does not attempt to leverage mobility. The authors of [4] have analyzed the trade off between replication and energy efficiency in an epidemic-like routing protocol. Spray and Wait [5] performs limited replication of a message according to a user specified parameter, while retaining other assumptions. A mathematical analysis of this protocol is available in [9]. Prophet [10] and MaxProp [11] leverage mobility in a DTN by selecting relays based on historical encounters such that the likelihood of delivery is maximized. RAPID [3] allows the user to target any metric, including the delivery delay, and works by estimating the marginal utility gained in replicating a packet to a host. Such protocols are often called utility based protocols since a relaying decision is made by comparing a node's utility to its contact's. [12] uses MCMC methods to perform utility based relay selection in a DTN. Scoop [13] is a DTN multicast routing protocol which leverages locally observed information about mobility to minimize the delivery delay. The R3 [14] protocol unifies mesh/MANET/DTN routing paradigms by using learned information about the distribution of link delays to perform replication based routing. DTN routing is most optimal when all future inter-node contacts are well known. In [15] a comprehensive linear programming formulation of DTN routing with limited contact bandwidths is presented, with the assumption that interrupted transfers can be resumed. However, it cannot be applied to opportunistic DTNs knowledge of all future contact durations is needed. In [16], the authors mention that jitter is an applicable QoS metric but propose two additional metrics.

The above protocols do not allow control of the PDV, with the possible exception of utility based routing protocols where a target metric can be specified. These utility based protocols need *the user* to provide the protocol with a formula which can estimate the target metric, in this case the PDV. As the complexity of the mobility model increases, estimating this metric becomes less trivial and more challenging mathematically. Raven on the other hand provides the user with a single paramter ρ which when increased, automatically lowers the PDV. Thus, no complex mathematical modeling on the user's part is needed. Additionally, the above protocols do not allow for *simultaneous* and *intentional* control of multiple target metrics; the effect these protocols have on the QoS metrics may be incidental. Very few protocols (like SprayWait) are able to choose between forwarding ($L = 1$) and controlled replication ($L > 1$) when required, which is important at high data loads since uncontrolled replication can be harmful. Raven controls replication using the K system parameter. It should be noted that Raven is designed keeping the PDM mobility model in mind and could theoretically work with other mobility models; but this is outside the scope of this paper and is future work. Prior art which is most similar to Raven is [17] in the sense that it is a DTN routing protocol for disaster response networks and uses the PDM mobility model. However, [17] focuses on reducing the energy spent on communication whereas Raven is optimized for controlling multiple metrics (most notably the PDV) *including* the energy consumption.

Stochastic Graph Theory: Problems involving graphs with either multidimensional or probabilistic edge weights have been investigated. An overview of the shortest path problem with probabilistic edge weights is found in [18]. In [19] a stochastic shortest path algorithm is used to solve a route planning problem in the presence of uncertain traffic delays. [20] uses expectations of link delays to solve a network routing problem. To the best of our knowledge, the K-Shortest Paths problem has not been extended to graphs with stochastic weights, and such graphs haven't been used to model DTNs. Note that some works use the word "stochastic" to refer to the problem of delivering broadcast or multicast traffic to nodes with a probabilistic guarantee, but the similarity ends there.

III. The Raven Routing Framework

In this section we first present the problem formulation followed by a detailed explanation of how Raven works, culminating in a distributed protocol. This paper poses two research questions: 1) Is it possible to incorporate risk-aversion in a DTN by controlling the PDV? If so, how can it be done using a routing protocol? Is replication necessary? and 2) What are the effects of enabling risk-aversion and replication on quantities like PDR and energy consumption? This section addresses question 1, whereas the second question is answered in the following Section. In the context of the PDM mobility model, the routing problem can be seen as the union of two subproblems: routing between Centers and routing within Centers. Following a description of these subproblems, the process of building the stochastic multigraph by estimating the travel time distributions is discussed. A formal definition of risk is then derived, followed by the K Safest Paths algorithm.

Preliminaries: A set of centers C and their locations L_c is known. The set of Mobile Agent categories is M, and each category m has n_m agents. For each category m, the min and max speeds of the agent are known. The radius R around each Center in which USAR agents operate is specified. Collectively, these quantities are called the PDM scenario $\mathcal{P}$. The risk-aversion factor ρ and the number of paths K is provided by the user. The scenario is not completely known because of the randomized movement - thus, replication is necessary.

A. Problem Formulation

The problem formulation involves the PDM model, stochastic multigraphs resulting from the PDM model, and risk-aversion. In a scenario represented by the PDM model, there are two types of data flows possible: between Centers and within Centers. Statically deployed chemical sensors deployed around a Center, for example, need to report data to the EOC periodically. Such a data flow is an example of the Center to Center model. Data from sensors is first collected on the static node at the Center and is then sent to the static node at the EOC using Mobile Agents other than USARs. The EOC on the other hand, may choose to push information to the USAR agents working at a Center. This is an example of a hybrid flow, because data from the EOC is first sent to the static node at the Center, where it will be disseminated among USAR agents using the "within Centers" flow.

Problem Formulation: Given a DTN scenario represented by the PDM mobility model with Centers C and Mobile Agent categories M, calculate the $K \geq 1$ least risk paths between source $S \in C$ and destination $D \in C$. The risk associated with a path is calculated based on a user defined risk-aversion coefficient $\rho \geq 0$, where $\rho = 0$ means the PDD should be minimized over the PDV and $\rho \to \infty$ means the PDV should be minimized over the PDD. Mathematically, for a source $c_s \in C$ and destination $c_d \in C$, calculate K least risk paths where each path P is a set of alternating Centers and Mobile Agents.

This formulation optimizes the risk but not the PDR & energy consumption because varying K (replication) only has a coarse grained effect on PDR/energy, whereas varying ρ (risk-aversion) has fine grained effects on PDD & PDV. Modeling and solving the Routing Within Centers problem within the Raven framework is left as future work. Currently, direct delivery is used to route data to a Center from a USAR agent and vice versa.

B. The Stochastic Multigraph and its Construction

A stochastic multigraph $\mathcal{S}$ maps the Centers C of the PDM model to vertices and Mobile Agent categories M to edges, while the edge weights are not scalars but random variables. In a stochastic multigraph $\mathcal{S} = G(V, E)$ each edge $e \in E$ has an associated mean μ_e and variance σ_e^2 corresponding to the travel time distribution. The weight of an edge W_e is defined as a pair of scalars $W_e = (\mu_e, \sigma_e^2)$. We only consider stochastic time-independent graphs where W_e do not change over time. A *stochastic path* p in $\mathcal{S}$ is a graph path whose edges and vertices are present in $\mathcal{S}$. The weight of this path W_p is the sum of the distributions of the edges in the path. If the distributions are Gaussian:

$$W_p = \sum_{e \in p} W_e = (\mu_p, \sigma_p^2) = (\sum_{e \in P} \mu_e, \sum_{e \in p} \sigma_e^2) \qquad (1)$$

Prior to constructing $\mathcal{S}$, the travel time distributions for each Mobile Agent category between each pair of Centers need to be calculated. In recent mobility model research, the time taken for two nodes to meet each other when starting from different positions is called the meeting time (if both nodes are mobile) or the hitting time (if one node is mobile) [9]. In the PDM mobility model, we define the meeting time between any two Mobile Agents u_1 and u_2 as the time taken for them to come in contact with each other when starting from different positions $L_{u_1}(0)$ and $L_{u_2}(0)$ and moving according to their respective movement models. If their radio range is R,

$$MT(u_1, u_2) = \min_t \{t : ||L_{u_1}(t) - L_{u_2}(t)|| < R\} \qquad (2)$$

The hitting time (HT) as defined in [9] is a special case of the meeting time, defined when one of the nodes is static. In the context of PDM, HT is defined between a Mobile Agent and a Center:

$$HT(ma_1, c) = \min_t \{t : ||L_c - L_{ma_1}(t)|| < R\} \qquad (3)$$

Routing Between Centers: In the stochastic multigraph as shown in Figure 1b, each edge represents the travel time between the two Centers, through a particular category of Mobile Agent. That is to say, the mean and variance of each edge is the time taken for an Ambulance (for example) to travel from the Rubble to the Triage. This is nothing but the hitting time (Equation 3), where ma_1 is an ambulance and c is the Triage. Since the ambulance starts at the Rubble, $L_{ma_1}(0) = L_{rubble}$ and $L_c = L_{triage}$. The PDM model is map based and hence closed form solutions for HT are highly dependent on the underlying map as well as the locations of the Centers. Therefore, values for HT, and hence the edge weights, need to be derived from simulation. The TheONE DTN simulator, when given a map and the list of Centers, can place Mobile Agents at any intersection and calculate the travel time to another Center. The Dijkstra algorithm is used to choose the shortest path between two points in the map. This way, the HT values can be stored in a lookup table and accessed later.

Construction: The algorithm for constructing $\mathcal{S}$ is shown in Algorithm 1. First, a vertex is created for each PDM Center (Step 1). Since each edge represents a Mobile Agent category, $|M|$ edges are drawn between each pair of vertices (Step 2). However, some edges (Step 3) are redundant - for example, a Patrol Car may not visit all Centers. Therefore it is necessary to cull some edges based on whether the movement model allows for agent to visit those two Centers (Steps 4-6). If the edge is allowed, the mean and variance to be assigned is first calculated (Step 7). This is nothing but the hitting time HT for an agent m_1 of category m, when it starts at Center a and hits Center b (Equation 3). The mean and variance of the HT is assigned to that edge (Step 8).

Algorithm 1 Construction of $\mathcal{S}$

Input: PDM scenario $\mathcal{P}$, ρ, K
Output: Stochastic graph $\mathcal{S}$
1: Create $|C|$ vertices in $\mathcal{S}$
2: Draw $|M|$ edges between each pair of vertices in $\mathcal{S}$
3: **for** edge $e(a, b, m)$ where $a, b \in C$ and $m \in M$ **do**
4:     **if** Movement model of m does not involve a or b **then**
5:         Delete e and **continue**
6:     **end if**
7:     $\mu(e), \sigma^2(e) \leftarrow$ HT(m, L_b) s.t $L_m(0) = L_a$ ▷ By simulation
8:     $e \leftarrow \mu(e), \sigma^2(e)$
9: **end for**
10: **return** $\mathcal{S}$

Algorithm 2 K-Safest Paths Algorithm (KSfP)

Input: Stochastic multigraph $\mathcal{S}$, K, source s, dest. d
Output: $\mathbb{K}$, a set of K paths in $\mathcal{S}$ between s and d
1: **for** edge e in $edges(\mathcal{S})$ **do**
2:     $weight(e) \leftarrow \mu_e + \rho * \sigma_e^2$ ▷ Modified Risk Formula
3: **end for**
4: $\mathcal{S}' \leftarrow \mathcal{S}$ with edge weights as above
5: $\mathbb{K} \longleftarrow$ Apply K-Shortest paths on $\mathcal{S}'$ with src/dest s/d
6: **return** $\mathbb{K}$

Algorithm 3 The Raven Routing Protocol

Input: PDM scenario $\mathcal{P}$, ρ, K, source s, destination d
1: Build $\mathcal{S}$ using Alg 1 input $\mathcal{P}, \rho, K$
2: **if** s and d are Centers **then** ▷ Routing Between Centers
3:     $\mathbb{K} \leftarrow$ Use Alg 2 input $\mathcal{S}, K, s, d$
4:     Source route along the paths in $\mathbb{K}$
5: **else if** s is a USAR, d is a Center **then**
6:     $c_s \leftarrow$ the Center around which s works
7:     Direct delivery to c_s
8:     Apply Raven at c_s with input $\mathcal{P}, \rho, K, c_s, d$
9: **else if** s is a Center, d is a USAR **then**
10:     $c_d \leftarrow$ the Center around which d works
11:     $\mathbb{K} \leftarrow$ Use Alg 2 with input $\mathcal{S}, K, s, c_d$
12:     Source route along $\mathbb{K}$ still packet reaches c_d
13:     Direct delivery from c_d to d
14: **else if** s is a USAR, d is a USAR **then**
15:     $c_s, c_d \leftarrow$ the Centers around which s, d work
16:     **if** $c_s = c_d$ **then**
17:         Deliver packet to c_d directly
18:     **else if then**
19:         Direct delivery to c_s
20:         Apply this algorithm at c_s with input $\mathcal{P}, \rho, K, c_s, d$
21:     **end if**
22: **end if**

C. Quantifying Risk

A stochastic path is considered "risky" if there is a high probability that a sample from $\mathcal{N}(\mu_p, \sigma_p^2))$ will deviate far from the expected value (μ_p) [21]. In order to quantify this "risk", we adopt the mean-risk probability model [22]. The risk R_e of an e in $\mathcal{S}$ is defined as $R_e = \mu_e + \rho * \sigma_e$ where the *risk-aversion coefficient* $\rho \geq 0$ is a user specified quantity. It represents how important the variance of the path weight is to the user. A ρ of zero in stochastic routing chooses the path with the least mean. Similarly, for a path p its risk R_p is defined as $R_p = \mu_p + \rho * \sigma_p$. However, it is not equal to the sum of the risks of its edges:

$$R_p = \sum_{e \in p} \mu_e + \rho * \sqrt{\sum_{e \in p} \sigma_e^2} \quad \Rightarrow \quad R_p \neq \sum_{e \in p} R_e$$

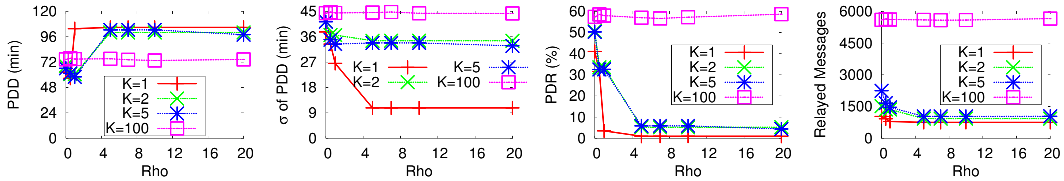
K-Safest Paths Problem (KSfP): In order to choose K least risk paths, we adopt the K-Shortest Paths (KShP) problem in deterministic graphs to stochastic graphs. A "safe" path of two paths p_1 and p_2 is the one with the lower risk $min(R_{p_1}, R_{p_2})$. The objective of KSfP is to choose the K safest paths of a stochastic graph $\mathcal{S}$, given a source node and a destination node. K is a natural number. Existing algorithms for KShP (classical version) include a modified Bellman-Ford algorithm that stores the top K shortest paths at each pass instead of storing only the shortest (JGraphT library), and Yen's algorithm [23]. The stochastic shortest path problem (KSfP with $K = 1$) is a non-convex combinatorial problem [19]. A dynamic programming approach is incorrect since sub-paths of optimal paths are not optimal. The risk of a path is not a linear combination of the risks of the edges, but is in fact non-linear as seen above ($R_p \neq \Sigma R_e$). We therefore propose the use of variance instead of the standard deviation for simplicity. While dimensional homogeneity is not present due to the use of variance which is the square of the standard deviation, the implementation of KSfP becomes straightforward and simple. The algorithm is shown in Algorithm 2. The stochastic graph is first converted into a deterministic graph (Steps 1-3). The edge weight is computed using the modified risk formula $R_p = \mu_p + \rho * \sigma_p^2$. Each edge e in the stochastic graph $\mathcal{S}$ is assigned a deterministic edge weight (Step 2). The modified graph $\mathcal{S}'$ is now completely deterministic (Step 4). Any KShP algorithm can now be applied (Step 5). The result is a set of paths $\mathbb{K}$ that have the least risk.

D. The Raven Routing Protocol

The main algorithm is shown in Algorithm 3. It is performed when a packet is generated at a source node s which may be a USAR Mobile Agent or a Center. The first step constructs the stochastic multigraph $\mathcal{S}$ (Step 1) using Algorithm 1 (Section IIIB). The next steps implement the hybrid data flow model as explained in Section IIIA. The first possibility is that both source and destination are centers (Step 2). The K Safest Paths algorithm is used (Step 3) to find the set of K paths (Section IIIC), and source routed along these paths. If the source is a USAR agent (Step 5), the packet is first delivered directly to the agent's Center (Steps 6-7), and then the Raven algorithm is used as if the packet was created at the agent's Center. The workflow is similar if the destination is a USAR agent (Steps 9-13). If both source and destination are USAR agents working at the same Center (Steps 14-16), then the source waits till it meets the destination (Step 17). If not, then a combination of the previous strategies is applied (Steps 18-20).

IV. Analysis and Performance Evaluation

In this section mathematical analysis on the coexistence of risk-aversion and replication as well as their effects on the QoS metrics is presented and the performance evaluation is discussed. The interdependence of ρ and K is shown in Table II. Borrowing terminology from [3], an intentional effect changes a metric *by design*, whereas an incidental effect does so *indirectly*. Some results are presented following a mathematical problem formulation.



(a) Effect of ρ upon the PDD (b) Effect of ρ upon the PDS (c) Effect of ρ upon the PDR (d) Effect of ρ upon energy

Fig. 2: Behavior of the QoS metrics as ρ and K change at a workload of 1GB per flow, 4MBps bitrate and 1x speed.

	PDD	PDV	PDR	Energy
ρ	Intentional	Intentional	Incidental	Incidental
K	Incidental	Incidental	Intentional	Intentional

TABLE II: The type of effect each Raven parameter (in rows) has on the QoS metrics of interest (in columns).

A. *General Problem Formulation*

Suppose that we have n normal distributions in the set $\{P\}$ (representing the paths from a given source to a given destination in the stochastic multigraph). Each of these distributions P_i has an associated mean and variance $(E[P_i], V[P_i]) = (\mu_i, \sigma_i^2)$. A risk-aversion coefficient $\rho \geq 0$ assigns a scalar quantity called risk $(= \mu_i + \rho * \sigma_i)$ to each P_i according to the mean-risk probability model. When one wants to be risk-averse, this set of n paths is ordered according to the risk of each path, resulting in a set $\{Q\}$ such that:

$$\{Q\} = Q_1, Q_2, Q_3, \ldots, Q_n \text{ where } Q_i \in P$$
$$i < j \Rightarrow (E[Q_i] + \rho * \sqrt{V[Q_i]}) < (E[Q_j] + \rho * \sqrt{V[Q_j]})$$

Without replication, data will be sent only on the path Q_1. However, with replication, the first $K \geq 1$ elements of $\{Q\}$ which are $\{Q_K\} = \{Q_1, Q_2, \ldots, Q_K\}$ are chosen. The delivery delay will now have a mean of $mean(\rho, K) = PDD = E[W]$ and a variance of $var(\rho, K) = PDV = V[W]$ where $W = min\{Q_1, Q_2, \ldots, Q_K\}$. These paths can be assumed to be i.i.d because of the assumptions of 1) infnite buffer and 2) the physical travel is the major contributor to the packet delivery delay. The c.d.f of the minimum of a set of independent distributions is defined as follows:

$$P(W \leq x) = 1 - P(Q_1 > x)P(Q_2 > x) \ldots P(Q_K > x)$$
$$= 1 - \prod_{i=1}^{K} P(Q_i > x) = 1 - \prod_{i=1}^{K}(1 - P(Q_i \leq x))$$
$$= 1 - \prod_{i=1}^{K}(1 - \Phi\left(\frac{x - E[Q_i]}{\sqrt{V[Q_i]}}\right)) = 1 - \prod_{i=1}^{K}(1 - \Phi_i) \quad (4)$$

where $\Phi(x)$ is the c.d.f of the standard normal distribution and Φ_i is, with abuse of notation, defined as scaling $\Phi(x)$ to a distribution with non-standard mean $E[Q_i]$ and variance $V[Q_i]$. Once we have the c.d.f, the mean and variance are:

$$P(W \leq x) = F_W(x) \text{ and } f_W(x) = F'_W(x)$$
$$mean(\rho, K) = E[W] = \int_{-\infty}^{\infty} x f_W(x)\, dx \quad (5)$$
$$var(\rho, K) = V[W] = \int_{-\infty}^{\infty} (x - E[W])^2 f_W(x)\, dx \quad (6)$$

Result R1: ρ has an incidental effect on PDR & energy while K has an intentional effect. For a given K, ρ and a packet/source/destination, a set of paths $\{Q_K\}$ is constructed. Let J be the union of all the Centers present on these K paths. The number of relayed messages, and hence the energy, is proportional to the cardinality $|J|$. This is because if a Center is present on multiple paths, the packet will be relayed to it only once since infinite buffers are assumed. Similarly the PDR is proportional to the number of paths K. For two different ρ, there is no guarantee that $|J|$ or K will change since ρ only changes the order of paths and not the number of paths. Thus, ρ only has an incidental effect on the PDR & energy. K has an intentional effect on the energy/PDR since the number of paths as well as $|J|$ (converges to the total number of Centers) are guaranteed to increase.

R2: $mean(\rho, K) \to -\infty$ as $K \to \infty$. This is because as K increases, the minimum of K normally distributed random variables will decrease. While the result is intuitive, a proof of this statement for the general case is difficult owing to the complexity of solving Equation 5 for a non-standard normal distribution. However, this result has been proved for K standard normal distributions [24] and is known as the extreme first order statistic. Suprisingly, such a result for variance does not seem to hold even for two random variables [25]. This result has been confirmed by the authors using Monte Carlo simulations on stochastic multigraphs extracted during simulation. As a corollary, $mean(\rho, K_1) < mean(\rho, K_2)$ if $K_1 > K_2$.

R3: As $K \to \infty$, the effect of ρ will be less and less pronounced. In other words, it is difficult to be risk-averse at high K. This is because 1) the order of Φ_i in Equation 4 does not matter since it is a product, and 2) ρ only changes the order but not the number of selected random variables.

R4: $m_1 \leq mean(\rho, K = 1) \leq m_2$ as $\rho \to \infty$, where m_1 is the mean of the P_i with the smallest mean and m_2 is the mean of the P_i with the smallest variance. The proof is trivial since an increasing ρ chooses a lower variance by definition.

R5: If $\rho_1 > \rho_2$, $mean(\rho_1, K = 1) > mean(\rho_2, K = 1)$. In order to understand this slightly counter-intuitive result, imagine the $\mu - \sigma$ Pareto frontier of a graph where each distribution P_i corresponds to a point $(x, y) = (\mu_i, \sigma_i)$. As ρ goes from 0 to ∞, the distribution minimizing $\mu + \rho * \sigma$ will be the one with smallest mean, then the next one on the $\mu - \sigma$ Pareto frontier, and so on until the bottom-most distribution is selected. Using this graph, it is easy to see that $mean(\rho_1, K = 1) > mean(\rho_2, K = 1)$.

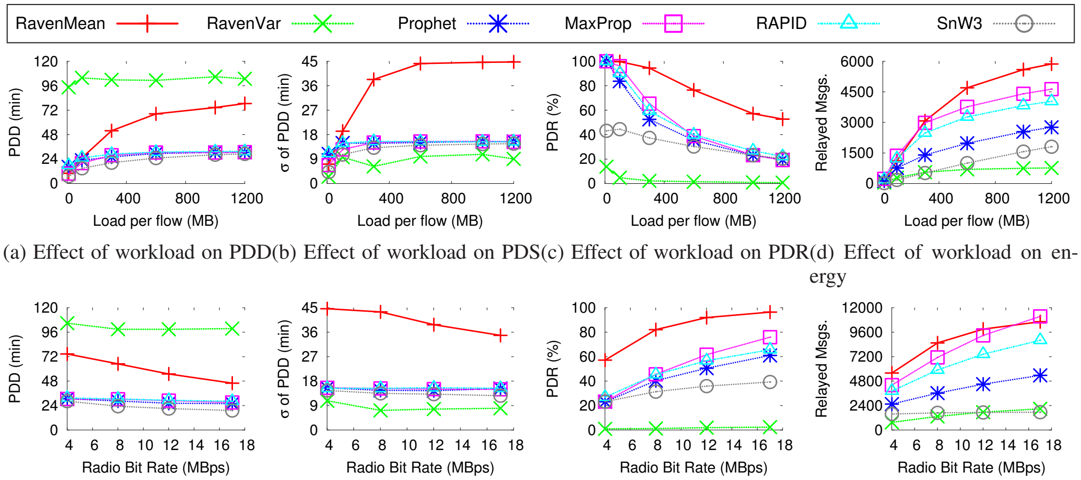
B. Performance Evaluation

Evaluations were performed in simulation using TheONE simulator with the PDM mobility model, using the Helsinki street map. Trace based evaluation is difficult since movement traces of first responders are not readily available. An EOC and TRIAGE are setup in the city, with 7 collapsed buildings where urban search and rescue is to be performed, resulting in 9 total Centers. 3 ambulances and 3 supply vehicles move in the city using their respective mobility models. Both categories have a speed of (0,40) m/s. 10 volunteers move using the Volunteer Mobility Model with their home centers as the EOC and a speed of (0,4) m/s. A patrol car moves using its mobility model among collapsed buildings 1, 2 and 4 with a speed of (0,40) m/s. A speed multiplier simulation parameter changes the min and max speeds of each Mobile Agent proportionately. Data is sent on three flows simultaneously and all packets are created at $t = 0$. Unless specified, the default load is $1GB$ per flow and the bandwidth of each node's radio is $4MBps$. The simulated time is 10000 seconds and each data point is averaged over 25 random runs. It took about 75 processor-hours of wall clock time to gather data for this section. The four metrics of interest are the three QoS metrics (PDD, PDV, PDR) and the energy consumed by the system. The number of relayed messages is an indication of the latter since the on board radio draws a large current (as compared to the storage device, for example). For reasons of dimensional homogeneity, we compare the standard deviation of PDD (called PDS for brevity) instead of the PDV. First, the effect of varying K and ρ upon the PDD and PDV is measured. Based on the results, we tune Raven by choosing a particular K and ρ for comparison with other state of the art protocols. The following sections demonstrate the effect of increasing load, increasing node speed and increasing radio bitrate upon the four metrics of interest. The protocols chosen for comparison are RAPID [3], Prophet [10], MaxProp [11] and SprayWait [9] (with $L = 3$). We chose $L = 3$ to demonstrate that simply fixing replication at a low number will not improve performance at high loads. RAPID is a utility based routing protocol tuned to minimize the PDD based on the marginal utility of replicating a packet, whereas Prophet and MaxProp attempt to characterize the mobility and replicate packets only to better hosts, i.e., those with a higher probability of meeting the destination.

Tuning Raven: For a fair evaluation, two variants of Raven are constructed: RavenMean and RavenVar, which are designed to minimize the PDD and PDV respectively. In order to configure the two variants with appropriate K and ρ, experiments were conducted. The performance for varying K and ρ is shown in Figure 2. One immediate observation is that as ρ increases, the PDV decreases (Figure 2b) but at the cost of increased PDD (Figure 2a). This is an expected result since a higher ρ places more emphasis on the variance of the paths and causes the algorithm to choose paths with lesser risk. Increasing ρ causes marked improvement at lower values while only incremental improvement is observed at higher ρ. The value at which ρ saturates depends upon the topology, which decides the Pareto frontier for the paths. Thus, choosing a ρ is highly dependent on the topology: the number of Centers as well as the speed of Mobile Agents. The PDR is very low (Figure 2c) since the simulation time was 10000 seconds (166 mins) and the increasing PDD causes packets to be delivered outside the simulation window. The PDD in (Figure 2a) is calculated *based on the delivered packets only*; since the PDR is high for high K, so is the PDD for high K. Result R3 is confirmed in the sense that at K, the effect of ρ decreases causing the metrics to stay constant. R1 is demonstrated since ρ has only an incidental and minor effect on the energy consumption (Figure 2d), causing the metric to stay fairly constant compared to the PDR. With an increase in K the amount of replication in the network increases. As expected, since a single packet travels on more and more paths simultaneously, the PDD decreases with increasing K (Figure 2a) and the PDR increases as well (Figure 2c). But this comes at a cost - both the PDV (Figure 2b) and the energy consumption increase with K (Figure 2d) for a given ρ. An infinite K is equivalent to Epidemic - a packet travels on all possible paths through the network. When a packet is flooded, the limited contact bandwidths are used inefficiently through redundant data transfers, causing some packets to be delivered quickly while other packets stay in the queue. As before, K follows the law of diminishing returns, showing only incremental changes at high K. Based on the above experiments, we choose $\rho = 0, K = 100$ for RavenMean and $\rho = 10, K = 1$ for RavenVar.

Effect of Load: The top row of Figure 3 shows the effect of steadily increasing load (the data generated per flow) on the four QoS metrics of interest. Before discussing the PDD, one needs to keep the PDR (Figure 3c) as reference since the PDD is calculated only for delivered packets. PDR decreases as load increases because the contact bandwidth is finite. RavenMean has the highest PDR since it has a very large K, while RavenVar has the lowest, since $K = 1$. All other protocols, except SprayWait, perform uncontrolled replication and hence suffer at high loads. At the reference value of 1GB, RavenMean has 3x the PDR of other protocols. The contact bandwidth between nodes is finite and limited and so, packets have to wait longer in the transmission queue at each node as the size of the queue increases along with the load. Therefore, the PDD increases with load (Figure 3a). The PDD for RavenMean is the more than other protocols (2x more) - but only because it delivers more packets. The other protocols have comparable PDR and thus have comparable PDD. Normally, the PDV is proportional to the PDD - but because of risk-aversion, RavenVar can force a lower PDV in exchange for a higher PDD (Figure 3b). Because of low K, RavenVar is unaffected by the load. Since all of the presented DTN routing protocols treat packets as independent and do not tune their replication based on the load, the energy consumed increases for increasing load (Figure 3d). RavenMean consumes 1.5x more energy than Rapid, but delivers 3x as many packets. MaxProp replicates while waiting for an ACK to clear a message from its buffer, causing it to have the highest energy consumption among other protocols.

Effect of Radio Bitrate: The effect of increasing radio bitrate on performance is shown in the bottom row of Figure 3. With increasing contact bandwidth, nodes can transfer more data during an opportunistic contact. The PDR increases almost linearly (Figure 3g). MaxProp takes maximum advantage of the increase, causing its PDR to appreciate by almost 4x. In comparison, RavenMean's PDR increases 1.5x because it has a fixed value of K that does not change with the workload. SnW3 is able to show only a 2x increase since its replication factor does not change, while Rapid and Maxprop show a 3x



(a) Effect of workload on PDD (b) Effect of workload on PDS (c) Effect of workload on PDR (d) Effect of workload on energy

(e) Effect of bitrate on PDD (f) Effect of bitrate on PDS (g) Effect of bitrate on PDR (h) Effect of bitrate on energy

Fig. 3: Behavior of the QoS metrics as (Top Row) load per flow changes and (Bottom Row) bitrate changes. Workload per flow was 1GB, speed multiplier was 1x and bitrate was 4MBps unless changed.

increase. The PDD (Figure 3e) decreases as expected but not for RavenVar since its K does not change. Its PDV (Figure 3f) is fairly constant but the PDV of other protocols decreases. As the contact bandwidth increases towards infinity, the PDV should decrease towards zero. The number of relayed messages (Figure 3h) increases since the node buffers are cleared more frequently owing to increased bandwidth, allowing more data to be moved in the DTN.

V. Conclusion

We have presented Raven, a DTN routing protocol that allows the user to control the PDV as well as other QoS metrics. Using stochastic multigraphs and the PDM mobility model, K least risk paths are chosen using the K Safest Paths algorithm. USAR hosts forward packets to each other based on the risk involved in delivering the message to their Center. Mathematical analysis of Raven shows several interesting properties that describe the coexistence of risk-aversion and replication. Raven is able to outperform other protocols based on metrics chosen by the user. When configured for minimizing PDD, it is able to deliver 3x more packets than other protocols.

Acknowledgements: This work was funded in part by NSF grants #1127449, #1145858, #0923203, #1216103.

References

[1] H. Chenji, W. Zhang, M. Won, R. Stoleru, and C. Arnett, "A wireless system for reducing response time in Urban Search & Rescue," IPCCC 2012.

[2] H. Zhu, L. Fu, G. Xue, Y. Zhu, M. Li, and L. Ni, "Recognizing Exponential Inter-Contact Time in VANETs," INFOCOM 2010.

[3] A. Balasubramanian, B. Levine, and A. Venkataramani, "DTN routing as a resource allocation problem," SIGCOMM '07.

[4] M. Khouzani, S. Eshghi, S. Sarkar, N. B. Shroff, and S. S. Venkatesh, "Optimal energy-aware epidemic routing in DTNs," MobiHoc '12.

[5] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks," WDTN '05.

[6] A. Vahdat and D. Becker, "Epidemic Routing for Partially Connected Ad Hoc Networks," Tech. Report CS-200006, Duke Univ., April 2000.

[7] M. Y. S. Uddin, D. M. Nicol, T. F. Abdelzaher, and R. H. Kravets, "A post-disaster mobility model for delay tolerant networking," WSC '09.

[8] M. Rubinstein, F. Ben Abdesslem, M. De Amorim, S. Cavalcanti, R. Dos Santos Alves, L. Costa, O. Duarte, and M. Campista, "Measuring the capacity of in-car to in-car vehicular networks," *IEEE Communications Magazine*, 2009.

[9] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Efficient routing in intermittently connected mobile networks: the single-copy case," *IEEE/ACM Trans. Netw.*, Feb. 2008.

[10] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," *SIGMOBILE Mob. Comput. Commun. Rev.*, Jul. 2003.

[11] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks," INFOCOM '06.

[12] A. Picu and T. Spyropoulos, "Distributed stochastic optimization in opportunistic networks: the case of optimal relay selection," CHANTS '10.

[13] D. Gunawardena, T. Karagiannis, A. Proutiere, E. Santos-Neto, and M. Vojnovic, "Scoop: decentralized and opportunistic multicasting of information streams," MobiCom '11.

[14] X. Tie, A. Venkataramani, and A. Balasubramanian, "R3: robust replication routing in wireless networks with diverse connectivity characteristics," MobiCom '11.

[15] J. Alonso and K. Fall, "A Linear Programming Formulation of Flows over Time with Piecewise Constant Capacity and Transit Times," Tech. Rep.

[16] M. Al-Siyabi, H. Cruickshank, and Z. Sun, "DTN QoS metrics and fair resources management model," CCECE 2011.

[17] M. Uddin, H. Ahmadi, T. Abdelzaher, and R. Kravets, "Inter-contact Routing for Energy Constrained Disaster Response Networks," *IEEE Transactions on Mobile Computing*, vol. Preprint.

[18] D. P. Bertsekas and J. N. Tsitsiklis, "An analysis of stochastic shortest path problems," *Math. Oper. Res.*, 1991.

[19] S. Lim, C. Sommer, E. Nikolova, and D. Rus, "Practical Route Planning Under Delay Uncertainty: Stochastic Shortest Path Queries," RSS 2012.

[20] P. Gupta and P. Kumar, "A system and traffic dependent adaptive routing algorithm for ad hoc networks," IEEE CDC, 1997.

[21] R. P. Loui, "Optimal paths in graphs with stochastic or multidimensional weights," *Commun. ACM*, Sep. 1983.

[22] E. Nikolova, "Approximation algorithms for reliable stochastic combinatorial optimization," APPROX/RANDOM'10.

[23] J. Y. Yen, "Finding the K Shortest Loopless Paths in a Network," *Management Science*, 1971.

[24] O. Barndorff-Nielsen, "On the Limit Behaviour of Extreme Order Statistics," *The Annals of Mathematical Statistics*, 1963.

[25] A. Ker, "On the Maximum of Bivariate Normal Random Variables," *Extremes*, 2001.