Towards a Formalism for Routing in Challenged Networks*

Ram Ramanathan
BBN Technologies
10 Moulton Street
Cambridge, MA 02138
ramanath@bbn.com

Prithwish Basu
BBN Technologies
10 Moulton Street
Cambridge, MA 02138
pbasu@bbn.com

Rajesh Krishnan
BBN Technologies
10 Moulton Street
Cambridge, MA 02138
krash@bbn.com

ABSTRACT

Research on challenged or delay/disruption-tolerant networks has exploded in the past few years with a plethora of algorithms targeted at different versions of the problem. Yet, there have been few formal studies on the fundamental nature of the routing problem in challenged networks. As a step toward closing this gap, we introduce a formal framework relating the problem and solution spaces in challenged networks. We define three fundamental types of challenged networks and several classes of routing mechanisms. We then prove a number of results on the power of each class of routing mechanism in terms of the network types that it can solve. We show that simple variants of MANET protocols can solve two but not all three network types. However, either complete schedule awareness or maximal replication is sufficient to solve even the most general type of challenged network. We extend these results to the bounded-storage and bounded-bandwidth cases. Finally, we briefly discuss a number of avenues in which the core formalism may be extended, including an infinite opportunity model and graph theoretic extensions.

Categories and Subject Descriptors

C.2 [**Computer-Communication Networks**]: Miscellaneous

General Terms

Theory, Design

Keywords

Challenged networks, Disruption Tolerant Networks, Formalism

*Supported by contract W15P7T-06-C-P638 under the DARPA DTN program.

CHANTS'07, September 14, 2007, Montréal, Québec, Canada.

1. INTRODUCTION

Recent years have seen the emergence of a new kind of multi-hop mobile wireless network characterized by severely challenged connectivity. These Challenged[1] Networks are different from Mobile Ad Hoc Networks (MANET) in that the network is disconnected as a rule rather than as an exception. This "turning around" of the connectivity assumption completely changes the solution landscape, and opens up exciting new areas of research. Solutions are applicable to military communications [17], inter-planetary networks [7], networks in under-developed areas [19, 1], or data exfiltration [6], to name a few.

Prior work in routing for challenged networks has focused almost entirely on algorithm development, leaving a need for a fundamental understanding of the problem to complement this development. In this paper, we examine the properties of the problem space (types of network challenges), and solution space (formal models of routing protocols) and derive results on the power of each class of routing protocols in solving the problem variants. Specifically, we classify challenged *dynamic networks* (DN) into *eventually connected* (ECDN), *eventually routable* (ERDN), and *eventually transportable* (ETDN), representing increasing levels of connectivity challenge. With regard to the solution space, we use a formal model of a Routing Mechanism (RM), that includes a *Forwarding* specification at each instant. We consider *path requiring*, *schedule aware*, *replicating* Routing Mechanisms, with and without storage and bandwidth limitations.

Within this formal framework, we derive several results. We show that a persistent path requiring RM can solve ECDNs and ERDNs but not ETDNs. The latter can be solved either by a schedule aware or maximally replicating RM. Our results lead to a determination, for each of the eight of RMs, which subset of the network types that combination can solve. We establish equivalences between storage requirements of single copy and replication-based routing mechanisms, and between storage and bandwidth. Further, if purging are allowed, we show a relationship between storage and delay diameter. Finally, we briefly discuss some directions in which our formal model can be extended, in particular Infinite Opportunity ETDNs, and ETDN graph theory, along with some conjectures.

Different *kinds* of formalisms are possible, and probably needed. Algebraic models, with basis in non-classic algebra [21], network calculus [3], and ordinary differential equations

[1]Also referred to by a number of other largely synonymous terms – Disruption/Delay Tolerant, Intermittently Connected, Opportunistic, etc.

[24] are some that exist. The formalism we propose is different, and along the lines of the theory of automata and formal languages [12]. In this theory, there exists a rich set of results on the power of each computational model (e.g. pushdown automata, finite state machines) in terms of the language it can accept/solve (e.g context-free languages, regular expressions). These results have resulted in a number of applications, notably in compiler design [12]. Analogously, we believe that a similar kind of formalism on the power of routing mechanisms vis-a-vis the solvable challenged network types could help develop better routing algorithms for challenged networks.

Many of our results are fairly straightforward, but we believe that the definitions, notation and results form a foundational framework which can be a starting point for a much richer and deeper formalism. As the first foray into this kind of formalism, we have interspersed proofs with intuition, informal discussion, and relations to standardized protocols, often spelling out details even though they may be obvious to the astute reader.

The remainder of this paper is organized as follows. In section 2 we define the network types and a notation for routing mechanisms. In section 3 we prove results on the solvability of network problems by routing protocol classes. In section 4 we focus on ETDNs and derive results for the bounded storage and bandwidth cases. Directions for future evolution of the formalism are discussed in section 5.

2. A FORMAL MODEL

In this section, we present some definitions and notation in order to characterize and classify network types and routing mechanisms.

We identify three types of challenged networks that differ in terms of the connectivity challenge they pose. Each is a different kind of *dynamic network*, defined below.

DEFINITION 2.1. *A* **dynamic network (DN)** *is a time-varying network. A DN at a particular instant of time t is denoted by* $G(t) = (V(t), E(t))$*, where t is a non-negative integer, and* $E(t) = \{ (u,v) : u,v \in V(t)$*, and can communicate at time t* $\}$.

The fact the time is discrete and not continuous makes it easier to argue about network properties, and should not take away anything from the expressive power as one can make the interval arbitrarily small. We also use G = (V,E) to denote the union of the graph/vertices/ edges over time.

The following defines the three types of dynamic networks that we shall study in this paper.

DEFINITION 2.2. *An* **Eventually Connected Dynamic Network (ECDN)** *is a DN* $G(t)$ *such that there exists one or more* T *for which* $G(T)$ *is connected.*

Thus in an ECDN, at some particular times you get a connected network. Note that this is different from a previous usage of "eventually connected" [5], but is truer to the words.

DEFINITION 2.3. *An* **Eventually Routable Dynamic Network (ERDN)** *is a DN* $G(t)$ *such that for every* $u, v \in V$*, there exists one or more* T_u^v *at which* $G(T_u^v)$ *has a path from u to v.*

Thus, an ERDN may never be connected, but we can rely on an end-to-end path being available between every pair of nodes at some point(s) in time.

DEFINITION 2.4. *An* **Eventually Transportable Dynamic Network (ETDN)** *is a DN* $G(t)$ *such that for every* $s, d \in V$*, there exists one or more sequences of times S* $= (t_1, t_2, t_3, ..., t_k)$*, where* $t_m \geq t_{m-1}$*, such that there is a path from* w_i *to* w_{i+1}*,* $i = 1 \ldots k-1$*),* $w_1 = s$*,* $w_k = d$*.*

Thus, an ETDN may never be connected and never have an end-to-end path at any given instant, but there is a temporally ordered link activation sequence that forms a path unioned over time, for every pair of nodes.

It is important to note that all of the above definitions, the connectivity, routability or transportability occurs one or *more* times – that is, *at least* once, not *exactly* once. Thus, the definitions and the results below apply equally to networks where the property occurs multiple times, including infinite times. However, for the particular case of infinite times, one may be able to prove additional results, as we discuss in section 5.

Examples of ECDN, ERDN and ETDN are illustrated in Figure 1. Figure 1(left) shows a partitioned network healed by a passing node. This is an ECDN, because at the point when the healing takes place, the network is connected. As another example, consider three battalions A, B and C that are on the corners of a triangle and mutually disconnected, as shown in Figure 1(center). A circular UAV trajectory connects, at t_i, (i = 1 to 3), battalions A-C, A-B, and B-C respectively This is an ERDN since the network is never connected, but there is an end to end path between any two nodes at some point in time. Finally, Figure 1(right) is clearly an ETDN because at no point is there a path between the nodes, but there exists a sequence of times as per definition 2.4.

Is there a subsumption order amongst these DNs? The following theorem confirms our intuition in this regard.

LEMMA 2.1. $S(ECDN) \subset S(ERDN) \subset S(ETDN)$*, where* $S(D)$ *denotes the set of networks of type* D*.*

PROOF. Consider a DN $G \in S(ECDN)$. Then, $\exists$ T such that $G(T)$ is connected. In a connected network, every node pair (x, y) has a path between them. Thus, $G(t)$ is an ERDN where $T_u^v = T$ is the same for all pairs (u, v), and therefore $G(t) \in S(ERDN)$.

Similarly, consider a DN $G(t) \in S(ERDN)$. Then, $\exists$ a time T_x^y when there is a path from x to y. Clearly, this is a particular case of definition 2.4 in which $S = (T_x^y)$ $s = x$, $d = y$. Therefore, $G(t) \in S(ETDN)$. $\square$

We now turn our attention away from the "problem" domain (networks) to the somewhat more abstract "solution" domain. The following definition classifies the set of routing algorithms to enable results about which class has the power to solve which DN problem.

We denote a *Routing Mechanism* by $\Re(a_1,\ a_2,\ \ldots\ ,\ a_m)$ where a_i is an *attribute* of $\Re$. An attribute denotes a certain choice of operation (e.g single copy) or a constraint on operation (e.g. limited storage) of $\Re$. The presence of an a_i in the parenthesis indicates that $\Re$ has that attribute, and the presence of $\overline{a_i}$ indicates that $\Re$ has the complement of that attribute.

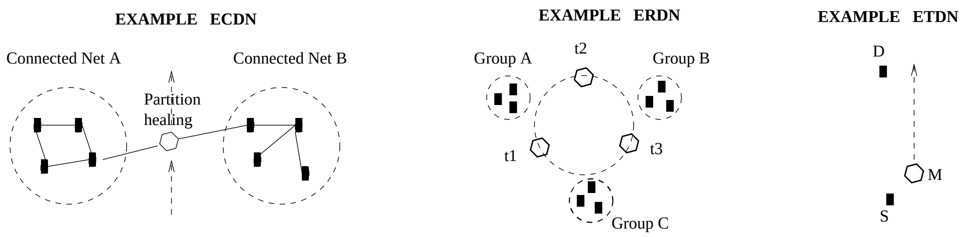

Figure 1: Partition healing (left) is an example of Eventually Connected Dynamic Network (ECDN), and a UAV sequentially connecting pairs of battalions (center) is an example of Eventually Routable Dynamic Network (ERDN). A message ferry (right) is a simple example of an Eventually Transportable Dynamic Network (ETDN).

The attributes we have defined and use are: *end-to-end path required* (PR) which means that $\Re$ requires an end-to-end path for packet transfer; *single copy* (SC) which means that $\Re$ does not replicate packets (intentionally); and *unavailable schedule* (US) which means that $\Re$ does not know in advance about all changes to links at all nodes in advance. Similarly, $\overline{PR}$, $\overline{SC}$ and $\overline{US}$ indicate the opposite, that is, no end-to-end path required, multi-copy routing allowed, and schedule availability[2], respectively.

Note that an attribute without a "bar" constrains the operation of the routing mechanism more than the same attribute with a "bar". We shall refer to the attributes without a "bar" (namely, PR, SC, and US) as *constrained* attributes and those with (namely, $\overline{PR}$, $\overline{SC}$ and $\overline{US}$) as *unconstrained*.

Thus, for example, $\Re(PR,\ SC,\ US)$ is a routing mechanism that requires an end-to-end path, and does not replicate intentionally, and does not have knowledge of future events (e.g. a conventional MANET protocol such as AODV). Epidemic routing is $\Re(\overline{PR},\ \overline{SC},\ US)$. Other constraints may be added as the framework evolves.

The basic unit of information transport is called a *bundle*, in deference to standard terminology in this field [8]. For the purposes of this paper, it is no different from the more well known *packet*.

At this point, it helps to have the notion of "solve" solidified. The following two definitions are crucial in this regard.

Definition 2.5. *Given a bundle B, a* Forwarding *is a mapping F: $(T \times V) \rightarrow A$ where T is a sequence of times, V is the set of vertices, and $A \in$ (* forw(x), forwcp(x), delete, keep*) is a set of actions meaning "forward", "forward but retain a copy", "delete the bundle", "keep the bundle" respectively.*

Definition 2.6. *A DN $G(t)$ is said to be* solvable *by a routing mechanism $\Re(a_1,\ a_2,\ \ldots\ ,\ a_m)$ (or equivalently, $\Re(a_1,\ a_2,\ \ldots\ ,\ a_m)$* can solve *$G(t)$) if and only if $\Re$ has a forwarding such that, every originated bundle B destined for D is delivered to D.*

Note that the above definition does not say that *every* routing mechanism with the listed attributes solves the given network problem, only that there is at least one mechanism that can solve it. On the other hand, if a network problem is not solvable by $\Re(a_1,\ a_2,\ \ldots\ ,\ a_m)$, then it means that no routing mechanism with the said constraints can solve it. We note that "solvable" here means delivery, no account is made of delay.

[2]This means *complete* schedule availability for every node for each and every time. Partial schedule availability will be regarded as Unavailable Schedule (US).

3. SOLVABILITY

We are now ready for some results. For the duration of this section, we assume that nodes have infinite storage capacity and links have infinite data rate. The next section relaxes these assumptions. We also assume that all networks considered have at least two nodes (for cleanness of exposition and not worrying about boundary conditions). Unless otherwise specified, results are for a single bundle that could be sourced anywhere and destined to anywhere and has infinite expiration time.

We begin by formalizing the intuitive notion of subsumption order among the DNs.

Lemma 3.1. *If a routing mechanism $\Re$ can solve ETDNs, then it can also solve ERDNs and ECDNs. If a routing mechanism $\Re$ can solve ERDNs, then it can also solve ECDNs*

Proof. By lemma 2.1 and definition 2.6. □

The above lemma helps us to state and prove theorems only for ETDNs rather than thrice, one for each of ECDN, ERDN and ETDN. We have another lemma with a similar purpose, and then we can move on to the less obvious ones.

Lemma 3.2. *Consider two routing mechanisms $\Re_1(\{a_i\})$ and $\Re_2(\{b_i\})$ such that either b_i is same as a_i or a_i is a constrained attribute and b_i the corresponding unconstrained attribute. In other words, $\forall i$ ($b_i = a_i$) OR (a_i is constrained AND $b_i = \overline{a_i}$). Then, given a DN $G(t)$*

(a) If $G(t)$ is solvable by $\Re_1$, then it is solvable by $\Re_2$

(b) If $G(t)$ is not solvable by $\Re_2$, then it is not solvable by $\Re_1$.

Proof. Suppose that F is a Forwarding using which $\Re_1$ solves $G(t)$. Clearly, F is also a valid Forwarding for $\Re_2$ since $\Re_2$ does not introduce any new constraints, only reduces constraints, and therefore $\Re_2$ can solve $G(t)$. Part (b) follows from the fact that $\Re_2$ contains every Forwarding that Re_1 does. □

We first consider the solvable power of the least capable RM $\Re(PR,SC,US)$. Recall that this class contains many traditional MANET protocols such as AODV and OLSR. Further, these protocols as specified are "non-persistent", that is, they drop a packet if there is no end-to-end path available to the destination. It is clear that these protocols cannot solve any of ECDN, ERDN and ETDN[3] since none of

[3]One could introduce, for completeness sake, an Always Connected Dynamic Network (ACDN) which can be solved by a MANET protocol, but we omit it to focus on "challenged" networks.

these network types are guaranteed to present a path upon packet arrival.

However, a *persistent* version of $\Re(PR,SC,US)$ can solve at least ECDN and ERDN, if we assume instantaneous topology/path availability.

LEMMA 3.3. *ECDN and ERDN are solvable by* $\Re$*(PR,SC, US).*

PROOF. First we show that $\Re(PR,SC,US)$ can solve ERDN. The proof is by existence, that is, by constructing a specific Forwarding that solves ERDN. Consider a bundle B originated at a node s, destined for d. Let T be the time at which the topology knowledge indicates that there is a path from s to d, say $(u_1, u_2, ..., u_m)$ where $s = u_1$, and $d = u_m$. Consider the Forwarding F:

$$
\begin{aligned}
F(t \leq T, s) &= keep \\
F(T, d) &= keep \\
F(T, u_i) &= forw(u_{i+1}) \\
F(T, v \in (V - d - u_i) &= drop
\end{aligned}
$$

In other words, F holds the bundle at s till the time T that the path becomes available and then instantaneously (because of infinite bandwidth and zero processing assumptions) has every node along the path send to the next node and other nodes drop it, if they get it. Clearly, F preserves single copy, does not use a schedule, and requires a path. It is also easy to see that the bundle is at d at T (by definition 2.3). Thus, ERDN is solvable by $\Re(PR, SC, US)$. By lemma 3.1, it also solves ECDN. □

We note that the proof can be modified in straightforward manner to relax the instantaneous topology/path availability. Specifically, it is easy to show that we only require that the topology be available *before* the path disappears, which in practice is a realistic assumption.

An interesting practical implication of Lemma 3.3 is that while MANET protocols cannot as such solve any of the challenged network types we consider, a simple modification where the forwarding is patient instead of dropping a packet can solve at least two of the three types. Many challenged networks might be ECDN or ERDN and it is probably much easier to modify a MANET routing protocol than invent brand new ones.

However, even a persistent version cannot solve ETDNs, as shown below.

LEMMA 3.4. *ETDN is not solvable by* $\Re(PR,\overline{SC},US)$*.*

PROOF. This follows from the fact that $\Re(PR,\overline{SC},US)$) requires a path and ETDN need not have a path. To be more formal, consider a three node network where S is the source of a bundle, D is the destination and M is a data mule from S to D. Clearly this is a ETDN by definition 2.4. Further, there is no time T for which this simple network has an end to end path, and $\Re$(PR,$\overline{SC}$,$\overline{US}$)) requires such a path[4]. □

The next result shows that even if you relax the path requiring condition, you still cannot succeed for ETDNs if you stick to single copies.

[4]Note that this is true even if all schedules were known, because $\Re$ is unable to do "custody transfer".

LEMMA 3.5. *ETDN is not solvable by* $\Re(\overline{PR},SC,US)$*.*

PROOF. We again prove by contradiction. Consider the DN in figure 2, referred to as $G1$. A train of n mules M_i comes in contact (has a link) with source S, at times t_i respectively, $1 \leq i \leq n$. The odd numbered mules also come in contact with destination D after a finite time, but the even numbered mules "veer off" and never ever are in contact with D. Two bundles B_1 and B_2 are sourced at S. B_1 is sourced between t_{k-1} and t_k, and B_2 between t_k and t_{k+1}, $2 \leq k \leq (n-2)$. Clearly, this is an ETDN since there exists a sequence of links as per definition 2.4.

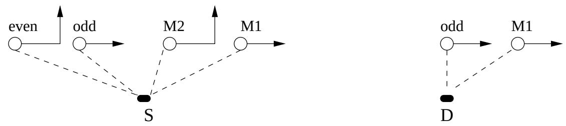

Figure 2: Mule train illustration for proof

Suppose $\Re(\overline{PR},SC,US)$ solves $G1$. Then, since $\Re$ is single copy, it must have chosen exactly one mule to have forwarded the copy to. Without loss of generality, suppose the Forwarding picks the m^{th} mule after sourcing. Then, B_1 will be forwarded to only mule M_{k+m} and B_2 only to M_{k+1+m}. Suppose without loss of generality that $k+m$ is odd, then clearly M_{k+1+m} is an even numbered mule and does not contact D and B_2 cannot be delivered. This contradicts the supposition of solvability of $G1$. □

So what *can* solve ETDNs? The following lemma shows that relaxing the single copy constraint to allow arbitrary replication does the trick. Specifically, we show that a non-path-requiring, multi-copy (a maximal copy to be specific) routing mechanism can solve ETDN.

LEMMA 3.6. *ETDN is solvable by* $\Re(\overline{PR}, \overline{SC}, US)$*.*

PROOF. We consider a specific Forwarding F^{MC} (MC for maximal copy) of $\Re(\overline{PR}, \overline{SC}, US)$, namely, one in which a node sends a copy of the bundle to every node that it can at every instant of time, unless it has it already. Formally, for a bundle B,

$$F(t, u \in V(t)) = forwcp(u_k), \forall (u, u_k) \in E(t), B \notin u_k \quad (1)$$

Consider a bundle B originated at $s = w_1$ and destined for $d = w_m$. By definition of ETDNs, there exists a sequence of nodes $(w_1, w_2, \ldots, w_m)$ such that at time t_i there is a link between w_i and w_{i+1}, that is, w_i can send (and by definition of F^{MC} *will* send) B to w_{i+1}.

We show by induction that F^{MC} solves ETDN. For the base case, note that at time t_1, s (w_1) sends the bundle to w_2 since it gives it to all nodes that it can send to at that time. For the inductive case, consider the bundle at node w_i, $i \geq 2$. The bundle is present at w_i at all $t_k \geq t_{i-1}$. By definition of ETDN, $\exists\, t_i \geq t_{i-1}$ when w_i can send to w_{i+1}, and since it has the bundle at all times $\geq t_{i-1}$, it has the bundle at t_i, and by operation of F^{MC}, the bundle is at w_{i+1}. Thus, ETDN is solvable by $\Re(\overline{PR}, \overline{SC}, US)$. □

The key to the above proof is the fact that a node continues to hold a copy, and unlike the single copy, is not forced to pick a node to hand it off to. Clearly, the infinite storage assumption is crucial here from a practical viewpoint, one that we shall seek to relax later.

PR	SC	US	What's solvable	Proof
Y	Y	Y	ECDN, ERDN	Lemmas 3.3, 3.4
Y	Y	N	ECDN, ERDN	Lemmas 3.3, 3.2, 3.4
Y	N	Y	ECDN, ERDN	Lemmas 3.3, 3.2, 3.4
Y	N	N	ECDN, ERDN	Lemmas 3.3, 3.2, 3.4
N	Y	Y	ECDN, ERDN	Lemmas 3.5,3.3, 3.2
N	Y	N	All three	Lemmas 3.7, 3.1
N	N	Y	All three	Lemmas 3.6, 3.1
N	N	N	All three	Lemmas 3.6, 3.2

Table 1: Summary table of which mechanism can solve which DN

Finally, an ETDN is solvable by a single copy routing mechanism if it has complete schedule information available because it can "guess right".

LEMMA 3.7. *ETDN is solvable by* $\Re(\overline{PR}, SC, \overline{US})$

PROOF. We consider a specific Forwarding F^{SA} (SA for Schedule Available) of $\Re(\overline{PR}, SC, \overline{US})$, namely, one in which a node sends the bundle to the next hop in an eventually transportable path. It can compute this path due to its knowledge of the complete schedule. Formally, for a bundle B,

$$F(t_i, w_i \in V(t)) = forw(w_{i+1}) \tag{2}$$

where t_i, w_i, w_{i+1} are such that w_i will be in contact with w_{i+1} at time t_i and w_{i+1} has a path to the destination of B in a finite time (note the recursive nature of this statement). Node w_i can determine w_{i+1} due to the complete topology awareness assumption coupled with the complete schedule availability property of $\Re(\overline{PR}, SC, \overline{US})$.

That F is sufficient to solve ETDN follows from the fact that, by definition, there exists in an ETDN at least one sequence of nodes (w_1, w_2, ... , w_m) such that at time t_i there is a link between w_i and w_{i+1}, that is, w_i can send B to w_{i+1}. And by definition of F^{SA}, it will forward along this sequence if it exists. □

Thus, to solve ETDNs, you need to use a non-path-requiring mechanism which is either allowed to replicate bundles or have full knowledge of future contact times. The intuitive argument that there needs to be enough "mixing" of nodes for these results to hold is covered by the definition of ETDN (see definition 2.4 which indirectly implies this.

We sum up our results with the following theorem.

THEOREM 3.1. *With reference to Table 1, a routing mechanism with attributes as in the first three columns can solve the dynamic networks as in the fourth column.*

PROOF. For a given row i in Table 1, the result is a direct consequence of the combined application of the lemmas in the last column of row i. □

4. STORAGE AND BANDWIDTH LIMITED ETDNS

We now focus solely on ETDNs. We relax the infinite storage and bandwidth assumptions and introduce a new storage attribute ST, and a "bandwidth availability" attribute BA (defined later below).

All references to storage are for network-wide storage, thus s is the sum of storages in all nodes[5]. Thus, a routing mechanism that is non-path-requiring, is single-copy, has no schedules, and a network wide storage of x is denoted by $\Re(\overline{PR}, SC, US, ST = x)$.

Since the path requiring condition is not acceptable for ETDNs even without storage constraints, it is obviously not going to work when storage constraints are imposed. Thus, in the below, we assume $\overline{PR}$ and omit it for brevity.

We need the following key definition to proceed further.

DEFINITION 4.1. *Let* $p(s, d) = w_1, w_2, \dots w_k$ *denote a path in an ETDN G between a source* $s = w_1$ *and destination* $d = w_k$*, such that* $\exists$ *a sequence of times* $S = (t_1, t_2, t_3, ..., t_k)$*, where* $t_m \geq t_{m-1}$*, such that* $(w_i, w_{i+1}) \in E(t_i)$*,* $i = 1 \dots k-1$) *(by definition 2.4 there exists at least one such path). Then,*

(a) The number of hops *in p is the number of nodes in the path excluding the source, that is,* $hps(s, d, p) = k - 1$.

(b) The delay *of p is the total time to go from source to destination, that is,* $dly(s, d, p) = \sum_{j=0}^{k-1} (t_{j+1} - t_j)$*, where* t_0 *is the time that bundle was originated.*

(c) The hop diameter *is the longest shortest-hop path, that is,* $\Delta_h = MAX_{s,d}\ MIN_p\ hps(s, d, p)$.

(d) The delay diameter *is the longest shortest-delay path, that is,* $\Delta_d = MAX_{s,d}\ MIN_p\ dly(s, d, p)$

The following shows that lack of future information (schedules) can be compensated for by replication even under the constrained storage model. Consider any window of m bundles sent from a source is released into the network at a single point in time T (continuous/flow-type comes in lemma 4.3).

LEMMA 4.1. *If* $\Re(SC, \overline{US}, ST = \psi)$ *can solve an ETDN G, then so can* $\Re(\overline{SC}, US, ST = \psi \cdot \frac{|V|}{\Delta_h+1})$.

PROOF. Consider one of the m bundles B sourced at s destined to d. Since $\Re(SC, \overline{US}, ST = \psi)$ can solve G, there must exist a path ($w_1 = s$, w_2, ... , $w_m = d$) such that at time t_i there is a path between w_i and w_{i+1}. Further, since by definition this must be true for any arbitrary pair (s, d), the worst-case path length is lower-bounded by the hop diameter Δ_h (we need to consider the worst case because it needs to solve any possible situation). Now, since the bundle needs to occupy storage at each hop plus the end nodes, and all this is true for each of the m bundles, we have

$$\psi \geq (\Delta_h + 1) \cdot size(B) \cdot m \tag{3}$$

By lemma 3.6 a maximal copy replication (refer F^{MC} in that lemma), can solve an ETDN. Let ψ' be the storage required by F^{MC}. Maximum replication can occupy no more than $\psi' = |V| \cdot size(B) \cdot m$ network-wide storage. Substituting for $size(B)$ from the above equation, we have $\psi' \leq |V| \cdot \frac{\psi}{\Delta_h+1}$. Thus, $\Re(\overline{SC}, US, ST = \psi \cdot \frac{|V|}{\Delta_h+1})$ can solve G. □

[5]It turns out that without any further assumptions a requirement of s network-wide storage implies a worst-case requirement of s per node since it may be that the dynamics and traffic origination require all bundles to be at one node at some point in time.

An interesting observation from the lemma is that the relative performance of a maximal copy mechanism is better for sparse graphs (when $\frac{|V|}{\Delta_h}$ is low) than for dense graphs. Since challenged networks are expected to be sparse, this is good news for replication-oriented schemes.

We recognize that our use of *network-wide* rather than node-specific storage is not practical since in a DTN it is not in general possible to do distributed buffer sharing. Nonetheless, we offer the lemma with the hope that future work may extend it to the more challenging node-specific buffers case.

We now turn our attention to bandwidth limited ETDNs. In particular, we consider the fact that the bandwidth available to transmit bundles over time is limited, which means that it takes non-zero time to transmit a bundle. We begin with the definition that captures the bandwidth and time factors involved.

Definition 4.2. *The required* **bandwidth-availability** *of a routing algorithm $\Re$ is defined as $BA = \sum_{e \in L_{\Re}} r_e t_e$ where $L_{\Re}$ is the set of links traversed by $\Re$, r_e is the datarate and t_e is the total time that link e is available and transmitting one bundle*[6].

In the lemma below, we show how the solving power of a single-copy schedule-aware routing algorithm with limited storage and unlimited bandwidth-availability (ones discussed in Section 4) is comparable to that of a multi-copy schedule-unaware algorithm with limited bandwidth availability.

Lemma 4.2. *If an ETDN $G(t) = (V, E(t))$ can be solved by $\Re(SC, \overline{US}, ST = \psi, BA = \infty)$, then it can be solved by $\Re(\overline{SC}, US, ST = \frac{|V|\psi}{\Delta_h+1}, BA = \frac{|E|\psi}{\Delta_h+1})$, where $|E|$ is the number of unique links in $G(t)$, and Δ_h is the hop-diameter of $G(t)$.*

Proof. As in Lemma 4.1, for m sourced bundles, the lower bound on storage in $\Re(SC, \overline{US}, ST = \psi, BA = \infty)$ is given by

$$\psi \geq (\Delta_h + 1) \cdot size(B) \cdot m$$

Consider an Epidemic algorithm F that forwards a copy of each bundle over every existing and new link. Since F can deliver a bundle successfully in an eventually transportable ETDN by transmitting the bundle over every available link in $G(t)$ *exactly once*, the bandwidth-availability of this algorithm is given by $|E| \cdot size(B) \cdot m$. Substituting for $size(B) \cdot m$ from above, and noting that F requires storage of at most $|V| \cdot size(B) \cdot m$, it follows that $\Re(\overline{SC}, US, ST = \frac{|V|\psi}{\Delta_h+1}, BA = \frac{|E|\psi}{\Delta_h+1})$ can indeed solve $G(t)$. □

4.1 Reducing Storage by Purging

We now consider the storage requirements of a multi-copy mechanism when bundles are constantly sourced over time. First, if bundles never leave the network (i.e., are never purged), this would clearly exhaust the storage quickly. We assume, therefore, that delivered bundles are purged using a purging mechanism that we consider as part of the routing mechanism. There are many ways in which purging can be done in practice (see for example [15]). We use an abstraction here for simplicity – a purge message is originated at the destination immediately upon delivery and is maximal-copy forwarded (epidemically) and occupies zero storage. Further, each bundle is of unit size[7].

Then, the following puts a bound on the storage needed to solve an ETDN without schedule availability.

Lemma 4.3. *$\Re(\overline{SC}, US, ST = \lceil \frac{2 \cdot \Delta_d}{\tau} \rceil \cdot |V|)$ can solve an ETDN with delay diameter Δ_d sourcing at most one bundle every τ seconds.*

Proof. Consider the Forwardings below, the first for a given bundle B and the second for a purge $P(B)$ of bundle B. The bundle is generated at source s and the purge is generated at destination d at the exact moment B reaches d.

$$\begin{aligned} F_B(t, u \in V(t)) &= forwcp(u_k), B \notin u_k, t \geq T_s(B) \\ F_{P(B)}(t, u \in V(t)) &= forwcp(u_k), P(B) \notin u_k, t \geq T_d(B) \end{aligned}$$

where $T_s(B)$ and $T_d(B)$ are the generation and delivery times, respectively, of a bundle/purge.

Since each bundle B can take no more than the delay diameter Δ_d time to be delivered, and each corresponding purge $P(B)$ can take no more than Δ_d to reach the nodes that contain B, the window of existence of B is upper bounded by $2 \cdot \Delta_d$. Within this time, at most $\lceil \frac{2 \cdot \Delta_d}{\tau} \rceil$ bundles can be generated. Each bundle can be replicated at most $|V|$ times, and therefore the total storage is bounded by $\lceil \frac{2 \cdot \Delta_d}{\tau} \rceil \cdot |V|$. □

This lemma may be applicable in practice to decide on the network-wide storage requirements if the delay diameter is known.

5. OTHER DIRECTIONS

We briefly discuss two directions in which the formalism of the previous sections can be "branched out", focusing more on possible foundational definitions and open conjectures than results. The first direction is a special case of the ETDN definition where infinite opportunities for transport exist. The second is the extension of graph theory to ETDNs with new definitions for concepts like "degree".

5.1 IO-ETDNs

We can extend our formalism to infinite opportunities to model networks in which there are unlimited number of opportunities to communicate over space and time even though a contemporaneous end-to-end path may not exist. This opens up the question of whether a single-copy strategy such as random walk might work (we showed earlier that a random walk strategy does not work if opportunities are limited).

We define a transport opportunity as a potential forwarding path from a source to a destination that is formed over space and time.

Definition 5.1. *A* **s-d Transport Opportunity** *in a DN $G(t)$ for a given $s, d \in V$, is a sequence of times $S =$*

[6]This metric captures the aggregate amount of link resources that were consumed for routing a bundle from source to destination. BA has units of bits. It follows that $BA = |L_{\Re}| \cdot size(B)$.

[7]It is easy to see that we still capture the essence of the relationship

$(t_1, t_2, t_3, ..., t_k)$, *where* $t_m \geq t_{m-1}$, *such that* $(w_i, w_{i+1}) \in E(t_i)$, $i = 1 \ldots k-1)$, $w_1 = s$, $w_k = d$.

Recall the definition of an Eventually Transportable Dynamic Network (ETDN) from Section 2. By definition a ETDN has at least a single s-d Transport Opportunity for every pair of nodes (s, d) in it. In this section, we will define a special case of ETDN. Unless otherwise specified, we assume that a ETDN has a finite number of nodes.

DEFINITION 5.2. *An* **Infinite Opportunities ETDN** *or* **IO-ETDN** *is an ETDN* $G(t)$ *such that for every* $s, d \in V$, $\exists$ *an infinite number of s-d Transport Opportunities.*

It is a well-known result that a random walk on finite static connected graphs is guaranteed to eventually visit every node. We prove below that this is not necessarily true for IO-ETDNs.

LEMMA 5.1. *A pure random walk does not solve IO-ETDNs.*

PROOF. Consider a four node chain (shown below) in which the middle link is alternatively on for a second and off for a second. The other two links are permanently on.

```
s-------x1-.-.-.-x2------d
```

Let us suppose it takes unit time for a bundle to traverse a link. Suppose a bundle from s destined for d is originated at $t = 0$. Using random walk, the bundle will arrive at $x1$ at $t = 1$ and find the link to $x2$ down. The bundle will therefore go back to s, and this cycle will repeat. The bundle never crosses over to x2. □

What if the random walk also used a random wait to escape the livelock? This is an open question, but we conjecture that it can still not guarantee to visit every node.

CONJECTURE 5.1. *A random walk with random wait at each node visited is not guaranteed to eventually visit every node in an IO-ETDN with probability* 1.

One way to prove this conjecture might be to construct an IO-ETDN that (even though it has a finite number of nodes) exhibits the same properties as an infinite lattice of three dimensions or more.

Finally, can any single copy algorithm (not restricted to random walk) solve IO-ETDNs? We do not know, but think not.

CONJECTURE 5.2. *An IO-ETDN is not solvable by* $\Re(\overline{PR}$, *SC, US).*

Disproving Conjecture 5.1 is one way to disprove Conjecture 5.2.

There is a rich set of such open problems in IO-ETDNs.

5.2 ETDN Graph Theory

Just as in graph theory, one can build a number of interesting results on the various properties/constraints of DNs, especially ETDNs. Since it is the ETDNs that are *not* ERDNs that are most interesting from a DTN viewpoint, we first define the notion of "pure ETDNs" before giving a couple of such graph theory type results.

DEFINITION 5.3. *A* Pure ETDN *(P-ETDN) is an ETDN that is not an ERDN (and so by Lemma 2.1) not a ECDN. Formally, S(P-ETDN) = S(ETDN) − S(ERDN).*

LEMMA 5.2. *Every P-ETDN is a multi-hop network, that is one in which* $\exists$ *at least one source-destination pair (s, d) such that a bundle from s to d has to go through at least one other node.*

PROOF. Consider a P-ETDN $G(t)$. Suppose to the contrary that the above theorem is false. Then, for every s, d, there exists a time, say T_s^d when s communicate with d. That is, at time T_s^d, there exists a path between s and d (for every s and d). But then this implies, by definition 2.3 that $G(t)$ is an ERDN which contradicts the pre-condition that $G(t)$ is a P-ETDN since by definition 5.3, P-ETDN cannot be an ERDN. □

The notion of the *degree* of a node needs to be redefined in the ETDN context.

DEFINITION 5.4. *The* instantaneous number of new contacts *of a node m (denoted by* con_m*) is the number of hitherto unseen neighbors that are now a neighbor. That is,* $con_m(t_i) = Nbrs_m(t_i) - \bigcup_j Nbrs_m(t_j))$, $j = 1 \ldots i-1$. *The* eventual maximum total contacts *of a DN* $G(t)$ *(denoted by* $\nu(G)$*) is* $MAX_m(Sum_i(con_m(t_i)))$.

This poses interesting solvability questions, for example:

CONJECTURE 5.3. *An ETDN with eventual maximum total contacts of at most two is solvable by* $\Re(\overline{PR}$, *SC, US)*

Other restrictions of ETDNs such as planarity, regularity [10] open up other interesting solvability questions. In general, rethinking the well known graph properties and algorithms in the ETDN context is an exciting direction.

6. RELATED WORK

There has been a surge in research on DTNs in the last few years, but nearly all of it has been focused on developing new routing strategies. These range from replication-oriented strategies such as Epidemic Routing [23], probabilistic forwarding and purging [11, 22, 13], and future contact prediction approaches [16, 4]. The use of network topology knowledge to increase the efficiency of routing has been studied in [14] and from a more practical viewpoint in [15]. A good survey of routing algorithms in DTNs is available in [25].

Theoretical models for networks in general have been developed, although along lines very different from this paper. In [21], a non-classic algebraic framework is used to formally investigate the convergence properties of routing protocols. A network calculus using a time-varying version of the (min, +) algebra is given in [3]. In [24] a unified framework based on Ordinary Differential Equations (ODEs) is used to study epidemic routing and its variations. The accommodation of time in the computation of shortest paths and other graph-theoretic problems has been studied as"evolving graphs"[9], and in terms of a space-time routing framework in [18], but neither addresses the solution space. More recently, [2] analyzes the performance of a deterministic online DTN routing algorithm with respect to knowledge (or lack thereof) of workload and schedule.

In relation to the above, our work is unique in at least two ways. First, we take a novel first-principles look at the problem and solution spaces, rather than propose a new algorithm. Second, unlike other theoretical models which are *algebraic* in nature ([24, 3, 21]), we focus on connectivity as the main discriminator and study the *solvability* (packet delivery) by various classes of algorithms. Our work is more in the style and spirit of the theory of automata and formal languages[12] and complements algebraic approaches.

7. CONCLUDING REMARKS

Disruption Tolerant Networking is rapidly gaining recognition as an important research area in mobile wireless networks. The field needs both principles and practical protocols as it evolves. This paper has taken a first step toward a formalism to help with the need for principles.

We have presented a formal framework for challenged networks and derived some preliminary results that can be built upon by other researchers. Classifying networks into whether they are eventually connected, routable or transportable, we have shown a hierarchy of solvability by routing mechanisms. In particular, we have studied three constraints on solutions, namely path requiring, schedule unavailability, and single copy, and enumerated the networks that are solvable under each of the eight combinations (Table 1). We then considered storage and bandwidth limitations and derived additional solvability results. We have also discussed directions in which the core formalism can be extended.

There is obviously significant scope for future work. First, the notational framework, in particular the use of "attributes" to characterize a routing protocol needs refinement for a cleaner exposition as we go forward. More network types and routing mechanisms could be added to the formalism along with associated solvability results. Results in section 4 need to be extended to node-specific storage for multiple simultaneous flows and incoroporate packet offered load into the results. As discussed in section 5 the formalism can be extended to IO-ETDNs, ETDN graph theory, etc. including possibly proving the conjectures mentioned. Finally, the development of efficient routing protocols motivated by the insights developed from the formalism is another direction to pursue. For example, such insights have already been instrumental in the design of a simple algorithm, described in [20] that hinges around storage efficient replication. Future development of the formalism are bounded to yield further useful insights.

8. REFERENCES

[1] Wizzy project. http://www.wizzy.org.za.

[2] A. V. A. Balasubramanian, B.N. Levine. Dtn routing as a resource allocation problem. In *Proc. ACM SIGCOMM*, Aug 2007.

[3] J.-Y. Boudec and P. Thiran. *Network Calculus: A Theory of Deterministic Queueing Systems for the Internet.* Springer, LNCS, 2001.

[4] B. Burns, O. Brock, and B. Levine. Mv routing and capacity building in disruption tolerant networks. In *In Proc. IEEE Infocom*, August 2005.

[5] D. Dolev and J. Welch. Crash resilient communications in dynamic networks. *IEEE Transactions on Computers*, 1997.

[6] P. J. et al. Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet. In *Proc. ASPLOS*, Oct 2002.

[7] S. B. et al. Delay-tolerant networking: An approach to interplanetary internet. *IEEE Communications Magazine*, June, 2003.

[8] V. C. et al. Delay-tolerant network architecture, July 2004. Internet-Draft.

[9] A. Ferreira. On models and algorithms for dynamic communication networks: the case for evolving graphs. In *In Proc. ALGOTEL*, 2002.

[10] F. Harary. *Graph Theory.* Addison Wesley, 1995.

[11] K. Harras, K. Almeroth, and E. Belding-Royer. Delay tolerant mobile networks (dtmns): Controlled flooding schemes in sparse mobile networks, 2005.

[12] J. Hopcroft and J. Ullman. *Introduction to Automata Theory, Languages and Computation.* Addison Wesley, 1979.

[13] S. Jain, M. Demmer, R. Patra, and K. Fall. Using redundancy to cope with failures in a delay tolerant network. In *SIGCOMM '05: Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 109–120, 2005.

[14] S. Jain, K. Fall, and R. Patra. Routing in a delay tolerant network. In *SIGCOMM*, Aug 2004.

[15] E. Jones, L. Li, and P. Ward. Practical routing in delay-tolerant networks. In *In Proc. ACM WDTN*, 2005.

[16] A. Lindgren, A. Doria, and O. Scheln. Probabilistic routing in intermittently connected networks. In *Proc. ACM Mobihoc*, 2003.

[17] P. Marshall. The disruption tolerant networking program, 2005. http://www.darpa.mil/sto/solicitations/DTN/briefs.htm.

[18] S. Merugu, M. Ammar, and E. Zegura. Space-time routig in wireless networks with predictable mobility. In *Georgia Tech Report GIT-CC-04-07*, March 2004.

[19] A. Pentland, R. Fletcher, and A. Hasson. Daknet: Rethinking connectivity in developing nations. *IEEE Computer 37(1), 78-83*, Jan 2004.

[20] R. Ramanathan, R. Hansen, P. Basu, R. Hain, and R. Krishnan. Prioritized epidemic for routing in opportunistic networks. In *Proc. ACM MobiOpp*, June 2007.

[21] J. Sobrinho. An algebraic theory of dynamic network routing. *IEEE/ACM Transactions on Networking*, Oct 2005. (13)5.

[22] T. Spyropoulos, K. Psounis, and C. Raghavendra. Spray and wait: An efficient routing scheme for intermittently connected wireless networks. In *Proc. ACM Sigcomm Workshop on DTN*, 2005.

[23] A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks, 2000.

[24] X. Zhang, G. Neglia, J. Kurose, and D. Towsley. Performance modeling of epidemic routing. In *In Proc. IFIP Networking*, 2006.

[25] Z. Zhang. Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: Overview and challenges. *IEEE Communication Surveys and Tutorials*, Jan 2006.