Prioritized Epidemic Routing for Opportunistic Networks[*]

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

Richard Hansen
BBN Technologies
10 Moulton St.
Cambridge, MA 02138
rhansen@bbn.com

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

Regina Rosales-Hain
BBN Technologies
10 Moulton St.
Cambridge, MA 02138
rrosales@bbn.com

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

ABSTRACT

We describe PRioritized EPidemic (PREP) for routing in opportunistic networks. PREP prioritizes bundles based on costs to destination, source, and expiry time. Costs are derived from per-link "average availability" information that is disseminated in an epidemic manner. PREP maintains a gradient of replication density that decreases with increasing distance from the destination. Simulation results show that PREP outperforms AODV and Epidemic Routing by a factor of about 4 and 1.4 respectively, with the gap widening with decreasing density and decreasing storage. We expect PREP to be of greater value than other proposed solutions in highly disconnected and mobile networks where no schedule information or repeatable patterns exist.

Categories and Subject Descriptors

C.2 [**Computer-Communication Networks**]: Network Protocols - *Routing Protocols*

General Terms

Algorithms, Design, Performance

Keywords

Opportunistic networks, Disruption Tolerant Networks, Epidemic Routing

1. INTRODUCTION

Recent times have seen the emergence of a new kind of mobile multihop wireless network known as DTNs (Delay/ Disruption Tolerant Networks), or ICNs (Intermittently Connected Networks), or *opportunistic networks*. The key distinguishing feature of a DTN[1] from a Mobile Ad Hoc Network (MANET) is that there may never be a contemporaneous end-to-end path, but the union of network snapshots over time may present an end-to-end path. Conventional MANET routing protocols typically drop packets in such situations and therefore are insufficient. Applications of DTNs include military communications [14], inter-planetary networks [6], networks in under-developed areas [15, 1], or data exfiltration [5].

We present a novel protocol for routing in DTNs called PRioritized EPidemic (PREP). The key idea behind PREP is to impose a partial ordering on the messages (called *bundles* following the DTNRG parlance [8]) for transmission and deletion. The priority function, which is slightly different for transmission and deletion, is based upon four inputs - the current cost to destination, current cost from source, expiry time and generation time. Inter-node costs are computed using a novel metric called *average availability*. Each link's average availability is epidemically disseminated to all nodes. As a result of this priority scheme, PREP maintains a gradient of replication density that roughly decreases with increasing distance from the destination.

PREP is derived from the recognition that Epidemic routing is unbeatable from the point of view of successful delivery as long as the load does not stress the resources (bandwidth, storage). Furthermore, unlike most existing works in the literature, Epidemic does not rely on extrapolating previous contact information. PREP uses the simplicty and power of Epidemic while fixing it in the one place that it is weak – high loads – producing a simple, yet robust and efficient protocol.

We evaluate PREP using simulation in *ns-2* and compare it to Epidemic Routing [18], and a representative MANET protocol, namely AODV. Epidemic and AODV represent two extremes in suitability for highly disrupted and stable networks respectively, and are therefore a good choice for relative comparison. Our results show that for the parameters studied, PREP's delivery ratio is about 1.4 times higher than that of Epidemic, and about 4 times higher than that of AODV. The gap between Epidemic and PREP further widens under storage limitations.

---

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

[1] We shall use the term DTN synonymously with the term opportunistic network in the remainder of this paper.

MobiOpp'07, June 11, 2007, San Juan, Puerto Rico, USA.

The remainder of this paper is organized as follows. In section 2 we briefly describe Prioitized Epidemic. Our simulation scenario and results are in section 3. After placing our contributions in the context of related work in section 4, we conclude in section 5.

2. PRIORITIZED EPIDEMIC

Before describing PRioritized EPidemic (PREP) in detail, we briefly discuss the key idea and rationale behind it. First, we argue that from a practical viewpoint, Epidemic Routing[2] [18] is essentially unbeatable in terms of delivery ratio when the load relative to resources (bandwidth and storage) is low, or when the network is very sparse (highly disconnected). In other words, it is only when the relative load is high in stable networks that Epidemic underperforms both MANET protocols and sophisticated disruption tolerant protocols that intelligently route using a single copy.

Thus, the question is how far can we get if we simply fix the one place Epidemic Routing is weak in – when it is forced to drop packets due to storage or bandwidth considerations. That is, if we can determine which of the copies of a packet in various nodes in the network are more "droppable" than others, we can significantly reduce the burden on resources without compromising too much on delivery. This line of thinking motivates a method for assigning relative priority among bundles for dropping and transmitting that is only engaged when necessary. While a number of schemes can be used to assign this priority, it is clear that the expiry time and the distance from destination should figure high on any list – the latter stemming from the following observation: Given two copies one closer in cost to the destination than the other, it is preferable to drop the more distant one as it has to work harder to reach the destination[3].

In a sense, PREP reverses the typical packet forwarding approach. Instead of keeping the storage and bandwidth utilization at a low level and replicating only when necessary (retransmisisons, multipath routing), PREP keeps the storage and bandwidth maximally utilized, dropping only when necessary.

PREP consists of two modular, independent components: a topology awareness scheme that enables estimating the routing cost from a given node to the destination, and a priority scheme for bundle processing (deleting and transmitting). We describe each of these below.

2.1 Topology Awareness

Each node executes a neighbor discovery algorithm to create and maintain a set of bidirectional links with other nodes (neighbors). This is done using a HELLO protocol similar to that in [17].

Attached to each link is a novel metric called *average availability (AA)*. The AA metric attempts to measure the average fraction of time in the near future that the link will be available for use. AA is a dimensionless quantity, but may be multiplied by the data rate to determine the average link capacity over time. AA is calculated as follows: Let $T_i = \min(T_a,\ T_m)$, where T_a is the time in the most recent past for which up/down information is available and T_m is a configured "max window" time. Let T_{up} be the total time within T_i that the link was "up". Then, the average availabilty for discovered links is defined as

$$AA_{ds} = \frac{T_{up}}{T_i}$$

If the link has been down for more than a configured T_g seconds (T_g could be same as T_m), the link is "forgotten". If and when the link appears again, it is considered a "new" link and therefore T_a is set to the amount of time that the link has been up (this is the most recent past since the old past has been forgotten). This will make the AA jump to nearly one immediately[4].

When a node notices a threshold change in a metric for one of its links, a *link state advertisement* (LSA) message listing all current links and their metric values is generated and given an incremented version number. The LSA is disseminated to other nodes in the network using *epidemic routing* [18]. Specifically, every node initiates a topology "sync" with a neighbor (defined as a minimal exchange of LSAs such that the topology awareness of both nodes are identical) when either the AA increases or decreases by a threshold amount or a sync was recently executed with another neighbor. To minimize overhead, only one sync per neighbor is allowed in every given time window.

The dissemination capability of the above method is much better than pure flooding in frequently disrupted networks. Specifically, knowledge of links whose change was disseminated prior to a particular event, but is not part of the knowledge of a node is updated to the node immediately. For example, when two partitions are joined, each node in each partition quickly learns the entire current connected topology. This does not happen in traditional flooding.

By virtue of the LSA dissemination, each node has some knowledge of the network topology. In particular, a node N will have knowledge of the subgraph consisting of nodes from which an LSA was transportable to N during some recent time period. This "best effort" topology awareness is used to compute routing costs. To do this, first a cost is assigned to each link as a function of the average availability: $(1 - AA) + 0.01$ (the small constant factor makes routing favor fewer number of hops when all links have AA of 1). Then, Dijkstra's shortest path algorithm [3] is used to find the lowest-cost route.

2.2 Bundle Drop and Transmit Priority

Each bundle is assigned a drop priority p_d, and a transmit priority p_t in a manner that will be described later in this section. Both p_d and p_t can be any real number, a lower value indicating higher priority. When a contact C comes up, bundles with a lower value of p_t are sent before bundles with a higher value of p_t. When the allocated buffer for bundles is full (or nearly full), bundles with a higher value of p_d are deleted before bundles with a lower value of p_d. In both cases, ties are broken randomly.

2.2.1 Bundle Drop Procedure

---

[2] In Epidemic Routing, a node A "infects" every contact B with packets that it has that B doesn't have. A summary vector is typically exchanged to determine the missing packets

[3] The issue here, of course, is how to determine the "distance", and this is the subject for a large part of our discussion of PREP

[4] Note that if T_a is not used, that is, the window is a constant, a link reappearing for a short time will get a low AA and hence may not get packets routed to it before it disappears

Each bundle has a field that contains the *hop count*, i.e., the number of hops the bundle has traversed thus far. The source initializes the hop count to zero and every node that receives a bundle increments the hop count field. Each node maintains a *low water mark* and *high water mark* for buffer occupancy. A drop procedure is initiated whenever the buffer occupancy exceeds the high water mark. It is stopped when the buffer occupancy falls below the low water mark. The use of two marks prevents the dropping procedure from being constantly invoked.

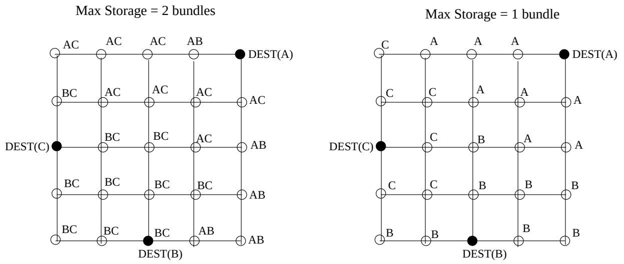

Figure 1: Example illustrating PREP drop procedure with three bundles A, B, C in transit with destinations as shown, on a network with storage limited to 2 bundles per node (left) and 1 bundle per node (right). PREP clusters bundles around their destinations and thereby improves delivery probability.

The drop priority of bundles is calculated as follows. We first take bundles that have a hop count value greater than or equal to a configured threshold V_{hc}. For these bundles, we give lower priority to bundles that have a larger shortest path cost to the destination as determined by the route computation procedure described earlier. Specifically, the priority p_d(B) of a bundle B is equal to the cost of the lowest-cost path from the current node to the bundle's destination D.

The bundles are deleted in decreasing order of p_d(B) until either the buffer occupancy falls below the low water mark or all taken bundles have been deleted. If all (remaining) bundles have a hop count value less than V_{hc}, and the buffer occupancy is still above the low water mark, these bundles are deleted randomly.

This bundle drop procedure has the effect, during a buffer crunch, of building a gradient of replication density centered at the destination of the replicated bundles. In other words, more copies are maintained near the destination, letting copies further away be deleted. An example of the distribution of bundles due to PREP's operation is illustrated in Figure 1.

The use of hop count allows bundles to get a head start before being considered for deletion. A similar problem, with a similar solution, is discussed in [4].

2.2.2 *Bundle Transmission Procedure*

A transmission procedure in PREP is invoked whenever contact is made with a peer to whom no transmission has taken place in the last T_{sync} seconds. The transmission procedure consists of the usual epidemic exchange of missing bundles (see [18]). Briefly, summary vectors describing content are exchanged and then missing bundles are sent from one to the other. Which bundles are transmitted first in either direction depends upon the transmit priority assigned to them at the holding node.

Parameter	Value
Nodes	25
Mobility	Random Waypoint
Speed	5 - 15 m/s (uniform distr)
Approx range	250m
Data rate	1 Mbps
Prop model	2-ray ground
Simulation time	20 minutes
Area (sqkm)	1.0 - 9.0
Load (bytes/sec/node)	40 - 200
Payload size	1000 bytes

Table 1: Parameters used in our simulations.

Tranmission priority is based on two factors: whether the peer to which transmission is to be done has a smaller cost to the destination, and on the time-to-expire of the bundle. Specifically, suppose the transmission procedure is invoked for a peer P for a bundle B whose destination is D. Then, if the cost of the shortest path from P to D is less than or equal to the cost of the shortest path from the current node to D, then B is placed in a "downstream" bin, else it is placed in an "upstream" bin.

The downstream bin is selected first and processed completely before going to the upstream bin. For each bin, the priority $p_t(B)$ of a bundle B is equal to its ranking in a radix sort on (expiryTime(B)$-$currentTime) and (creationTime$-$currentTime). Radix sort (or lexicographic/postal sort) sorts first by the most significant element (former) and then by the next most significant element (latter) and so on (only two in our case). Ties are broken randomly. We note that in the case that expiry times are not given (infinity), the bundles that have been around longer get higher priority (note that (creationTime $-$ currentTime) is negative integer, this is deliberate).

3. SIMULATION RESULTS

We present a comparative simulation analysis of PREP with Epidemic routing and Ad Hoc On Demand Distance Vector (AODV). Epidemic and AODV[5] represent two extremes in suitability for highly disrupted and stable networks respectively and therefore make a good choice for relative comparison.

We have studied the performance of PREP using the *ns-2* simulator. We implemented the Epidemic protocol based on [18] but with some performance enhancements. We built PREP over the Epidemic protocol by introducing the drop and transmit priority before handing over the bundle to Epidemic. The Topology Awareness mechanism as described in section 2.1 was also implemented over Epidemic - that is, the link state updates are generated and given to the Epidemic module for dissemination. We have used the AODV protocol implementation "as-is" from the simulator for comparison.

The parameters for the simulation are given in Table 1. Most other parameters us *ns2* defaults. Nodes were generated randomly in a specified square of area A and move according to the well-known Random Waypoint mobility

---

[5] We chose AODV as a representative of a "typical" MANET protocol

model. Bundles with 1000-byte payloads are sourced at an average of b bytes/second at every node with a randomly chosen destination. For our studies, we focused on the dependence on A, b, and the amount of storage in the node.

Due to space restrictions, we shall only report on the performance with respect to two metrics: the *delivery ratio* which is the fraction of sourced bundles that are delivered to the destination; and the *delay CDF* which is the fraction of bundles that are delivered within a given delay bound.

Figure 2 shows the dependence of delivery ratio on the area of operation, for AODV, Epidemic and PREP. Increasing the area makes the connectivity more sparse and increases the level of disruption. While all three mechanisms are affected by disruption, AODV is affected the most, as expected. At a representative value of Area=5, PREP's delivery ratio is 20 percentage points (42%) better than that of Epidemic and 54 percentage point more than AODV (400%). The initial rise for PREP can be attributed to the fact that at area 1 the reduced spatial reuse is more dominant.

Figure 3 shows the dependence of delivery ratio on the load for an area of 9.0 sq km. AODV cannot handle the disruption at area 9.0 sq km and is only finding routes for a handful of destinations. Thus, increasing the load has no effect (since to those destinations it can comfortably handle the load). PREP does better than Epidemic, with the gap widening with increasing load as the greater efficiency of PREP is magnified. We note that this is despite the fact that PREP injects a fair amount of routing control packets for topology awareness.

Figure 4 shows the cumulative delay function (CDF). Delays for PREP and Epidemic are similar for the bundles that both protocols can deliver. AODV can deliver its bundles in shorter time due to immediate forwarding, but delivers far fewer bundles.

The effect and value of PREP's bundle drop procedure (see section 2.2.1) is illustrated in figure 5 which shows the dependence of delivery ratio on storage.[6]. As the storage decreases, Epidemic's delivery falls approximately linearly whereas PREP's remains almost flat, with a factor of 7 difference at low storage! This is due to the bundle drop priority procedure that ensures that valuable bundles are not dropped. In contrast, Epidemic drops bundles arbitrarily.

4. RELATED WORK

The simplest solution to the DTN problem is brute-force unconstrained replication, or Epidemic Routing [18]. A number of ideas have been explored to improve the efficiency of Epidemic Routing, including probabilistic forwarding and purging [9], controlled replication called "spray and wait" [16], and using portfolio theory (Sharpe Ratio) and erasure coding [10].

Several works make use of contact history to predict future contacts to guide routing decisions, including [13] (PROPHET), [2] (MV), [4] (MaxProp), and [7]. The approach taken is to compute a (transitive) delivery probability based on how often nodes meet each other. MaxProp additionally uses priorities to forward to peers and select packets to drop.

The use of network topology, traffic patterns etc. to increase the efficiency of routing has been studied in [11],

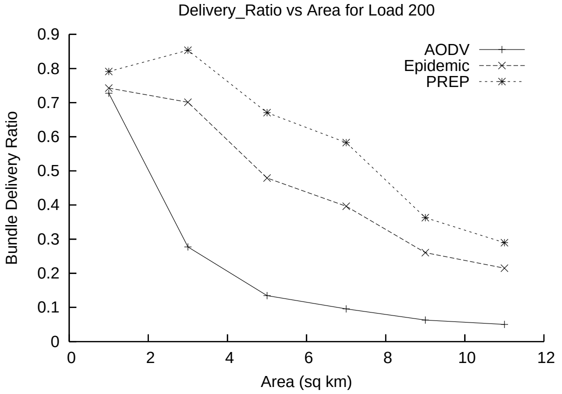

Figure 2: Percentage of bundles delivered as a function of area

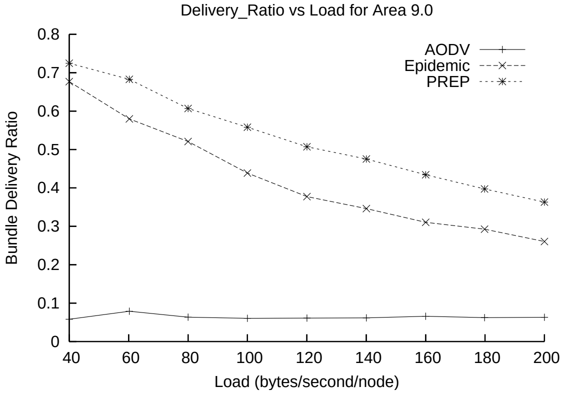

Figure 3: Percentage of bundles delivered as a function of load

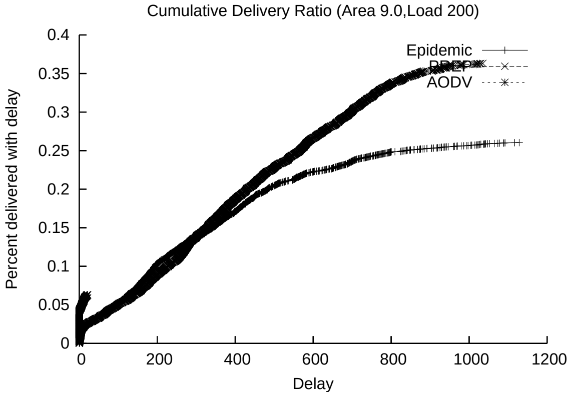

Figure 4: Cumulative Percentage of bundles delivered within a given delay

---

[6] The x-axis shows the high water mark for PREP and the maximum storage for Epidemic, which is slightly higher.

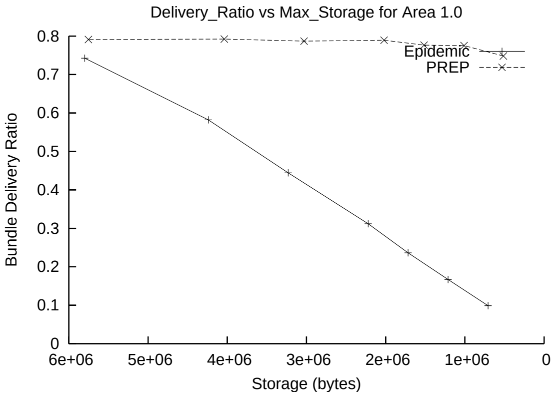

Figure 5: Percentage of bundles delivered as a function of maximum storage available per node

where the DTN problem is formulated rigorously and a number of progressively information-intensive algorithms are studied. A more practical treatment is given in [12]. Other works, such as [20], [21] explore various other aspects like ferrying, multicasting, etc. A good survey of routing in DTNs is available in [19].

In relation to the above, our work is unique in a number of ways. First, unlike many existing protocols, PREP does not rely upon extrapolating from contact history and therefore can be applied to a wide variety of scenarios. Second, unlike the other Epidemic variants and priority-based approaches it uses routing costs derived from topology updates to prioritize bundles. Finally, our simulation work examines the impact of storage limitations in addition to the impact of disruption level and load.

5. CONCLUDING REMARKS

Disruption Tolerant Networking is rapidly gaining recognition as an important research area in mobile wireless networks. Research on routing for such networks has exploded in the last couple of years. However, while there has been no dearth of sophisticated (and complicated) solutions, there has been no proper examination of the power of variants of Epidemic Routing.

We have presented a novel mechanism called *Prioritized Epidemic (PREP)*. PREP uses expiry time information and topology awareness to decide which bundles to delete or hold back when faced with a resource (buffer, bandwidth) crunch. Our simulation results have shown that PREP outperforms both Epidemic and a representative MANET protocol by a significant margin.

Our future work in this area involves the further development of a formal framework for DTN routing, making PREP a more adaptive protocol, and the inclusion of a non-replication component under stable network conditions. We are also currently implementing a highly modular DTN system based on our routing algorithm, and plan to evaluate this on our testbed in the short term.

6. REFERENCES

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

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

[3] E.W. Dijkstra. A note on two problems in connexion with graphs. *Numer. Math. 1:269, 271*, 1959.

[4] J. Burgess et al. Maxprop: Routing for vehicle-based disruption tolerant networks. In *In Proc. IEEE Infocom*, 2006.

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

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

[7] T. Spyropoulos et al. Single-copy routing in intermittently connected mobile networks. In *In Proc. IEEE SECON*, 2004.

[8] V. Cerf et al. Delay-tolerant network architecture, April 2007. Internet RFC 4838.

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

[10] Sushant Jain, Michael Demmer, Rabin Patra, and Kevin 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.

[11] Sushant Jain, Kevin Fall, and Rabin Patra. Routing in a delay tolerant network. In *SIGCOMM*, Aug 2004.

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

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

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

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

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

[17] P. Jacquet T. Clausen. Optimized link state routing protocol (olsr), October 2003. Internet RFC 3626.

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

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

[20] W. Zhao, M. Ammar, and E. Zegura. A message ferrying approach for data delivery in sparse mobile ad hoc networks. In *In Proc. ACM Mobihoc*, 2004.

[21] W. Zhao, M. Ammar, and E. Zegura. Multicasting in delay tolerant networks: Semantic models and routing algorithms. In *In Proc. ACM Workshop WDTN*, 2005.