

Questioning Flooding as a Routing Benchmark in Opportunistic Networks

Muhammad Arshad Islam* Marcel Waldvogel*

* Distributed Systems Laboratory
University of Konstanz – Germany
firstname.lastname@uni-konstanz.de

Abstract—Routing in Opportunistic Networks, as a relatively young discipline, still lacks coherent, simple and valid benchmarks. It is customary to use epidemic routing as performance benchmark for Opportunistic Networks. We identify and describe the current simulation practices that do not expose the shortcomings of flooding as an upper bound. In this paper to provide a step towards a routing benchmark, which is flexible, provides results close to an upper bound, is simple to implement, and thus might be a candidate for a common benchmark. This new method called EPO¹, does not suffer from bottlenecks that limit the performance of epidemic flooding, even when bandwidth is scarce. Our analysis shows that networks are not suffering from that much severe congestion as suggested by flooding and thus giving a better insight to the underlying network.

I. INTRODUCTION

Since initial introduction of Delay Tolerant Networks in research horizon for interplanetary communication[?], several offshoots have spawned e.g. *Vehicular Networks*, *Mobile Social Networks* and *Opportunistic Networks*. Similarly, several practical application such as emergency response in case of a catastrophe, military operations and non-interactive Internet access in rural areas[?] have accumulated the usability of such networks. Although, there have been a few practical deployment of opportunistic networks[?], [?], simulation is still the favorite tool assisting us in analyzing opportunistic networks with several variations, e.g. movement of devices, variable bandwidth, obstacles, environmental effects etc. In a real world scenario, where repetition of events is very unlikely, scientist apply simulations techniques on data that is either created artificially or is obtained from a real world phenomena. This helps a great deal to improve and refine the protocols. We have observed that several efforts have been attempted and many are still underway to develop an optimized routing protocol for Opportunistic Networks. Depending on how the mobile nodes move and what characteristic a node possesses, how dense the node population is, and how far sender and receiver are apart and furthermore how big the messages are, message delivery times may vary substantially from a few minutes to many hours or days. In other words, communication performance in such scenarios not only depends on the routing and forwarding algorithms in use but also on the topological structure of the network.

Irrespective of whether the simulation incorporate real life traces or an artificially generated network for such asynchronous path scenario, almost everyone is benchmarking his/her results against the flooding protocol. It is mostly argued that flooding can deliver optimal performance provided the overhead is ignored. We argue that routing algorithms, including flooding, may scale differently with variation in network resources as discussed above. To assess the performance accurately of routing algorithms in any opportunistic networks, it is, therefore necessary to have such performance benchmarks that scale uniformly to the available resources and have the capability to perform better than other protocols, thus deserving to be an upper-performance bound. We identify the situation where flooding may fail to perform and based on its deficiency, we devise a new algorithm known as EPO that may be seen as a first step towards more accurate performance benchmark. EPO utilizes concepts of oracles as presented by Jain, Fall and Patra[?] but deploys them in its own peculiar way. These subtle changes ensure that EPO will scale appropriately with small changes in available bandwidth. Furthermore, we argue that as the networks in question are constituted from heterogeneous devices with huge variation in communication and storage capabilities, therefore it is not easy to ignore the resource aspect while benchmarking in such networks.

II. CURRENT TRENDS

Although, simulation is a powerful tool to analyze any phenomena and its side effects that are either not in our control or hard to repeat, the reliability of the results gathered are highly dependent on environments and parameter used to simulate. To understand opportunistic networks, traditionally, the underlying network is obtained from either real life traces or artificial network generator[?] that can be trusted to obtain real life compatible traces. In several cases, artificially generated networks [?], [?] have supposition such as, bounded area, node count, node speed/velocity, node coverage area and a mobility model for the nodes[?], [?], [?]. Despite of the fact that these variables can be tuned to create dense or sparse networks, Islam and Waldvogel[?] have shown that artificially generated simulation environments are not a good replacement for real world scenario. In a real life scenario, it is probable that not all the devices are available for the whole period of

¹Inspired by use of Erythropoietin as Performance enhancement drug

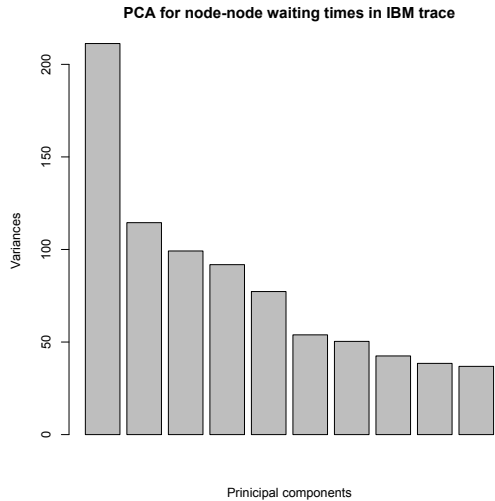


Fig. 1. PCA for Dev-Dev Waiting times in IBM Network

simulation, i.e. new nodes join the network in the middle of trace period. Existing nodes leave the network in the middle of trace period. Moreover, different segments of network not only have time dependent but highly irregular contact patterns[?]. As discussed earlier in section ??, we can find a few real life deployments of opportunistic networks[?], [?], all of them receive either the assistance by inserting Internet-Connected data mules and/or fixed contact schedules such as, timings of public transportation system can be predicted with high probability.

We argue that it is not easy to find a particular contact behavior or pattern among the nodes in real life traces. We have analyzed three different networks obtained from real life traces that have considerable variations among network size and density. Details of the traces are presented in section ??. We extracted different metrics such as contact waiting times, durations, frequencies to find traces of any regular patterns but none of the metrics analyzed helped us doing so. We used *Principal Component Analysis* PCA on the dissimilarity metrics mentioned above, expecting that the first few components accounting for most of the variability. The presence of strong components could be an evidence that the majority of nodes in the networks can be classified in a few categories as far as contact patterns are concerned. We considered only office timings i.e. 0900-1700 hrs for experimentation, assuming that variations during the office hours are bound to be lower than evening hours, and we performed PCA on all the above mentioned metrics, unfortunately without any hint of patterns as shown in Fig. ??.

As stated earlier, to assume that all the participating nodes in opportunistic networks have the same communication capabilities is impractical. In wireless scenarios, traffic volume plays a very critical role [?] therefore, scalability (with regard to traffic volume as well as network size) of a routing protocol must be considered. If we assume that all the transmission among the devices is comprised of same volume, and network size does not go through major changes, it may lead to

Fig. 2. Conceptual performance vs knowledge trade-off for different oracles.[?]

unreliable conclusion. Ignoring the traffic volume, specially in the case of shortest path or single copy strategies may result in impractically optimistic results. Lee and Geria[?], [?] have presented arguments that repeated utilization of the shortest path will amplify the congestion: in a random communication pattern, the nodes in the center of the network carry a disproportionately large amount of the entire traffic, drastically decreasing the throughput of the flows they forward or even temporarily resulting in a disconnected network. This affects most long-range flows, as they have a higher probability of intersecting the central hot spot.

A. Epidemic routing as benchmark

Given the myths about the performance of flooding in opportunistic networks, it has been customary to benchmark new protocols against flooding[?], [?], [?]. In all the cases, a protocol is declared to be a winner, efficient or better if it performs close to flooding. As discussed in section ??, flooding is notorious for its overheads pertaining to bandwidth and local storage consumption. We would like to present a few contradictory examples where flooding fails to perform as expected and thus losing its credibility as an upper bound. (i)Let us assume that a device at the edge of a network that has very few links, wants to flood the network with a message but none of its links are good enough for the message to be replicated to any other device. If this device has other smaller messages, it has to deploy a smart scheduling technique so that those messages are given propagation priority to the next hop. Conversely, flooding has no availability of such an oracle that can predict the duration of the next contact. Another way of solving the issue could be to use the resume feature, i.e. the message is fragmented according to the available contact durations and transmission is resumed from that point onwards at which the two devices have the next contact. We have to face other questions and complications if we follow this approach, e.g., may the next device propagate further the partial fragment any further? May the next device obtain the remaining fragment from another device? if yes, every device may end up with several non contiguous fragments of the message and to figure out an optimized solution which fragment hole to fill first, is a complicated question on its own. (ii)In another scenario, if a node is located close to the center of the network and is equipped with state of the art communication hardware (large range of channels, good signal strength, high bandwidth etc), such a node can handle the throughput that flooding presents. If the premise of good hardware is not there, even flooding may fail to reach the whole network in spite of the availability of sufficient network resources unless it is used in a better way as shown below. Such scenarios shows that flooding can also fail miserably to perform given the simulation account for more realistic conditions and that we need to improve on our performance benchmarks.

Based on above arguments, we feel the need for a better benchmark for opportunistic networks that adapts to underlying network. Jain et al [?] have proposed several oracles based algorithms as depicted in Fig. ??, starting from First Contact (FC) that has no knowledge to *Linear Programming* (LP) that utilizes the maximum knowledge including contact summary, local queuing, and traffic demand. [?] has also presented a modified Dijkstra's algorithm presented in Alg. ?? that computes a path given a time varying graph. This algorithm assumes availability of the whole time varying graph that is equivalent to knowing the future patterns of devices in an opportunistic network.

```

Input:  $G = (V, E), s, T, w(e, t)$ 
Output:  $L$ 
 $Q \leftarrow \{V\};$ 
 $L[s] \leftarrow 0, L[v] \leftarrow \infty \forall v \in V \text{ s.t. } v \neq s;$ 
while  $Q \neq \{\}$  do
  Let  $u \in Q$  be the node s.t  $L[u] = \min_{x \in Q} L[x];$ 
   $Q = Q \leftarrow \{u\};$ 
  for each edge  $e \in E, \text{ s.t. } e = (u, v)$  do
    if  $L[v] > (L[u] + w(e, L[u] + T))$  then
       $L[v] \leftarrow L[u] + w(e, L[u] + T);$ 
    end
  end
end

```

Algorithm 1: Dijkstra's Algorithm modified to use time varying edge costs. [?].

Inspired from this hierarchy, we propose an algorithm EPO that assumes an oracle that has information about future contact timings, size of local queues at devices and future traffic demands of individual devices. EPO uses the same method as presented in Alg. ?? to find the shortest path but utilizes the computed shortest path in an unorthodox manner. The idea is to find the shortest path to the destination for one message and propagate that one message at a time through the network. Unless the message reaches the destination or no shortest path is available, propagation of other message are temporarily suspended. A question that may be asked here is, what if the shortest path is not good enough to deliver the message? It may be the case that the duration of overlap times of at least one hop are not long enough to transfer the message to the next hop and the whole chain of hops in the shortest path is disturbed.

We have experimented with two different methods in order to take care of this problem. (i)The size of the message plays an important role in shortest path computation whereby the message is propagated only on that particular shortest path that is good enough to transfer the the given message volume. (ii)It is also possible that there are several paths available but none of them is good enough to deliver the given message, at the same time their commutative throughput exceeds easily the requirements of the message. In this scenario, we propagate the message fragment on the first shortest path that is just big enough to be delivered by this path, and allocate the remaining fragment for the next shortest path. This fragmentation process

```

Input: Message  $M$  of size  $M_S$  and destination  $D_M$ 
 $deliveredBytes = 0;$ 
 $edgeBuffer = \emptyset;$ 
while  $deliveredBytes \neq M_S$  do
  if shortest path  $P$  exist to  $D_M$  then
    /* Calculating Bytes deliverable
       by Path  $P$  */
     $B_p = MAX;$ 
    foreach Edge  $E \in in P$  do
       $B_p = \text{Min}(B_p, \text{capacity}(E));$ 
       $\text{Insert}(edgeBuffer, E);$ 
    end
    Propagate  $M$  on path  $P$  to  $D_M;$ 
     $deliveredBytes = deliveredBytes + B_p;$ 
  else
    /* restore the edges consumed in
       unsuccessful transmission */
     $\text{restoreAll}(edgeBuffer);$ 
     $\text{exit}();$ 
  end
end

```

Algorithm 2: EPO-X propagation for a given message M

continues until the whole message is delivered and if, in the end, there are still message fragment left undelivered with no more paths available, all the bandwidth consuming transactions made by earlier fragments are revoked. This algorithm practically utilizes such an oracle that can foresee the commutative throughput of all the paths to any destination and propagation is attempted only when the whole message is deliverable, otherwise it is ignored.

Based on the two approaches discussed above, we have implemented two different versions of the proposed benchmark, i.e. EPO and EPO-X, where both variations compute the shortest path using modified Dijkstra algorithm[?]. EPO, for a message of a given size, finds such a shortest path that is good enough to deliver the amount of bytes of the message. If such a path is not found, that message is discarded, otherwise the message is propagated to destination and the next message is not propagated till first message has reached its destination. Whereas, EPO-X, presented as Alg ??, not only finds the path to the destination but also determines how many bytes can the first shortest path deliver. The message size is reduced to the threshold that the first shortest path can deliver, and that fragmented message is delivered. EPO-X, then looks for the second shortest path to deliver the remaining fragment of the message and this process continues until the whole message is delivered. If it is not possible to deliver the whole message, all the transmission adjustments made for the earlier fragments of the current message are revoked. This step ensures that no unsuccessful transmission effects the delivery of the messages that are created later in time. Both EPOs, of course, have a greedy aspect such that they may sacrifice a vital link for a big message that is created later than a small message that happens to be using the same link. In our opinion, EPO is not only a better replacement for the upper bound, but will also deliver accurate information about the Network characteristics, such

Message count	100
Message size	1.6E3...1.6E7 B
Size distribution	Power law
Bandwidth (low)	100 kiB/s
Bandwidth (high)	10,000 kiB/s

TABLE I
SIMULATION PARAMETERS

as contact frequencies, contact durations, number of paths, edge stability, etc.

III. SIMULATION SETUP

We have considered three different kinds of data sets, all of which have been obtained from CRAWDAD. The motivation behind choosing these three traces have been to have a broad spectrum between dense and sparse networks. Two of the data sets have been synthesized from reality mining project [?] from MIT spans on 16 months i.e. February 2004 to August 2005 whereas, the third data consist of the SNMP logs for one month from a IBM campus[?]. As the duration span of MIT reality mining is longer than IBM trace, we have filtered the MIT data to match the time span of IBM traces.

The sparse network is obtained from bluetooth logs of MIT traces where each node scans every five minutes for active Bluetooth neighbors and stored the duration of contact times. For the sake of comparison with other traces and simplicity, we limit ourselves to one month of connectivity trace, where any visible Bluetooth device was considered a candidate connection. Reduction of the trace time span has been done on the basis of connectivity times i.e, one month where nodes have maximum connectivity in terms of time duration. The highest connectivity period i.e. November 2004 showed 1858 bluetooth nodes suggesting a huge number of undesigned nodes as compared to the designed² 81 nodes that were designated to gather the data. It is here noteworthy that a few undesigned devices had more connectivity and interaction with the network than the designed nodes.

In the case of IBM Access Point trace, SNMP is used to poll access points (AP) every 5 minutes, from July 20, 2002 through August 17, 2002. A total of 1366 devices have been polled over 172 different access points during approximately 4 weeks. We have extracted the traces of 928 device after discovering existence of 3 clusters in this network and then choosing the biggest cluster with respect to node count. To turn these samples into continuous data, we assume that the snapshot data remains constant for the next 5 minutes. In the rare cases where this would cause an overlap with another snapshot from another access point, we assume that the transition happens halfway between the two snapshots. We assume that two nodes that are connected to one access point during overlapping time period are connected to each other. Thus, key features of such a network are low mobility and medium transmission range.

The third trace, MIT Cell Tower, is used according to the similar principal as that of IBM traces. The only difference being, instead of access points, cell towers are used to gather

the contact times of the nodes with each other, thus the resulting network can be characterized as a very dense network due to high range of cell tower. Due to several lapses in data gathering, mentioned by the creators of the data, only 89 of 100 devices are included, which visit 32768 different cell towers. Similarly to Bluetooth traces, November 2004 turns out to be the maximum activity month with 81 devices and 12592 distinct cell towers.

It is imperative to mention that the assumption that two devices connected to one base-station(access Point or cell tower), introduces inaccuracies[?]. On one hand, it is overly optimistic, since two devices attached to the same access point may still be out of range of each other. On the other hand, the data might omit connection opportunities, since two nodes may pass each other at a place where there is no base-station, and this contact would not be logged. Another issue with these data sets is that the devices are not necessarily co-located with their owner at all times (i.e. they do not always characterize human mobility). Despite these inaccuracies, such traces are a valuable source of data, since they span many months and include thousands of devices.

a) Simulator: The motivation behind the simulator is to help us find the delays incurred by messages and overhead suffered by networks during execution of different routing algorithms. The output is analyzed on the basis of both number of messages as well as amount of data delivered. As already mentioned, three different traces have been used that significantly differ in the number of nodes involved, number, frequency, and distinctness of meetings that were taking place among the participants. For the purpose of this simulation, nodes connected to the same access point or the same cell tower are considered to be close enough physically to directly exchange messages with each other. IBM traces come out to be a sparsely connected network and MIT Cell Tower, a dense network, as the range of access points is smaller than that of cell towers. We have created 100 messages for the simulation with different sizes. The smallest size is 1600Bytes where as the the largest message size is 1.6E7 Bytes. We have followed power law to assign the sizes in this range; i.e. many small messages and a few huge messages. The peripheral simulation parameters are summarized in Table ??.

A. Simulated Strategies

We have simulated the following three routing strategies for the sake of comparison.

b) Flooding: A node attempts to forward all of the messages it has, to all of it's neighbors that are not in the possession of the message[?].

c) Directed Flooding: In this case, nodes propagate history about all of their contacts in a transitive manner among each. Flooding is done only on those hops that have shown the accessibility to the destination for any given message.

d) Perfect Oracle: This strategy has the advantage of Contacts Oracle[?] that can foresee the appearance of an opportunistic contact. Perfect oracle also utilizes the modified Dijkstra algorithm to find the shortest path, but it cannot ensure that the computed path will deliver the message. As soon as,

²Nodes running the scanning software are referred to as designated

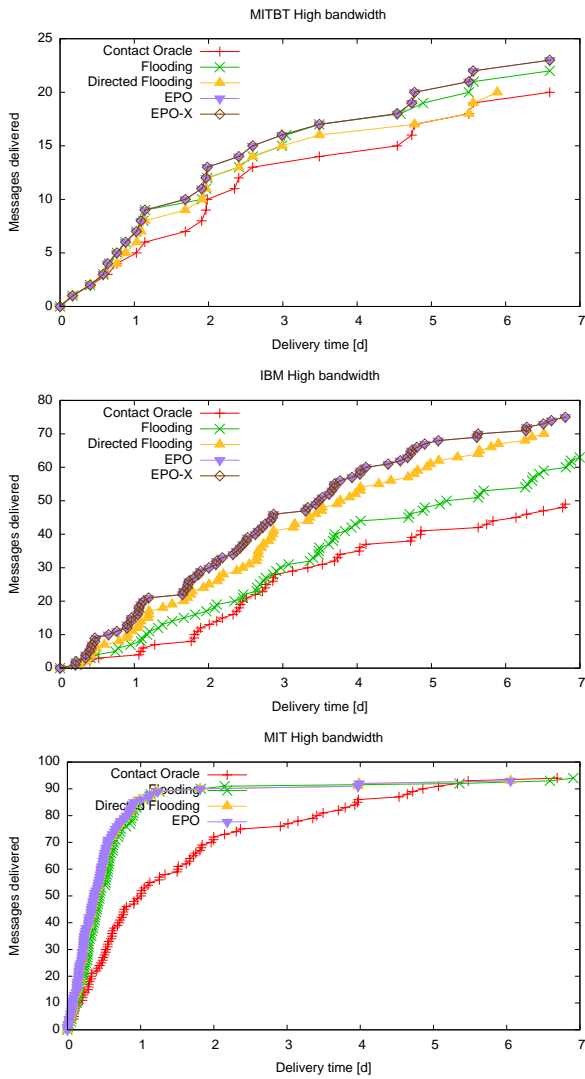


Fig. 3. Performance plots high bandwidth

a message is not able to be forwarded to next hop, path is recalculated.

IV. RESULTS DISCUSSION

When we analyze the two sets of results presented here in for high Fig ??, and for low Fig ?? bandwidth, the difference in delivery ratio in size as well as number of message prove the hypothesis that traffic volume does play a significant role in the case of opportunistic networks. Fig ?? shows that flooding is not always the winner even in the presence of abundant bandwidth, whereas a slightly intelligent scheme like Directed Flooding can outperform flooding in both low and high bandwidth scenarios. Moreover, in the case of high bandwidth Fig ??, plots for both EPO are similar, which suggests that almost all the first shortest paths had delivered the messages and the effects of transmission overlaps (two messages using the same link) have been minimal. Contact oracle came out to be last because there is no guarantee that a second shortest path exist by the time propagation of the message is halted on one of the hops due to lack of bandwidth.

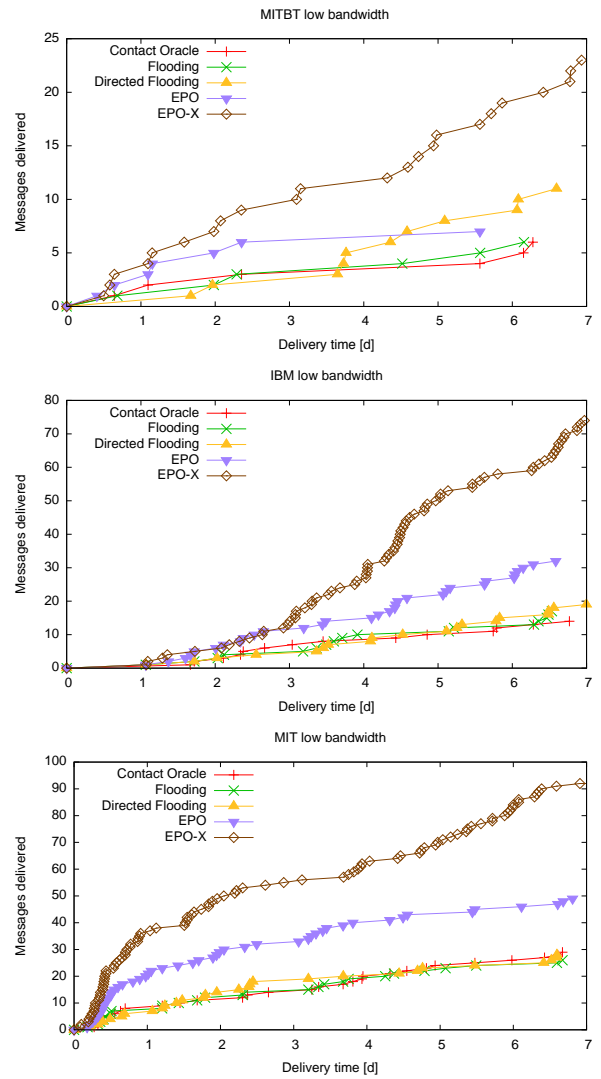


Fig. 4. Performance plots low bandwidth

The sparsity of the MITBT network is visible by the fact that Flooding delivered a mere 24 messages in MITBT and approx. 70 messages in IBM trace while MIT being the dense of all the traces, shows that flooding delivers 90% messages within a day.

In low bandwidth graphs Fig ?? results differences between EPO and EPO-X is more distinct with EPO-X being superior. Although, EPO has performed better than flooding since it has not been able to find paths for several huge messages. In contrast, EPO-X has exploited the cumulative available bandwidth through all the path to the destinations. It is interesting that if we compare performance of EPO-X in high and low bandwidth cases, the cumulative delivery ratio is somewhat similar. The only difference is the delay incurred by the messages as the buffer duration of messages has been long. The bigger the message is, the more round trips are needed to complete the full message transmission to the next hop. In the sparse case of MITBT, we notice the greedy nature of EPO as it has not performed better than Directed flooding. The explanation is that a few larger messages that are delivered

early on by EPO, have consumed the links that were pivotal for the delivery of some messages that are generated later.

V. CONCLUSION

In this paper, we have presented a different approach to simulate flooding that emphasizes its shortcoming by using messages of variable sizes. The above discussed results show that the behavior of opportunistic networks varies with the bandwidth variations. Moreover, flooding cannot be the winner whenever bandwidth is scarce and thus is not a suitable upper bound for such networks.

Furthermore, we have presented insight into two variations of a technique that not only be considered as a better upper bound (and can be improved further on) but also highlight the hidden aspects of the underlying network. Strategies like EPO exploit all the available resources and help identify the bottlenecks in opportunistic networks, paving the way for development of realistic and efficient routing protocols. Aspects, such as commutative throughput on the diameter of the network and identification of bottlenecks in clusters, will assist us to better understand such networks. In the future, our knowledge enhanced by these finding, will play an important role in developing suitable routing protocols.