Performance Evaluation of Routing Protocols in Opportunistic Networks

Performance Evaluation of Routing Protocols in Opportunistic Networks

Jinbang Chen
East China Normal University, China
jbchen@cs.ecnu.edu.cn

Wei Zhang
East China Normal University, China
wzhang@cs.ecnu.edu.cn

Paolo Giaccone
Politecnico di Torino, Italy
paolo.giaccone@polito.it

***Abstract*—We focus ourselves on a deterministic opportunistic-network in which the contact times between nodes are known in advance or can be predicted, and we consider two classes of traces collected in real life: public transportation system and human mobility. Our contribution mainly consists of two-folds: on one hand, we have exploited several real life mobility traces, trying to understand the pattern of the mobility and the interaction; on the other hand, we have evaluated the trace-driven performance of routing algorithm MC-DHCD, in terms of delay and throughput. We believe that our work, even though based on some simplified assumptions, will help the researchers to better design the routing protocol in an opportunistic network environment, which is able to provide feasible QoS requirements in terms of delay and throughput.**

***Keywords*—Routing protocol, performance evaluation, mobile networks, QoS.**

I. Introduction

As an emerging packet switching paradigm, opportunistic network[1] has attracted more and more attention in the research community. In such networks, end to end communication relies mainly on the cooperation among nodes involved in the networks as no fixed infrastructure exists. Since the network connectivity is intermittent and at any instant the network may be partitioned, routing in such networks is a big challenging issue.

Routing in a opportunistic network means that: at any given time, each node should find when and where to forward the data stored in its buffer so that it reaches the destination in a timely manner. It is therefore important to notice that the performance highly depends on the mobility pattern of the network nodes. Many works [2, 4, 5, 6, 7, 9, 11, 12, 13, 17] have addressed the routing issue of opportunistic networks. All these methods can be broadly divided into two classes: flooding protocols and forwarding protocols. Flooding protocols distribute many copies of the message to a large number of nodes with the hope that one of these intermediate nodes will reach the destination. As an alternative, forwarding protocols forward a single copy of each message along a carefully selected path.

This paper focuses on deterministic opportunistic networks in which the contact time between any two nodes are known (or can be predicted) in advance. The objective is to find how the performance (in delay-capacity region) depends on the nodes'mobility, and to see how the existing routing algorithm performs, which can reversely help the researchers in designing and improving the routing mechanisms in the future.

Our contribution mainly consists of two-folds: on one hand, we have exploited several real life mobility traces, trying to understand the pattern of the mobility and interaction; on the other hand, we have evaluated the trace-driven performance of routing algorithm MC-DHCD, mainly in terms of delay and throughput.

The remainder of this paper is organized as follows. In Section II, we have described the system model for the routing issue and the related framework for the performance evaluation. We then study the mobility pattern of four traces collected in real life, consisting of the contact density over time, the distribution of capacity the nodes can carry and forward, *etc.* in Section III, and the performance evaluation (in delay-throughput region) of the routing algorithm *MC-DHCD* applying on the traces collected are given in Section IV. Finally, Section V concludes the paper.

II. System Model

In this section, we first introduce the DTN graph, which is used to model a representative delay tolerant network relying on the movement of the nodes to transfer data. For the sake of simplicity, we consider the deterministic case in this article, meaning that all the contact times at which two nodes are going to meet is well known in advance. We then discuss two important performance metrics - delay and throughput, and provide the definition of these two metrics in a DTN environment. Finally, the routing algorithm MC-DHCD proposed in [16] is briefly presented. In later sections, we have conducted the performance evaluation by applying this algorithm to four mobility traces collected in real life.

A. The DTN graph

In order to discuss the DTN routing problem, we need a model that describes the network. In general, the DTN network can be modeled in the form of a graph, and any computing system participating in the network is called a *node* in the graph. Such nodes can be for instance vehicles running on a highway, or satellites orbiting around the earth as well as mobile phone-holder walking down the street. Each time a node meets another node, in this case, a communication and thus an exchange of data may occur between them. We define the opportunity to communicate as a *contact*, which is characterized by a duration of time, a capacity, and a propagation delay (assumed to remain constant during the

[1] Opportunistic network has the similar meaning to Delay Tolerant Network (DTN). In this paper, they are interchangeable during the whole discussion.

contact duration). Each contact is represented in the graph by a direct *edge* connecting two nodes together. The direction indicates in which way data flow between the two nodes. Due to mobility, failures, or other events, more than one edge can exist between a generic pair of nodes. Generally speaking, whenever an exchange of data occurs, an edge is inserted into the DTN graph connecting the corresponding nodes together. In addition, depending on the type of connection used, buffering constraints may also need to be considered. Therefore, in graph theory, this model is a time-varying multi-graph, in which more than one edge may exist between a pair of nodes (see Figure 1).

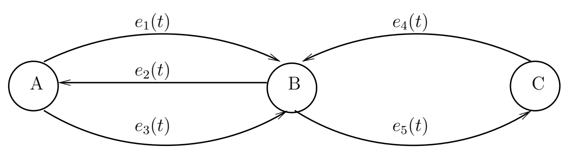


Fig. 1: An example of DTN graphs. Nodes may be connected by multiple edges, representing different physical links.

The reason for using a multi-graph is straightforward: it may be possible to select between two distinct (physical) connection types to move data between the same pair of nodes. Furthermore, the link capacities (and to a lesser extent, propagation delay) are time-dependent (capacity is zero at time when the link is unavailable). Thus, the set of edges in the graph must capture both time-varying capacity and propagation delay as well as multiple parallel edges.

We say that two nodes meet when they are able to communication with each other (roughly, they are closer than their transmission ranges). When node a meets node b, a contact event from a to b occurs. For simplicity, we assume that contacts are atomic and are associated with a *contact time* and a *contact capacity*, which is the maximum amount of data which can be transferred during that contact. With this assumption, accordingly, each edge is characterized by two fundamental parameters in the DTN graph: capacity and time of the contact. The first one represents the amount of data the two nodes have exchanged while the latter one represents the instant of the contact (when the data exchange occurs), which refers to a starting time so it is originally an absolute time.

As we have presented above, we model the set of all possible contacts through a directed multi-graph, denoted as *multi-contacts graph*. In order to better understand the algorithm to be introduced later, if e is an edge, here we define a time and a capacity functions $t(e)$ and $c(e)$, returning respectively the contact time and (measured in bytes or other units in a particular circumstance) the maximum amount of data that can be exchanged during the corresponding contact event. So, if for instance edge $e(u,\ v)$ is characterized by a capacity of 1,000 bytes and a contact time of 600 minutes (from starting time), then $c(e)$ will return 1,000 while $t(e)$ will return 600. In graph theory, this model is a multi-graph $G = (V,\ E)$ where V is the set of the nodes and E is the set of the edges. Starting from a generic trace containing contacts among the nodes of a DTN, it is always possible to build the corresponding multi-graph.

B. Performance metrics

When routing in a delay tolerant network, the time constraint is one of the main concerns as people always expect to transfer data as fast as possible, or at least in a reasonable and acceptable time scale. In addition, how much data can be flowed relying on routing through relay nodes are significantly important. We define in this section, two metrics - delay and throughput for the performance investigation of the routing protocol for DTNs.

1) Delay in a DTN: We define the *network delay* as the time interval between the instant the packet to be sent is injected into the network and the instant it arrives to the destination. Thus, if for instance a source node meets directly the destination node (*i.e.* a single-hop path), then the network delay equals to zero. Otherwise, we calculate it by making the difference between the instant when the second-last relay node transfers the message to the destination and the instant when the source node sends the packet to the first relay node. Note that this is one but not the only measure of how long it takes for a packet to reach the destination once injected into the network.

2) Throughput in a DTN: Another important metric to take into account when evaluating the performance of a routing protocol in DTNs is *throughput*. Before giving its definition, it is necessary to clarify what we call *maximum transferable amount of data* associated with a path. In general, a path is presented by a sequence of nodes and edges, which denotes the way a generic packet must follow in order to move from its source node to its destination node. Therefore, given a path, we are able to predict at which time instant the packet will arrive at a certain node and when it will leave the node. Since a path is possibly composed of several edges with different capacities, to send a generic amount of data over the path becomes difficult because there will be a capacity representing a bottleneck. As a result, the *maximum transferable amount of data* of a path is defined as the minimum capacity among all the edges belonging to the path. We define *throughput* achieved by a generic couple of nodes as the total amount of data sent by the source node and received by the destination node, obtained by summing up all maximum transferable data among all possible paths.

C. Algorithm Description

In a DTN graph, a routing path P of H hops is presented by a sequence of edges $(e_h)_{h=1}^{H}$ and by its capacity $r(P)$, which is the amount of data sent across that path. All the possible routing paths and their corresponding capacities make up the whole routing plan. We use the same method as depicted in [16] to calculate the delay $D(s,d)$ and the throughput $G(s,d)$. To be precise, the average delay $D(s,d)$ between a source s and a destination d is evaluated as the sum of the delay of each routing path, weighted by its capacity. The total throughput achieved $G(s,d)$ between a source s and a destination d can be calculated as the sum of the capacities of each routing path, divided by the whole duration of the trace.

We know from [15, 16] that, given a set of source-destination pairs, to find all the optimal routing paths we have to solve a multi-commodity flow problem. Given the source s and the destination d of a flow, finding the minimum delay path is straightforward with the algorithm proposed in [8]. As a simple modification of the classical Dijkstra algorithm, it always chooses the node with the lowest exploitable contact time as the best neighbor of a starting node.

In [16], a novel routing algorithm for deterministic DTNs called *MC-DHCD* is proposed. The idea is to visit the graph in two directions – forward (from s to d) and backward (from d to s). On one hand, *F-DHCD* (Forward optimal Delay Hops Capacity Dijkstra) finds a minimum delay path from s to d, while it tries, greedily, to minimize the number of hops and maximize the capacity. This path is not always DHC-optimal (Delay Hop Capacity-optimal, refer to [16] for more detail). One other hand, *B-DHCD* (Backward optimal DHCD) exploits the information regarding the delay-optimal path to prune the decision tree very efficiently and to build the DHC-optimal path.

In general, the *MC-DHCD* (MultiCommodity minDelay minHops maxCapacity Dijkstra) algorithm solves the global multi-commodity problem iterating the following two phases: (1) for each flow, it finds the DHC-optimal path between the corresponding sources and destinations, exploiting the sequence of *F-DHCD* and *B-DHCD*; (2) it then allocates the capacity using a max-min fair allocation. The algorithm ends when no capacity can be allocated anymore. In summary, *MC-DHCD* is a simple greedy approach to find an approximation of the optimal solution, based on a DHC-optimal search of the paths. More details for the algorithm *MC-DHCD* can be referred to [16].

III. Mobility Study

As for mobile ad-hoc wireless networks (*MANETs*), the end-to-end communication is achieved through the cooperation among the nodes. In such networks, a direct routing path is not always possible since the network connectivity is intermittent and the network may be partitioned at any time instant. Therefore, the routing issue becomes significantly important but challenging, and the performance to be obtained highly depends on the mobility of the peers, including contact opportunities, the duration for exchanging information, the mobility speed, *etc.*.

To support the QoS constraints, mainly in terms of delay and throughput, or any other metrics in a particular environment, the routing mechanism need to be carefully designed as no fixed infrastructure exists for the communication between any two nodes. To understand the mobility pattern of nodes and further to exploit it for routing, we introduce in this section two types of mobility traces collected in real life – public transportation system and human mobility. With these traces, we expect to extract useful characteristics about the mobility and the interaction among peers, reflecting the general manner how the peers move and helping to understand in which way the data could be reliably transferred among them.

A. The Dataset

The datasets considered for the mobility study in this paper consists of four traces, named as UMass1, UMass2, Milano and Imotes respectively, all of which are publicly available in CRAWDAD repository [1]. Basically, we classify them into two categories: vehicle-based transportation system (UMass1 and UMass2) and human mobility with mobile devices (Milano and Imotes), representing two kinds of typical mobile wireless networks in people's daily life.

The datasets of UMass1 and UMass2, are identically collected by the experiments of bus-to-bus transfer scheduled at UMass campus day by day, but in two different time periods. During the experiments, a WiFi node is attached to each bus for the data communication. As buses travel their routes, they encounter other buses and in some cases are able to establish pair-wise connections and transfer data between them. UMass1 consists of 30 buses operated in the experiments, providing 60-days data and 22102 contact events, whereas UMass2 is composed of 37 buses, with 81-days data and 34763 contacts collected in total. For more detail of these datasets and their related testbed, readers are recommended to refer to [3].

The dataset of Milano are collected at University of Milan (detail given in [14]) from 44 mobile devices, modeling the time evolution of contacts among people in a university campus area, where faculty members and students were involved. Milano trace is recorded with a duration of 11 days, in which 14589 contacts are collected in total. In some sense, it could be seen as the human mobility trace in the campus area that underlies opportunistic communications.

The dataset of Imotes are from the experiment conducted in Cambridge [10]. In this experiment, mobile users and stationary nodes with fixed location are both performed to track not only contacts between different mobile users, and also contacts between mobile users and various fixed locations. The experiment lasts about 55 days, and all the data are gathered finally from 36 mobile participants and 18 fixed locations. As here we focus only on the mobility behaviors, we keep all the contacts between the mobile users in the dataset, and exclude the others. With this operation, we finally obtain the trace containing 10873 contacts in total.

To provide a standard input to the simulator, we reshape each contact to a sequence with the format of "*time, source, destination, capacity of the contact*".

B. Statistic analysis

Throughout the whole discussion, we define a contact event as the occurrence of data exchange lasting some time between two nodes, such as buses or mobile devices involved in the opportunistic networks when they meet within their communication regions. In this process, we refer the sender as the source, and the receiver as the destination. To describe a contact event, the following information are in general required: source-destination pair, the occurrence time, and the amount of data exchanged (*i.e.* capacity, bandwidth or throughput).

When designing the routing mechanism for the DTNs, we need to take into account carefully the mobility behavior of the peers involved. Thus, the mobility study becomes an more and more important issue. To better study the common characteristics of the traces and to understand the underlying pattern of real life mobility, some statistical methods are introduced and deployed.

In our trace analysis, the following three metrics are proposed to represent the trace mobility: the contact events over the time, the cumulative number of contacts over the time, the distribution of contact bandwidth/capacity. All of them help us to study the trace mobility, and at the same time remind the researchers to consider the maximum capacity issue and the trade-off between delay and capacity over the delay tolerant networks when they design the routing protocol for DTNs .

The graph of contact events[2] over the time gives a picture of the occurrence of all the contact events in the temporal scale, which deploys two information of each contact event: source-destination pair and the occurrence time of corresponding contact, ignoring all the capacity information. With this plot, we are able to see clearly how frequent the contact event occurs during the whole period of the experiment, and further to compare the different feature (sparse or dense) among different time periods with the same duration. In addition, as a complement, the graph of the cumulative number of contacts over the time is given as well, showing the increasing tendency of the contact events as the time increases.

We use two graphs to represent the contact bandwidth distribution: the frequency graph of contact bandwidth and the inverse cumulative distribution of contact bandwidth. The former one tells us how the contact bandwidth distribute in each non-overlapping interval, while the later one helps us to characterize the contact bandwidth.

1) Contact events and Cumulative number of contacts: When studying the pattern of the mobility, we have plotted the graphs of contact events over time for all the four traces: UMass1, UMass2, Milano and Imotes. Similar characteristics are clearly observed for the four traces. To avoid redundancy, we report only the results for the case of UMass1 which is representative enough for the presentation, given in Figure 2(a). On one hand, the periodic (or nearly periodic) behavior is observed as the fact that the buses are served according to the same time schedule day by day in the experiment for UMass 1 and UMass2, while people holding the mobile devices tend to follow a habitual route in their daily life for Milano and Imotes. On the other hand, the contacts at night and in the weekend are quite rare in the experiment, proved by the sparse symbol "+" in the graph.

Therefore, for the performance evaluation in Section IV, we consider only the first three-week of contact information for UMass1 and UMass2, all the 12-day contact information for Milano and Imotes, without taking into account the contact events occurred at night and during weekend.

2) Distribution of contact bandwidth: Figure 2(b) reports the inverse cumulative distribution of contact bandwidth for UMass1. It shows that the majority of the contact bandwidths lie in the interval between 100 Kbytes and 10,000 Kbytes (*i.e.* 10 Mbytes). In addition, the distribution of the contact bandwidths approximately follows a power-law. Similar features are observed for other traces (UMass2, Milano and Imotes), where the only difference for Milano and Imotes lies on the fact that most of the contact capacities locate in the interval between 100 and 10,000 seconds[3] We here skip the placement of the graphs for UMass2, Milano and Imotes since the results for them are observed to be similar to UMass1.

IV. Performance Analysis for MC-DHCD

In general, many metrics could be used to evaluate the performance of a routing algorithm for delay tolerant networks. For the sake of simplicity, we refer the performance as the delay-capacity region throughout our whole discussion. The definition of delay and capacity in a DTNs environment have been presented in Section II-B.

In this section, we have applied the routing algorithm *MC-DHCD* to 4 mobility traces collected in real life, so as to study and to analysis the performance in the delay-capacity region. We aim at finding how does the performance of the routing protocol (in delay-capacity region) depends on the mobility.

In addition, two statistic measures are introduced for the analysis: *Coefficient of variation* and *Entropy*. They can help us to better understand the statistical characteristics of the results obtained. Recall that, the coefficient of variation (*CV*) is in genearl defined as the ratio of the standard deviation σ to the mean μ: $c_v = \frac{\sigma}{\mu}$, reflecting the dispersion over the mean value; while the entropy is a measure of the uncertainty associated with a random variable, lying between 0 and $\log_2(n)$ - for a random variable X with n outcomes $\{x_i,\ i = 1, \cdots, n\}$.

A. Simulation results

To evaluate the performance of the routing algorithm, we have applied the *MC-DHCD* algorithm to several real opportunistic scenarios, in which UMass1 and UMass2 represent the contact traces of the public transportation system in a university campus, while Milano and Imotes depict the contact traces of human's activity in a campus area.

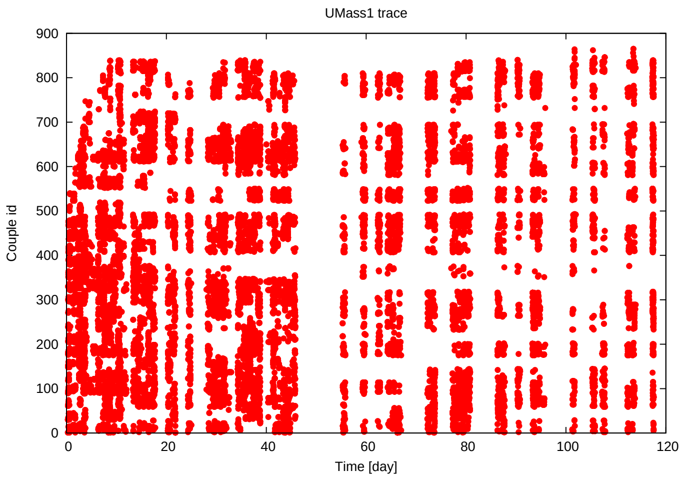
Before taken as the input to the simulator, all the four traces are pre-processed, resulting that each contact event is presented in a standard 4-tuple: "*time, source, destination, capacity of the contact*". When the simulation is performed, sets of routing paths are generated for all the possible $n(n-1)$ source-destination pairs, where n is the number of nodes, and the performance values in delay-capacity region are computed as well.

1) Scenario of Public Transportation System – UMass1 and UMass2: We consider the contact events in the first three weeks, where the contacts occurred at night and during weekend are excluded. We believe that it is representative and steady since the periodic behavior is clearly observed in mobility study in Section III.

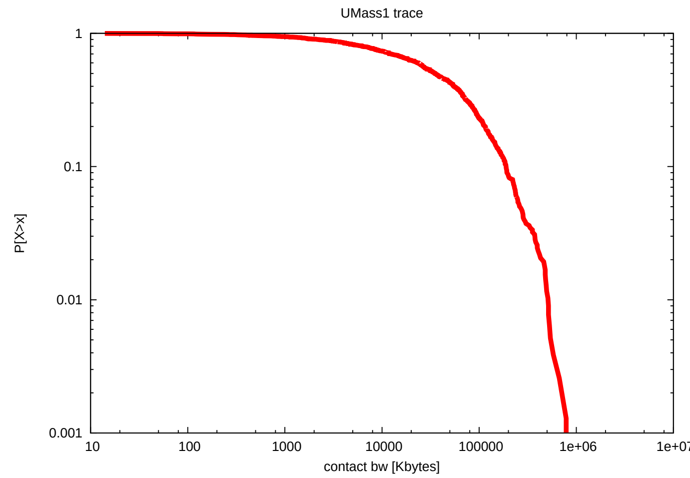
For UMass1, the throughput and the corresponding delay (mean value) for all traffic is given in Figure 3(a). The graph shows that the overall delay ranges from a minimum of 2 hours to a maximum of 93 hours – approximately four days. Surprisingly, only 24 couples of buses (2.99%) have a mean overall delay lower than 6 hours and only 74 (9.2%) lower than 10 hours. It means that only a limited set of buses could be used for spreading, for example, the newspaper of the campus (requiring a delay not higher than 6 hours) or electronic mails (delay not higher than 10 hours). This result is quite far away from our expectation – a great number of couples are characterized by low delays.

[2]To distinguish different contact events, we have assigned an unique id to each contact event.

[3]Note that for the trace of Milano and Imotes, the contact bandwidth is present by the duration each contact lasts, with the unit of second.

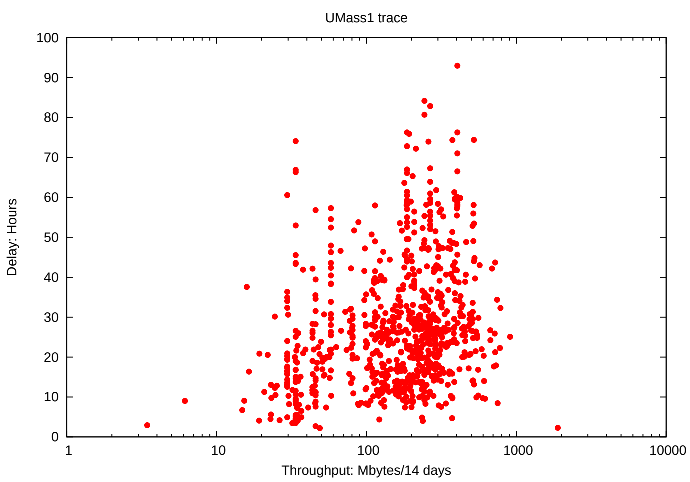


(a) Contact events over the time

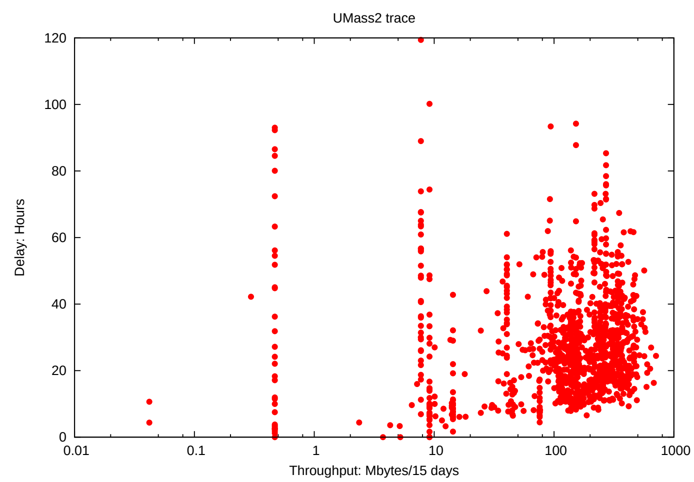


(b) Inverse cumulative distribution of contact bandwidth

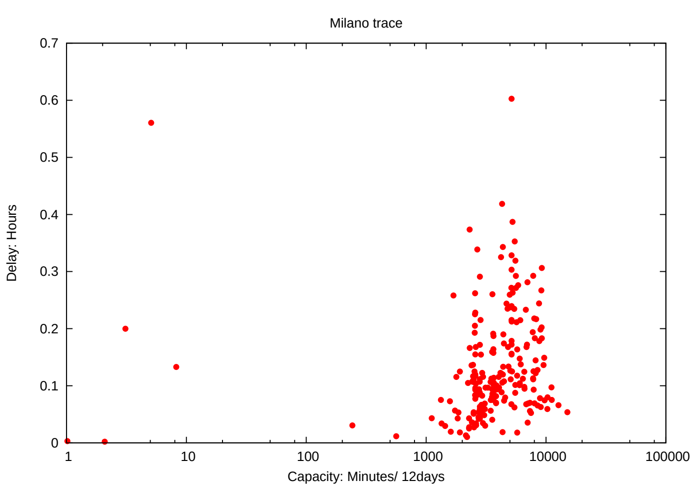
Fig. 2: Mobility Study – UMass1



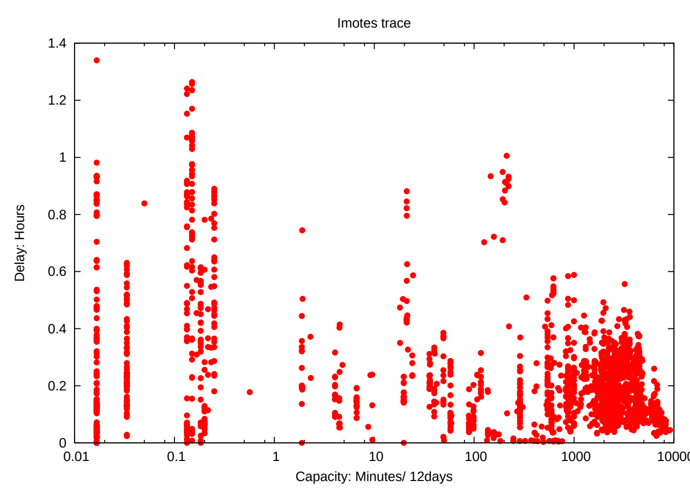
(a) UMass1



(b) UMass2



(c) Milano



(d) Imotes

Fig. 3: The performance in delay-capacity region

For UMass2, the simulation results are shown in Figure 3(b) where the throughput and the corresponding mean delay are reported for each couple of buses. The graph shows that only 28 couples of buses have a mean delay lower than 6 hours and only 100 lower than 10 hours, accounting for 2.5% and 8.9% respectively. It means that only a small set of traffic flows experience average delays acceptable in the application.

In addition, an interesting aspect is that there are groups of bus-pair (source-destination pair) characterized by identical throughputs but different mean delays. Intuitively, this identical throughput should be reasonably small. These groups of couples are marked out by the same source or destination nodes. All these nodes have very small outgoing (source) or incoming (destination) edge capacities. In fact, a capacity bottleneck is present either at the source or at the destination for all these groups. This implies that all maximum throughput paths are affected by such bottleneck, whereas the delays depend on the actual paths.

The statistic measures for UMass1 and UMass2 are given in Table I and Table II respectively.

TABLE I: Statistic Measures - UMass1

	average value	coefficient of variation	entropy	entropy/log2(N)
delay	29.70 [hours]	0.539	6.114	0.984
capacity	215.51 [Mbytes]	0.751	6.872	0.951

TABLE II: Statistic Measures - UMass2

	average value	coefficient of variation	entropy	entropy/log2(N)
delay	28.85 [hours]	0.478	6.156	0.950
capacity	191.36 [Mbytes]	0.689	8.237	0.966

2) Scenario of Human Mobility – Milano and Imotes: Since Milano and Imotes are both traces collected from real life presenting human mobility, we tend to put them together for the performance comparison. It makes sense due to the fact that they are more similar to each other than UMass trace (UMass1 and UMass2), and could be regarded as the same class of delay tolerant networks.

For Milano, the performance in delay-capacity region is given in Figure 3(c). The graph suggests a reasonable and acceptable result in the delay domain for real life application. Another interesting aspect is that, there are about 80.2% of couples of nodes whose overall mean delay is smaller than or equal to 12 minutes, and about 94.1% of couples of nodes whose overall mean delay is smaller than or equal to 18 minutes. It means that large amount of mobile nodes could be used for spreading. Therefore, if Milano trace can represent the real life human mobility properly, we may say that to obtain acceptable performance using proper rouing strategy in opportunistic networks is possible.

For Imotes, the result is given in Figure 3(d), which is encouraging as the overall mean delay ranges from 0 hours to 1.34 hours. Moreover, 1752 couples of nodes have a mean overall delay lower than 0.5 hours and 1921 lower than 0.5 hours, accounting for 89.8% and 98.5% respectively.

The statistic measures for UMass1 and UMass2 are given in Table III and Table IV respectively.

TABLE III: Statistic Measures - Milano

	average value	coefficient of variation	entropy	entropy/log2(N)
delay	0.14 [hours]	0.632	4.627	0.973
capacity	4493.21 [minutes]	0.557	6.725	0.967

TABLE IV: Statistic Measures - Imotes

	average value	coefficient of variation	entropy	entropy/log2(N)
delay	0.18 [hours]	0.559	4.929	0.806
capacity	1578.13 [minutes]	1.191	7.735	0.958

Different from UMass1 and UMass2, Milano and Imotes both show quite small overall mean delay, ranging within 1.4 hours, which is quite positive (Milano is 0.14 hour, while Imotes is 0.18 hours). Remember that Milano and Imotes are collected to represent the human mobility. The delay results suggest that they are more valuable for real life application. In addition, from the graph, we found that most of the overall mean delays range in low value scale (for Milano, 220 couples of nodes have a overall mean delay lower than 0.5 hours (99%); for Imotes, 1752 couples of nodes have a overall mean delay lower than 0.5 hours (89.8%)). On the other hand, we found that Milano has lower average delay while Imotes has lower coefficient of variation (concentrated distribution).

V. Conclusion and Future Work

We have studied the mobility pattern of four traces collected from real life and assessed the trace-driven performance of a routing algorithm for DTNs, in terms of minimum delay and maximum throughput. The four contact traces representing the public transportation system and human mobility, exhibit periodic feature and (approximate) power law of contact bandwidth distribution from our study. As a modified version of the classical Dijkstra algorithm, *MC-DHCD* aims at finding the optimal routing plan (set of routing paths) for delay tolerant networks. We applied it to two classes of contact traces - public transportation system (UMass1,UMass2) and human mobility (Milano, Imotes). By analyzing the results in delay-throughput region, the performances are not very encouraging since only few nodes exchange data with reasonable delay and throughput, in particular for the case of public transportation system. Still, we are the first ones to discuss the performance issue of DTN routing. We have highlighted that the trace mobility could significantly affect the performance of the routing mechanism. It is therefore necessary to take into account it when one is designing the routing protocol for DTNs. However, we have only scratched the surface of things. We believe that the new traces, which have become publicly available, will motivate our future investigations. We expect that our work could attract more researchers' attention, and encourage them to deliver more contributions in this issue.

Acknowledgment

This work was supported by the Italian Ministry of University and Research (MIUR) through the PRIN project CARTOON, and by the Chinese Fundamental Research Funds for the Central Universities under the Grant NO.78210250. The main part of this work was conducted when the first author was at Politecnico di Torino, Italy.

References

[1] Crawdad repository. Available at: http://web.scalable-networks.com.

[2] Aruna Balasubramanian, Brian Levine, and Arun Venkataramani. Dtn routing as a resource allocation problem. In *Proceedings of the 2007 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '07, pages 373–384, New York, NY, USA, 2007. ACM.

[3] John Burgess, Brian Gallagher, David Jensen, and Brian Neil Levine. MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networking. In *Proc. IEEE Infocom*, April 2006.

[4] Zhenxin Feng and Kwan-Wu Chin. A survey of delay tolerant networks routing protocols. *CoRR*, abs/1210.0965, 2012.

[5] Sabrina Gaito, Elena Pagani, and Gian P. Rossi. Fine-Grained Tracking of Human Mobility in Dense Scenarios. In *6th IEEE Annual Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks Workshops*, June 2009.

[6] Matthias Grossglauser and David N. C. Tse. Mobility increases the capacity of ad hoc wireless networks. *IEEE/ACM Trans. Netw.*, 10(4):477–486, August 2002.

[7] David Hay and Paolo Giaccone. Optimal Routing and Scheduling for Deterministic Delay Tolerant Networks. In *Sixth Annual Conference on Wireless on Demand Network Systems and Services*, February 2009.

[8] Sushant Jain, Kevin Fall, and Rabin Patra. Routing in a delay tolerant network. In *Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '04, pages 145–158, New York, NY, USA, 2004. ACM.

[9] Evan P. C. Jones, Lily Li, and Paul A. S. Ward. Practical routing in delay-tolerant networks. In *Proceedings of the 2005 ACM SIGCOMM Workshop on Delay-tolerant Networking*, WDTN '05, pages 237–243, New York, NY, USA, 2005. ACM.

[10] Jeremie Leguay, Anders Lindgren, James Scott, Timur Friedman, and Jon Crowcroft. Opportunistic content distribution in an urban setting. In *Proc. ACM SIGCOMM 2006 - Workshop on Challenged Networks (CHANTS)*, Pisa, Italy, September 2006.

[11] Cong Liu and Jie Wu. Scalable routing in delay tolerant networks. In *Proceedings of the 8th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '07, pages 51–60, New York, NY, USA, 2007. ACM.

[12] Mengjuan Liu, Yan Yang, and Zhiguang Qin. A survey of routing protocols and simulations in delay-tolerant networks. In Yu Cheng, Do Young Eun, Zhiguang Qin, Min Song, and Kai Xing, editors, *WASA*, volume 6843 of *Lecture Notes in Computer Science*, pages 243–253. Springer, 2011.

[13] Khalil Massri, Alessandro Vernata, and Andrea Vitaletti. Routing protocols for delay tolerant networks: A quantitative evaluation. In *Proceedings of the 7th ACM Workshop on Performance Monitoring and Measurement of Heterogeneous Wireless and Wired Networks*, PM2HW2N '12, pages 107–114, New York, NY, USA, 2012. ACM.

[14] Paolo Meroni, Sabrina Gaito, Elena Pagani, and Gian Paolo Rossi. CRAWDAD data set unimi/pmtr (v. 2008-12-01). Downloaded from http://crawdad.cs.dartmouth.edu/unimi/pmtr, December 2008.

[15] Alessandro Di Nicolo. Optimal routing algorithms for Delay Tolerant Networks. In *Technical Report*, pages 20–22, 2007.

[16] Alessandro Di Nicolo and Paolo Giaccone. Performance limits of real delay tolerant networks. In *Fifth Annual Conference on Wireless on Demand Network Systems and Services*, January 2008.

[17] Xiaolan Zhang, Jim Kurose, Brian Neil Levine, Don Towsley, and Honggang Zhang. Study of a bus-based disruption-tolerant network: Mobility modeling and impact on routing. In *Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking*, MobiCom '07, pages 195–206, New York, NY, USA, 2007. ACM.