# CFP: Integration of Fountain Codes and Optimal Probabilistic Forwarding in DTNs

Ying Dai
Software Institute
Nanjing University
dy06@software.nju.edu.cn

Panlong Yang, Guihai Chen
Dept. of Computer Sci. and Tech.
Nanjing University
gchen@nju.edu.cn

Jie Wu
Dept. of Computer and Information Sci.
Temple University
jiewu@temple.edu

*Abstract*— **There has been much research focusing on the routing problem in delay tolerant networks (DTNs). Much of the work has mainly focused on coding schemes for message distribution, while other work has been done on the probabilistic forwarding. Coding schemes achieves higher delivery rate via redundancy before forwarding, while probabilistic forwarding will efficiently limit the abuse of the store and forward scheme, maintaining relatively high performance. Providing a reliable and efficient forwarding scheme proves to be challenging as coding and forwarding schemes should be jointly considered. In our paper, we present an optimal probabilistic forwarding scheme using fountain code, which we named as CFP, where reliability and efficiency can be achieved at the same time. In CFP, We use fountain codes to encode messages and provide the forwarding rule to decide whether to forward messages to another node. The probabilistic forwarding problem is modeled as an optimal stopping problem and our forwarding rule also consider the influence of fountain codes. We perform trace-driven simulations and compare CFP with other protocols. Simulation results show that, considering the delivery rate, delay, and number of forwarders, CFP performs better than other implemented protocols(Epidemic, OPF) in our simulation.**

## I. Introduction

The ability to transport data from a source to a destination is a fundamental ability of all communication networks. Recently, the concern about delay tolerant networks (DTNs) routing has been growing due to its great challenges. In DTNs, an instantaneous and constant end-to-end path may not exist due to its lack of permanent connectivity. Moreover, the unreliable and unexpected node behaviors in DTNs make the routing problem even harder. An original routing protocol in DTNs is epidemic, in which any node bearing messages forwards them to any possible node within its radio range, until the destination receives the message. This would maximize the delivery rate, but cause an obvious problem - the poor scalability, where forwarded messages will possibly floods over the network. Moreover, the feedback mechanism is difficult to design, as the acknowledgement also need a relatively long time routing process.

Much efforts has been made to improve the efficient use of network resources. Regrading this topic, there are two research directions. On the one hand, a lot of work is done on coding before messages are transmitted. One message is segmented into smaller frames, which could be forwarded independently over different paths. Moreover, a source node could distribute not only the original frames, but also redundant coded frames. If the destination receives enough number of frames, the original message could be recover from them. On the other hand, during the transmitting process, many researches focused on finding an effective forwarding rule. When the node with a message copy meets another node, the rule decides whether or not to forward messages. There also exists an optimal stopping policy for message forwarding [7]. In this way, the total number of message copies is reduced and the network resources are saved.

Unfortunately, neither these two types of schemes could effectively solve the DTN routing problem alone. In the coding based scheme, the reliability is achieved via redundancy, while in the optimal forwarding scheme, the messages delivery is constrained within limited hops, where routing efficiency would be lost. How to achieve an efficient and effective coding based optimal forwarding scheme is still challenging, as it is difficult to balance between these two types of schemes.

In this paper, we jointly consider the two fields of research work mentioned above and present our protocol CFP, which is the "integration of fountain codes and optimal probabilistic forwarding in DTNs". We use fountain codes, fragmenting the original message for efficient distribution. In addition, we present a forwarding rule to increase the delivery ratio while making better use of network resources. The forwarding of each frame could be modeled as an optimal stopping problem. In addition, we also consider redundant factors in our forwarding rule which is related to the fountain code we used. We take the influence of fountain codes on probabilistic forwarding into consideration. We propose the modified stopping rule. Also, we give the results based on different levels of fountain codes and compare the consumption of network resources between CFP and other protocols.

The main contributions of our paper are as follows:

- To the best of our knowledge, ours is the first one to propose an optimal probabilistic forwarding protocol with fountain codes in DTNs, which could effectively utilize the reliability in the coding scheme and the efficiency in the optimal forwarding scheme.
- We consider the influence on the forwarding rule caused by fountain code. We perform better simulations with the trace data from UMassDieselNet [1] and show the results of comparing CFP with other protocols.

Our paper is organized as follows: In Section II, some related work is presented. In section III, we introduce several preliminaries. Also, in Section IV, we give a detailed description of our model. Moreover, in Section V, we show implemented simulation and the validated results. We make a further discussion on our proposed algorithm CFP in Section VI. Finally, in Section VII, we describe our possible future work.

## II. RELATED WORKS

There has been much work done on both fountain codes and probabilistic forwarding, which aims to reduce the cost of forwarding while retaining a high performance rate.

Referring to fountain codes, [2] and [3] first erasure code a message and distribute the generated code-blocks in DTNs. [4] proposes general network coding techniques. [5] provides general phases to use fountain codes, including initialization phase, inference phase, encoding phase, storage phase, and decoding phase. Although [5] is based on sensor networks, it gives a full description of how to use fountain codes. In [6], the authors give the closed form description of the performance of delay tolerant ad hoc networks. [6] accounts for both the overhead of the forwarding mechanism, captured in the form of a given bound on energy, and the probability of successful delivery of the entire message to the destination within a certain deadline. Also, fountain codes are accounted for in closed form.

On the other hand, [7] proposed the optimal probabilistic forwarding protocol (OPF), which integrates an optimal delivery probability metric and an optimal forwarding rule. Firstly, the authors assume that the optimal delivery probability of a copy in a node $i$, heading for destination $d$, with a remaining hop-count $K$ ($K > 0$), and with a residual time-to-live $T$, is denoted by $P_{i,d,k,T_r}$. Secondly, the optimal forwarding rule in [7] is when a copy, whose remaining hop-count is $K$, is in node $i$ and node $i$ meets node $j$ at time-slot $T_r$, the decision on whether to forward depends on whether replacing the copy in $i$ with two new copies in $i$ and $j$, respectively, will increases the overall delivery probability. If the message is forwarded in time-slot $T_r$, then in the next time-slot, there are two new copies with remaining hop-count $K-1$ in $i$ and $j$, respectively, whose delivery probabilities are $P_{i,d,k-1,T_{r-1}}$ and $P_{j,d,k-1,T_{r-1}}$, respectively. To maximize the delivery probability, the optimal forwarding rule forwards the message if

$$1 - (1 - P_{i,d,k-1,T_{r-1}})(1 - P_{j,d,k-1,T_{r-1}}) \geq P_{i,d,k,T_{r-1}}. \quad (1)$$

In the case that a node meets several other nodes in the same time-slot, forwarding the copy to the node with the highest delivery probability is the optimal strategy to maximize delivery probability.

## III. PRELIMINARY

Before we start, there are two preliminaries: fountain codes [6] and optimal stopping rule problem [7]. Using fountain codes, the original message can be divided to several fragments with certain level of redundancy. In optimal stopping rule, the action is forwarding.

In coding theory, fountain codes are a class of erasure codes with the property that a potentially limitless sequence of encoding symbols can be generated from a given set of source symbols such that the original source symbols can ideally be recovered from any subset of the encoding symbols of size equal to or only slightly larger than the number of source symbols.

The theory of optimal stopping is concerned with the problem of choosing a time to take a particular action, in order to maximize an expected reward or minimize an expected cost.

There have been many work done on the two subjects. Here, we introduce both briefly.

### A. Fountain codes

Fountain codes are a class of erasure codes. For $k$ source blocks $x_1, x_2, \ldots, x_k$ and a probability distribution $\Omega(d)$ with $1 \leq d \leq k$, a fountain code with parameters $(k, \Omega)$ is a potentially limitless stream of output blocks $y_1, y_2, ..., y_n (n > k)$. Each output block is obtained by XORing $d$ randomly and independently chosen source blocks, where $d$ is drawn from a specially designed distribution $\Omega(d)$. The original source symbols can ideally be recovered from any subset of the encoding symbols of size equal to or only slightly larger than the number of source symbols.

### B. Optimal Stopping Rule Problem

The stopping rule problem is defined by two objects:

1) A sequence of random variables, $x_1, x_2, \ldots, x_m$ whose joint distribution is assumed known.
2) A sequence of real-valued reward functions, $y_0$, $y_1(x_1)$ $y_2(x_1, x_2), \ldots, y_m(x_1, x_2, ..., x_m)$, each of which is related.

Given these objects, the problem can be described as:

1) You are observing the sequence of random variables, and at each step n, you can choose to either stop observing or to continue.
2) If you stop observing, you will receive the reward $y_n$, which is the function of $x_1, x_2, ..., x_n$.
3) You want to choose a stopping rule to maximize your expected reward.

A stopping rule problem has a finite horizon if there is a known upper bound $T$ on the number of stages at which one may stop. If stopping is required after observing $x_1$,...,$x_T$, we say the problem has horizon $T$. In principle, such problems may be solved by the method of backward induction. Since we must stop at stage $T$, we first find the optimal rule at stage $T-1$. Then, knowing the optimal reward at stage $T-1$, we find the optimal rule at stage $T-2$, and so on, back to the initial stage (stage 0).

## IV. SYSTEM MODEL

Our model combines fountain codes and optimal probabilistic forwarding. It consists of three important parts:

1) Before transmitting, the original message must be split into multiple frames. Through the operations of fountain

codes, the source frames would be coded into new multiple frames, which would then be sent.
2) During every independent process of forwarding, each node has to decide whether to forward the frame to the next node or directly send to the destination.
3) For the destination, it needs to receive a certain number of frames so that it is able to decode the original message.

### A. Using LT Codes to Split Messages

We use Luby transform (LT) code [8] to split messages. LT codes are the first practical realization of fountain codes, which uses Ideal Soliton or Robust Soliton distributions [8]. LT codes are based on sparse bipartite graphs to trade reception overhead for encoding and decoding speed, the same as other fountain codes. The distinguishing characteristic of LT codes is that its algorithm to encode and decode the message is relatively simple.

The Ideal Soliton distribution $\Omega_i s(d)$ for $k$ source blocks is given by

$$\Omega_i s(i) = \begin{cases} \frac{1}{k}, i = 1 \\ \frac{1}{i(i-1)}, i = 2, 3, ..., k \end{cases} \quad (2)$$

By XORing d source blocks chosen from $k$ source inputs, where d is drawn according to the probability distribution above, we can now generate a new random combination of the original frames $x_1$, $x_2$, . . . , $x_k$.

### B. Optimal Forwarding Strategy

After the encoding part is completed, each frame needs to be transmitted independently. During the process, the node needs to transmit the frame to the next node and must decide when to stop, that is to say, when transmitting the data to the destination rather than forwarding to another node. Fortunately, as introduced in the above "Related Works" part, [7] has proposed a strategy of optimal forwarding protocol. However, it did not include the consideration of fountain codes. So, we make some changes to the optimal forwarding protocol in [7].

In [7], the optimal forwarding rule is to forward the message if

$$1 - (1 - P_{i,d,k-1,T_{r-1}})(1 - P_{j,d,k-1,T_{r-1}}) \geq P_{i,d,k,T_{r-1}}. \quad (3)$$

$P_{i,d,k,T_r}$ denotes the optimal delivery probability of a copy in node $i$, heading for destination $d$, with a remaining hop-count is $K(K > 0)$, and with a residual time-to-live $T_r$. And it has modeled each forwarding as an optimal stopping problem. In our model, this is still an optimal stopping problem, but we has added fountain codes before messages are started to transmit. Since the destination need only a certain number of frames to recover the original message, instead of all the frames, the optimal forwarding rule in [7] would be over-optimal here. Suppose the 1 - $\delta$ is the success probability ($\delta > 0$) that the destination to decode with M number of frames, we add $\delta$ to the optimal forwarding rule above. If

$$(1 - (1 - P_{i,d,k-1,T_{r-1}})(1 - P_{j,d,k-1,T_{r-1}})) * \delta \geq P_{i,d,k,T_{r-1}}, \quad (4)$$

the message would be forwarded. This rule reduces the chance for a node to forward a message and save the use of network resources.

Now we come to the computation of $P_{i,d,k,T_r}$. First, the meeting probability of two nodes in any time slot of length U is estimated under the assumption of exponential inter-meeting time [9], [10] by

$$M_{i,j} = 1 - exp(-\frac{U}{I_{i,j}}). \quad (5)$$

$P_{i,d,k,T_r}$ equals the sum of (1) the probability that the copy will be forwarded in time-slot $T_r$ and then be delivered, and (2) the delivery probability $P_{i,d,k,T_r-1}$ when the message is not forwarded in time-slot $T_r$ multiple by the probability that node i does not meet any node in $T_r$ that satisfies the forwarding criteria. [7] has given the algorithm to calculate $P_{i,d,k,T_r}$, which can also be applied here.

### C. Computation of Delivery Ratio

After the destination receives the frames, it needs a certain number of these frames to decode the original data. Using fountain codes, we know that for any $\delta$, in order for the destination to be able to decode the original message with the probability of at least 1 - $\delta$, it has t receive at least

$$M = klog(k/\delta) \quad (6)$$

packets [6]. Here, $k$ still denotes the number of the original frames.

On the other hand, the probability of a successful delivery of a single frame can be computed as follows: let the time between contacts of pairs of nodes be exponentially distributed with given inter-meeting intensity $\lambda$. The validity of this model has been discussed in [11], and its accuracy has been shown for a number of mobility models. Also, the transmitted message is relevant during some time $\tau$. Let $X_i(t)$ be the number of the mobile nodes (excluding the destination) that have, at time t, a copy of frame i. Denote by $D_i(\tau)$ the probability of a successful delivery of frame i by time $\tau$. Then, given the process $X_i$ (for which a fluid approximation will be used), we have

$$D_i(\tau) = 1 - exp(-\lambda \int_0^\tau X_i(s)ds) \quad (7)$$

Fortunately, [6] has given the computation of the total delivery rate, denoted by $P_M(\tau)$. In addition, the number of packets that reach the destination during the time interval [0,$\tau$] has a Poisson distribution with parameter $-L(\tau, p^*)p^*$, where $p^*$ is also given in (10) of [6].

$$P_M(\tau) = \sum_{i=0}^{M-1} \frac{(-L(\tau, p^*)p^*)^i}{i!} exp(L(\tau, p^*)p^*) \quad (8)$$

## V. Simulation

We evaluate our protocol using the UMassDieselNet trace. Also, we implement other protocols - OPF [7] and Epidemic [15] - to compare with our protocol.

TABLE I
SETTINGS FOR UMASSDIESELNET TRACE

| parameter name | default | range |
|---|---|---|
| initial hop-count (K) | | 1~5 |
| message time-to-live ($T_r$) | 10 hours | |
| simulation time | 1 day | day 1~55 |
| $\delta$ | 0.6 | |

### A. Simulation Methods and Settings

In the UMassDieselNet [12], [13] bus system consisting of 40 buses, the bus-to-bus contacts (the durations of which are relatively short) are logged. Our experiments are performed on traces collected over 55 days during the spring 2006 semester, with weekends, spring break, and holidays removed due to reduced schedules. The bus system serves approximately ten routes. There are multiple shifts serving each of these routes. Shifts are further divided into morning (AM), midday (MID), afternoon (PM), and evening (EVE) sub-shifts. Drivers choose buses at random to run the AM sub-shifts. At the end of the AM sub- shift, the bus is often handed over to another driver to operate the next sub-shift on the same route or on another route. Unfortunately, the all-bus-pairs contacts provided in the original traces show no discernible contact pattern among the nodes. We performed the data process in [14] to generate the contacts at a sub-shift level which exhibit periodic behavior. This process translates 55 days of the bus-to-bus contacts into contacts between sub-shifts.

The default settings of the UMassDieselNet trace simulation are shown in Table I. We use 1~5 initial hop-counts in OPF. We use the 55 days of traces to run respective simulations. In each simulation, every node (sub-shift) sends a message for a random destination node every five minutes. Since most contacts in the UMassDieselNet trace are between hours 6 and 20, messages are only sent during hours 6 to 12, and we set a uniform initial time-to-live of all messages to 10 hours. We use one minute as the time unit for residual time-to-live $T_r$. The mean inter-meeting time between all pairs of shifts is generated from the 55 days of sub-shift based contacts. In addition, the $\delta$ we chose is 0.6.

### B. Simulation Results

The comparisons of the delivery rate, delay, and number of forwarding are shown in Figures 1, 2, and 3, respectively. From Figure 1, the delivery rate of Epidemic is better than OPF and CFP. And the CFP has a better delivery rate than OPF . We can tell from Figure 2 that the delay of CFP is longer than OPF and Epidemic.

To better compare these three protocols, we compare the values of delivery rate/delay/number of forwardings. The results are shown in Figure 4. CFP is higher than OPF, while Epidemic is the smallest of these three.

We also compare the results among different $\delta$, as shown in Figure 5. Also, in Figure 6, we compute the average value of every $\delta$ from Figure 5. The vertical axis is the average value of points in a single line of Figure 5 while the horizontal axis is the different values of $\delta$, which denote the different lines in Figure 5. We can tell from Figure 6 that there exists an optimal value of $\delta$.

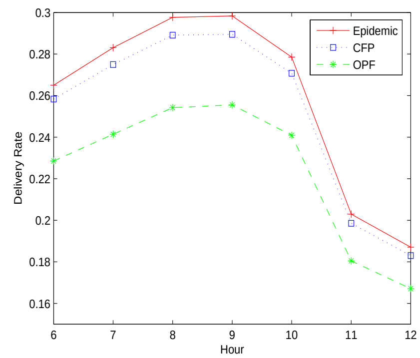


Fig. 1. The comparison of delivery rate among different protocols.

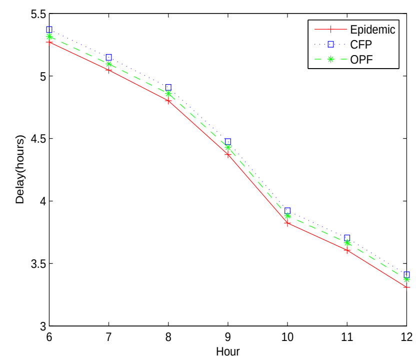


Fig. 2. The comparison of delay among different protocols.

## VI. DISCUSSION

In our model, we use the optimal stopping rule when forwarding messages. But, there should be more places in our model that we could apply the optimal stopping rule. We can also model the level of fountain codes as an optimal stopping problem. That is, the choose of $\delta$ should be an optimal stopping problem. The smaller $\delta$, the more packages the destination need to decode. The larger $\delta$, the more copies there would be, which places an influence on the performance of the whole network. So, when to stop the choose of $\delta$ would be an interesting optimal stopping problem.

Moreover, the influence caused by fountain codes could be made of use. A distributed strategy could be applied here to decide which nodes have the higher probability to forward messages, and which nodes have a lesser probability, due to the situation of each node. In this way, the network resources could be made more use of and the scalability of our protocol would be improved.

## VII. CONCLUSIONS AND FUTURE WORK

We present a new protocol which combines fountain codes with optimal probabilistic forwarding in DTNs. We also perform a trace-driven simulation and prove that our protocol is

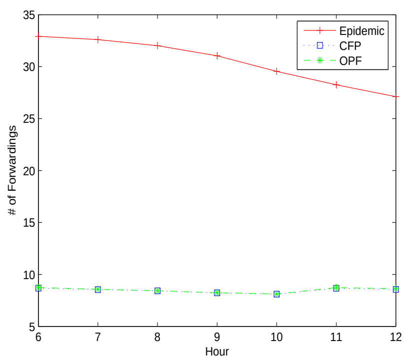


Fig. 3. The comparison of the amount of forwarding.

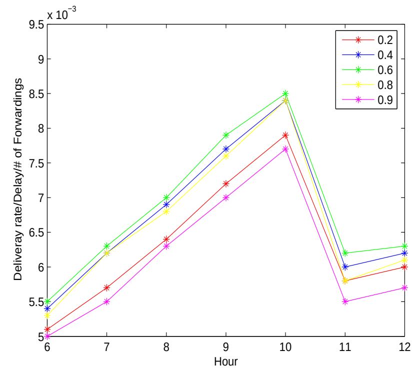


Fig. 5. The comparison of results among different $\delta$

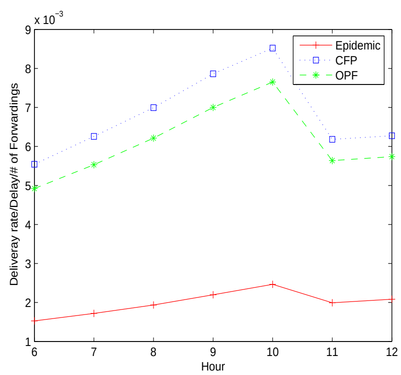


Fig. 4. The comparison of delivery rate/delay/number of forwardings

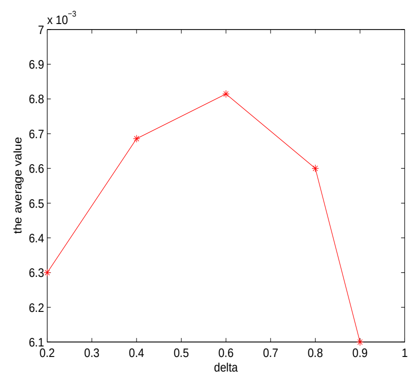


Fig. 6. The average value of different $\delta$

better than the Epidemic and OPF protocols. Some settings we used in our simulation would have influence on our simulation results. In the future, we plan to do a certain number of experiments and implement our protocol in the real environment to achieve a more accurate and reliable result. Also, the comparison between OPF and CFP on the consumption of network sources will be extensively conducted. We plan to compare the scalability of these two protocols in the future.

## References


[1] http://prisms.cs.umass.edu/dome/umassdieselnet

[2] Y. Wang, S. Jain, M. Martonosi, and K. Fall. Erasure Coding Based Routing for Opportunistic Networks. *SIGCOMM Workshop on Delay Tolerant Networking (WDTN),* 2005.

[3] S. Jain, M. Demmer, R. Patra, and K. Fall. Using Redundancy to Cope with Failures In a Delay Tolerant Network. *SIGCOMM Comput. Commun. Rev.,* 2005.

[4] C. Fragouli, J. Y. L. Boudec, and J. Widmer. Network Coding: An Instant Primer. *SIGCOMM Comput. Commun. Rev.,* 2006.

[5] S. A. Aly, Z. Kong, and E. Soljanin. Fountain Codes Based Distributed Storage Algorithms for Large-scale Wireless Sensor networks. *IPSN,* 2008.

[6] E. Altman and F. D. Pellegrini. Forward Correction and Fountain Codes in Delay Tolerant Networks. *Proc. of INFOCOM,* 2009.

[7] C. Liu and J. Wu. An Optimal Probabilistic Forwarding Protocol in Delay Tolerant Networks. *Proc. of ACM MobiHoc,* 2009.

[8] M. Luby. LT Codes. *Proc. 43rd Symposium on Foundations of Computer Science,* 2002.

[9] T. Spyropoulos, K. Psounis, and C. Raghavendra. Spray and Wait: An Effcient Routing Scheme for Intermittently Connected Mobile Networks. *Proc. of ACM WDTN,* 2005.

[10] A. Balasubramanian, B. N. Levine, and A. Venkataramani. DTN Routing as a Resource Allocation Problem. *Proc. of ACM SIGCOMM,* 2007.

[11] R. Groenevelt and P. Nain. Message delay in MANETs. *Proc. of SIGMETRICS,* 2005.

[12] X. Zhang, J. F. Kurose, B. Levine, D. Towsley, and H. Zhang. Study of a Bus-Based Disruption Tolerant Network: Mobility Modeling and Impact on Routing. *Proc. of ACM MobiCom,* 2007.

[13] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine. MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networking. *Proc. of INFOCOM,* 2006.

[14] C. Liu and J. Wu. Routing in a Cyclic MobiSpace. *Proc. of ACM MobiHoc,* 2008.

[15] A. Vahdate and D. Becker. Epidemic Routing for Partially Connected Ad Hoc Networks. *Technical Report, Duke University,* 2002.

[16] T. Camp, J. Boleng, and V. Davies. A Survey of Mobility Models for Ad Hoc Network Research, *Wireless Communications & Mobile Computing (WCMC),* 2002.

[17] D. Elizabeth and H. Mads. Social Network Analysis for Routing in Disconnected Delay-Tolerant MANETs. *Proc. of ACM MobiHoc,* 2007.

[18] P. Hui, J. Crowcroft, and E. Yoneki. BUBBLE Rap: Social-based Forwarding in Delay Tolerant Networks. *Proc. of ACM MobiHoc,* 2008.

[19] J. Leguay, T. Friedman, and V. Conan. DTN Routing in a Mobility Pattern Space. *Proc. of ACM WDTN,* 2005.

[20] J. Y. L. Boudec and M. Vojnovic. Perfect Simulation and Stationarity of a Class of Mobility Models. *Proc. of INFOCOM,* 2005.

[21] T. Karagiannis, J. Y. L. Boudec, and M. Vojnovic. Power Law and Exponential Decay of Inter Contact Times between Mobile Devices. *Proc. of ACM MobiCom,* 2007.