Computer Communications 73 (2016) 47–55

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Efficient and Transparent Use of personal device storage in opportunistic data forwarding

Sayed Amir Hoseini [a,b,*], Azade Fotouhi [a,b], Mahbub Hassan [a,b], Chun Tung Chou [a], Mostafa H. Ammar [c,1]

[a] School of Computer Science and Engineering, University of New South Wales (UNSW), Sydney, Australia
[b] National ICT Australia
[c] Georgia Institute of Technology, Atlanta, GA, USA

ARTICLE INFO

Article history:
Available online 23 October 2015

Keywords:
Delay tolerant networks
Opportunistic networking
Bayesian decision
Mobile device storage

ABSTRACT

We consider a growing research trend of using personal mobile devices for forwarding opportunistic network data. Because personal device storage is meant to support user applications, opportunistic networks must use it in a manner that remains completely transparent to the user. One way to make a device's storage use transparent is to allow priority access to the storage to user applications, even if the storage is currently occupied by network data yet to be forwarded. This means that data given to a device waiting to be forwarded can be overwritten by application data and may, thus, be lost. In this paper we consider random access memory (RAM) as the primary storage location in a mobile device. We propose three algorithms of different sophistications to answer the question of how much data should be moved when a contact opportunity arises between two devices in such a way to first maximise the data transferred while minimising the probability that this data will be overwritten when applications claim priority access. We collect 33 h of high-resolution RAM usage traces of two real smartphones over a 3-day period under a variety of usage scenarios to evaluate and compare the performances of the proposed algorithms. Surprisingly, we find that autoregression forecasting of RAM usage cannot outperform the simplest algorithm that greedily occupies all of the RAM that is found unused at the time of contact. We show that Bayesian inference is very effective in minimising the risk of data loss in such uncertain environments and significantly outperforms the greedy approach as well as autoregression forecasting.

1. Introduction

Networks using opportunistic communication (also known as Delay or Disruption Tolerant Networks) are typically made up of mobile devices that are intermittently connected to each other [1]. Data in such networks is moved along from one node to another as nodes make contact. Receiving nodes that are not the data destinations will *store, carry, and forward* [2] the data until a contact is made with another node. A variety of routing protocols [3,4] have been proposed to determine the best sequence of contacts to use in such networks.

An important common feature of all these protocols is the requirement for intermediate nodes to store the data, potentially while the node is moving until it can be forwarded to another node. Surprisingly there is little in the research literature to discuss where this data needs to be stored and what the implications of using such storage are on the performance of opportunistic networks. We first note that there are primarily two types of intermediate nodes when opportunistic communication is used. Some nodes can be considered as infrastructure nodes that have been tasked with moving the data along. Examples include message ferries [5], data mules [6], and throwboxes [7]. In this case such nodes can be provisioned with memory that can be used exclusively to store data as it is carried and before it is forwarded.

In other systems, [8] user devices (such as smartphones and tablets) act as intermediate nodes. In such cases, the device's own memory can be used to store data until it can be forwarded. Random access memory (RAM), internal memory, and external memory (e.g., SD card) are possible memory options. Opportunistic networks require high throughput during contact opportunities in order to enable the transfer of significant amounts of data during even short contacts. Of the three types of memory available on a device, DRAM-based RAM technology provides the highest throughput [9–11]. Also

* Corresponding author at: School of Computer Science and Engineering, University of New South Wales (UNSW), Sydney, Australia. Tel.: +61(2) 93856198.
E-mail addresses: amirhoseini@cse.unsw.edu.au (S.A. Hoseini), azadef@cse.unsw.edu.au (A. Fotouhi), mahbub@cse.unsw.edu.au (M. Hassan), ctchou@cse.unsw.edu.au (C.T. Chou), ammar@cc.gatech.edu (M.H. Ammar).
[1] Mostafa Ammar's work is partially supported by National Science Foundation through grants NETS 1409589 and NETS 1161879.

http://dx.doi.org/10.1016/j.comcom.2015.10.003
0140-3664/

the cost of RAM has fallen drastically in recent years fuelling large RAM capacity in smartphones. While early smartphones could barely store more than a few kilo bytes in RAM, today's smartphones are shipped with at least 1 GB of RAM. Mobile social networking, content sharing, social discovery, pervasive and urban sensing and opportunistic computing are the most relevant applications which employ smart phone memory [12]. We will show in this paper that even when such applications are deployed, a mobile phone's RAM is typically underutilised, thus allowing this available capacity to be used as storage in support of opportunistic communication.

Unfortunately, a major issue arises from the fact that the smartphones are *personal devices*. As such, the phone storage is primarily dedicated to serve user applications. If operators wish to use this resource opportunistically, they must do so in a manner that remains *completely transparent to the user*. This raises a new challenge in opportunistic use of phone storage that has not been addressed before in the opportunistic communication literature.

One way to make a device's RAM use transparent is to allow applications priority access to the memory, even if the memory is currently occupied by data yet to be forwarded. This means that data given to a device waiting to be forwarded can be overwritten by application data and may, thus, be lost. In this paper we consider the question of how much data should be moved when a contact opportunity arises between two devices in such a way to first maximise the data transferred while minimising the probability that this data will be overwritten when applications claim priority access over a mobile device's memory.

To motivate the problem further, we collected and analysed a trace of available smartphone memory (RAM) from a user over three days (see Fig. 1(a)). As we can see, the unused memory can be very dynamic and vary significantly over a short span of time. For example, at about the 7900th sample, we observe 800 MB of unused RAM, but it drops to only 300 MB during the next few minutes. If a node came in contact with this phone at the 7900th and forwarded 800 MB of data, the phone would have to wipe out 500 MB of that data to make room for user applications. This highlights the risks involved in using smartphone memory for carrying opportunistic data.

The decision making can be considered as a dilemma between the utilisation of the unused memory and the reliability of data transport. If too much of the unused memory is utilised, there may be excessive data loss due to memory reclaim by the user. On the other hand, if low data loss is desired, unused memory may not be exploited efficiently. In this paper, we propose three decision making algorithms of different sophistication to answer the question of how much data should be moved when a contact opportunity arises between two devices in such a way to first maximise the data transferred while minimising the probability that this data will be overwritten when applications claim priority access. The first algorithm is the simplest algorithm that greedily occupies all of the RAM that it finds unused at the time of contact. The second is more sophisticated and uses autoregression forecasting to predict the minimum amount of memory that will still be available until the data is forwarded to the destination. In the third, we propose the use of Bayesian Decision Theory to minimise the risk of data loss using knowledge from past observations.

Using 33 h of high-resolution RAM usage traces of two real smartphones over a 3-day period under a variety of usage scenarios, we evaluate and compare the performance of the proposed algorithms. Surprisingly, we find that autoregression forecasting of RAM usage cannot outperform the simplest algorithm that greedily occupies all of the RAM that is found unused at the time of contact. We show that Bayesian inference is very effective in minimising the risk of data loss in such uncertain environments and significantly outperforms the greedy approach as well as autoregression forecasting.

The rest of the paper is organised as follows. The decision framework is defined in Section 2. We present the three decision algorithms in Section 3. Data collection and performance evaluation are presented in Section 4. Related work is discussed in Section 5. Section 6 concludes the paper.

2. Decision framework

In this section, we present a framework to define the decision that a node with data to be forwarded has to make when it comes in contact with a mobile phone. The aim of the forwarding node is to decide how much data to be offloaded to the mobile phone to carry. We assume that the smartphone receiving the data will carry it for a *transit* period before forwarding it along the next hop. We will consider the available memory as an integral multiple of a base chunk size B megabytes. We use the following notations:

- C: memory capacity of the phone (it has a total $C \times B$ megabytes capacity)
- k: amount of memory not used by user applications (*available* memory) at time of contact; $0 \leq k \leq C$
- j: minimum amount of *available* memory during the entire transit period; $0 \leq j \leq C$
- i: amount of data the node decides to transfer to the phone; $0 \leq i \leq k$

The decision framework needs to consider the tradeoff between two types of losses: *data losses* (L_D) and *opportunity losses* (L_O). Data losses occur when the exploited unused memory to carry data is reclaimed by the user. In general, data losses have a higher chance of occurring if the source node offloads a higher amount of data to the phone. On the other hand, opportunity losses occur when there is available memory in the phone but the source does not make use of this unused memory. This tends to occur when too little data is offloaded to the phone. Note that although it is possible that new memory becomes available after the mobile phone has left the source location on the way to the destination location, we do not consider this as opportunity loss because there is no way that this memory could be used in the first place. Both losses are undesirable; the data loss decreases the reliability and retransmission of lost data is a waste of communication resource. On the other hand opportunity loss reduces the opportunistic network capacity. Given the tension between these two losses, the decision framework must find a way to balance between them.

Choosing i is the decision that will influence the opportunity loss and the data loss. Given i, j, and k, we have the following three cases:

- $i = min(k, j)$: $L_O = 0$; $L_D = 0$
- $i < min(k, j)$: $L_O = min(k, j) - i$; $L_D = 0$
- $i > min(k, j)$: $L_O = 0$; $L_D = i - min(k, j)$

A good decision is one that minimises both losses, i.e., achieves good throughput (low opportunity loss) with minimum data loss. In the following section, we propose and discuss three different decision algorithms with contrasting features and properties.

3. Decision algorithms

In this section, we define three different decision algorithms, Greedy, Autoregression, and Bayesian. We also consider an Oracle algorithm to explain the best possible outcome that could be achieved with any decision making.

3.1. Oracle

This is the ideal decision with $i = j$. As such, there is no opportunity loss and no data loss. To make such decisions, the source node has to have the perfect knowledge of the minimum amount of memory that will remain available throughout the journey, i.e., it needs to know j exactly. Clearly, this is not achievable, but it serves as a benchmark to compare different approaches.

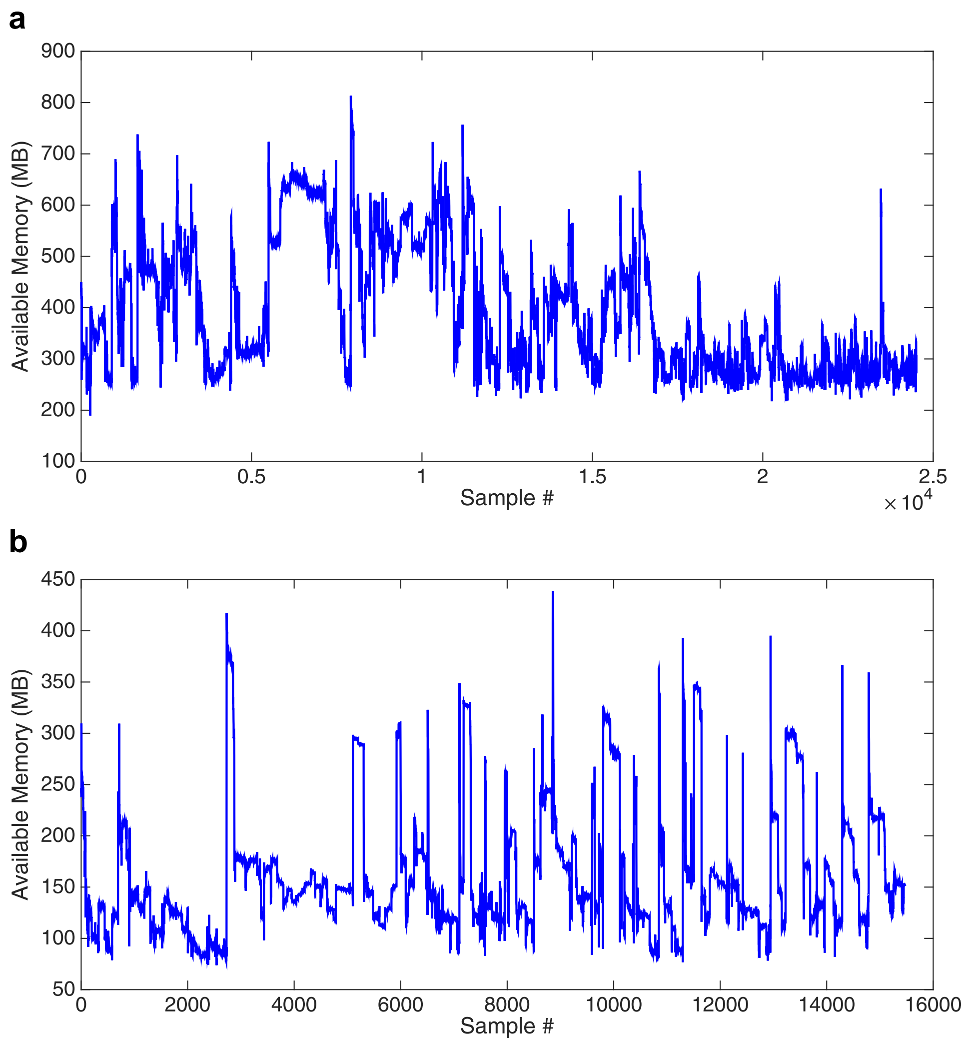

Fig. 1. The amount of unused memory (a) Samsung Galaxy S4 equipped with 2 GB of RAM (b) Samsung Galaxy S3 equipped with 832 MB of RAM.

3.2. Greedy

In this approach, we have $i = k$, i.e., the source node behaves greedily and uses up all the available memory that it finds at the source. This decision making is very simple to implement as the source node only needs to know the current available memory, which is readily obtained from an operating system call. This algorithm has no opportunity loss, but it is likely to incur heavy data loss if memory fluctuates significantly during transit.

Data loss can be reduced at the expense of increased opportunity loss by using a safety net. In particular, when the safety net is applied, Greedy does not use up the entire available memory (k), but leaves c amounts of available memory unused. More precisely, the Greedy decision with a safety net c can be defined as:

$$i = \begin{cases} k - c & \text{if } 0 \le c \le k \\ 0 & \text{if } c > k \end{cases} \tag{1}$$

The value of c is chosen by the network provider in an attempt to tradeoff between data and opportunity losses. We will study this tradeoff using memory traces obtained from mobile phones in Section 4. Safety net is also very simple to apply as the provider would decide on the value of c offline. Note that Greedy is a special case of Safety Net with $c = 0$.

3.3. Autoregression time series forecasting

In this approach, the available memory is sampled at equal intervals and the samples are considered as a time series $\{y(t)\}$ where t is a positive integer and $y(t)$ denotes the amount of available memory at the tth sampling instance. The goal of this approach is to predict j over a horizon of h future samples and then use the predicted value to decide the amount of data to load on the phone. For example, if $\hat{j}$ represents the predicted value of j, the source node will load $\hat{j}$ data units if it is smaller than k, i.e., we have $i = min\{k, \hat{j}\}$. Next, we show how $\hat{j}$ is obtained.

The first step is to analyse the data and establish the autoregression (AR) order that represents the memory availability time series best. AR model of order p is defined as:

$$y(t) = \mu + \rho_1 y(t-1) + \rho_2 y(t-2) + \cdots + \rho_p y(t-p) + \varepsilon(t) \tag{2}$$

where μ and ρ_j are the estimated coefficients of the AR model, and $\varepsilon(t)$ is zero mean Gaussian white noise. AR coefficients can be either estimated once offline using data from many different phones (a global model for all phones for all times), or they can be updated online at each sampling interval using the most recent observations from the same phone (phone and time specific model).

Assuming, the AR coefficients are available, the next step is to perform an h-step ahead forecasting, where 1-step ahead prediction is obtained as:

$$\hat{y}(n+1) = \mu + \rho_1 y(n) + \rho_2 y(n-1) + \cdots + \rho_p y(n-(p-1)) \quad (3)$$

The 2-step ahead prediction is:

$$\hat{y}(n+2) = \mu + \rho_1 \hat{y}(n+1) + \rho_2 y(n) + \cdots + \rho_p y(n-(p-2)) \quad (4)$$

If $h > p$, the h-step prediction follows the recursive model:

$$\hat{y}(n+h) = \mu + \rho_1 \hat{y}(n+h-1) + \rho_2 \hat{y}(n+h-2) + \cdots + \rho_p \hat{y}(n-(p-h)) \quad (5)$$

Once all the h forecasts are computed, j is predicted as:

$$\hat{j} = min\{\hat{y}(n), t < n \le t+h\} \quad (6)$$

AR is more complex and has higher executing overhead than Greedy, but it has the potential to yield superior performance in terms of lower losses if accurate prediction of j can be achieved. Like Greedy, the provider can also adjust opportunity-data loss tradeoff by applying a safety net c to $\hat{j}$ as follows:

$$i = \begin{cases} min(k, \hat{j}) - c & \text{if } min(k, \hat{j}) \ge c \\ 0 & \text{if } min(k, \hat{j}) < c \end{cases} \quad (7)$$

3.4. Bayesian decision

Bayesian decision theory [13] is a well-known tool that allows minimisation of risk under uncertainty. In our case, the risk can be defined as a cost of $\lambda(i|j)$ for making decision i given j is the minimum available memory over the horizon of interest:

$$\lambda(i|j) = \begin{cases} (i-j)c_d & \text{if } i > j \\ (j-i)c_o & \text{if } i < j \\ 0 & \text{if } i = j \end{cases} \quad (8)$$

where c_d and c_o are the costs for unit data loss and unit opportunity loss, respectively.

Assuming that we have the conditional probability $P(j|k)$ (we will describe how this conditional probability can be estimated from past memory measurement later in this section), we can compute the conditional risk $R(i|k)$ as:

$$R(i|k) = \sum_j \lambda(i|j) P(j|k) \quad (9)$$

This means that when the source node observes k available memory, it can quantify the risk of taking each action i. With Bayesian Decision (BD), the best action i_{opt} is the one that minimises the risk. Mathematically, this can be stated as:

$$i_{opt} = \arg\min_i R(i|k) \quad (10)$$

This optimisation can be solved easily. The only missing ingredient is how to compute the conditional probability $P(j|k)$, which we will discuss next.

Applying Bayes' rule, we obtain:

$$P(j|k) = \frac{P(k|j)P(j)}{P(k)} \quad (11)$$

We will show how we can learn the three probabilities on the right-hand side of the above equation from past memory measurements. Let t denote the current time which is the time at which the phone is at the source location and h the number of sampling intervals to travel from the source location to the destination location. We assume that we have measurements of the available memory in the time instances in the interval $[t-M-h, t]$. We consider time windows with $h+1$ measurements. The first time window is $[t-M-h, t-M]$, and subsequent time windows are $[t-M-h+1, t-M+1]$, $[t-M-h+2, t-M+2]$, ... until $[t-h, t]$. For the time interval $[t-M-h+\ell, t-M+\ell]$ (where $\ell = 0, \ldots, M$), we use $\hat{\omega}(t-M+\ell)$ to denote the minimum memory available in the time interval. We can estimate the probability distribution that the memory available is j in a time horizon of h by:

$$P(j) = \frac{\sum_{\tau=t-M}^{t} I[\hat{\omega}(\tau) = j]}{M} \quad (12)$$

where $I[\,\cdot\,]$ denotes the indicator function, which is 1 if its argument evaluates to true, and 0 otherwise.

Similarly, we can estimate the conditional probability $P(k|j)$ by:

$$P(k|j) = \frac{\sum_{\tau=t-M}^{t} I[(y(\tau-h) = k) \wedge (\hat{\omega}(\tau) = j)]}{\sum_{\tau=t-M}^{t} I[\hat{\omega}(\tau) = j]} \quad (13)$$

where "$\wedge$" denotes logical conjunction.

With Eqs. (11) and (12), we can obtain:

$$P(k) = \sum_{j=0}^{S} P(k|j)P(j) \quad (14)$$

One problem with using the above method to determine the probabilities is that the denominator of Eq. (11) which is obtained from Eq. (14) is equal to zero. This is a common difficulty in Bayes inference and is known as the *zero-frequency* problem. This problem can be avoided by using the m-estimate approach [14]. In this approach, Eq. (13) is replaced by:

$$P(k|j) = \frac{\sum_{\tau=t-M}^{t} I[(y(\tau-h) = k) \wedge (\hat{\omega}(\tau) = j)] + mp}{\sum_{\tau=t-M}^{t} I[\hat{\omega}(\tau) = j] + m} \quad (15)$$

where p is our prior estimate of the probability we wish to determine, and m is a constant called the equivalent sample size, which determines how heavily to weight p relative to the observed data. A typical method for choosing p in the absence of other information is to assume uniform prior. Thus, $p = \frac{1}{M}$ is the uniform probability and m is taken equal to 0.001 to have minimal impact on the estimation.

Table 1 summarises and compares the decision making algorithms. We see that Greedy is simple to implement, whereas AR as well as BD are inherently more complex as they require analysis of past observations. For a quantitative performance evaluation, in the following section, we simulate these algorithms using memory traces from real phones.

4. Performance evaluation

In the previous section, we have proposed and discussed three possible algorithms that providers can implement to decide about the amount of data they could load into a phone. We compared these algorithms in terms of their general expected behaviour. In this section, we perform a quantitative evaluation of these algorithms using RAM traces from real phones.

4.1. Data collection

We used two Samsung Galaxy smartphones for data collection. First phone (SP1) is of model S4 with 2 GB RAM and Android 4.3; Second one is S3 with 832 MB RAM and Android 4.1. The memory usage trace was collected with sampling period of 5 s using an application called "MKSysMon" freely available from Google Play. This application is capable of recording a number of items including available RAM which we used. The data was collected during the active hours of the day, i.e., from 9 a.m. to 8 p.m., over 3 and 2 days, which produced two traces of 24,504 and 15,469 samples of unused RAM for SP1 and SP2 respectively. The traces cover commonly used applications such as messaging, calling, email, watching video, web browsing and so on. The SP1 user was also instructed to occasionally use the phone to play intensive graphic games, namely Asphalt8 and Fifa2014, which are known to consume significant memory.

Table 1
Comparison of decision making algorithms.

Algorithm	Decision	OLR	DLR	Complexity
Oracle	$i = j$	0	0	N/A
Greedy	$i = k$	0	Could be high	Low
AR	$i = min\{k, \hat{j}\}$	Low if $\hat{j} \approx j$	Low if $\hat{j} \approx j$	High
Bayesian	$i = \arg\min_i R(i\|k)$	Can be adjusted by c_d and c_o	Can be adjusted by c_d and c_o	High

DLR and OLR are quantitative measures of, respectively, data loss and opportunity loss. They will be defined in Section 4.2.

Table 2
Trace statistics.

	Standard deviation σ	Mean μ	Coefficient of variation $C_v = \frac{\sigma}{\mu}$
SP1	123	394.4	0.31
SP2	61	160.7	0.38

The amounts of unused memory of the phone during the active hours of data collection are shown in Fig. 1 for both smartphones. As we can see, the unused memory can be very dynamic and vary significantly over a short span of time. For example for SP1, at some points, the unused memory can be as high as 800 MB, but it can drop to only 300 MB during the next few minutes. This dynamic behaviour of unused phone memory highlights the challenge involved in making optimal decisions about phone memory use.

Finally, Table 2 shows some basic statistics of the traces. In particular, it shows that SP2 has a higher coefficient of variance (C_v). The impact of high C_v on different decision making algorithms will be explored in the following subsections.

4.2. Performance metrics

Recall that in Section 2, we defined L_O and L_D as the amount of opportunity loss and data loss, respectively, for a particular decision made at a particular time for specific values of i, j, and k. As we have collected data from two phones with different memory capacities, we can normalise these losses as $\frac{L_O}{j}$ and $\frac{L_D}{i}$. As we have large number of samples collected from each phone, for a given horizon length of h, we calculate the average opportunity loss rate (OLR) over the entire trace as follows:

$$OLR = \frac{\sum_{t=M+1}^{N-h} L_O(t)}{\sum_{t=M+1}^{N-h} j(t)} \quad (16)$$

where $L_O(t)$ is the opportunity loss at the t^{th} sample, $j(t)$ is the minimum available memory over $[t, t+h]$, and N is the total number of samples in the trace. The first M samples of the trace are left out for initialisations, such as computing the conditional probabilities of the BD algorithm. Similarly, the data loss rate (DLR) of a trace is obtained as:

$$DLR = \frac{\sum_{t=M+1}^{N-h} L_D(t)}{\sum_{t=M+1}^{N-h} i(t)} \quad (17)$$

4.3. Performance of Greedy

As explained earlier, Greedy can be implemented with a safety net. For every sample values, we have calculated DLR and OLR for 21 different values of parameter c, ranging from 0 to 20 incrementing by 1. Note that $c = 0$ represents the Greedy algorithm, which has no opportunity loss, but may incur high data loss. A positive value of c would reduce data loss at the expense of some opportunity loss. The trade off between data and opportunity loss is plotted in Fig. 2 for the two traces collected from two different phones.

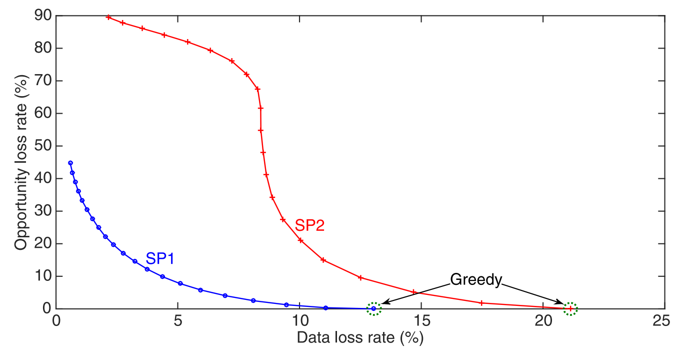

Fig. 2. Data loss vs. opportunity loss tradeoff achieved by applying a safety net to the Greedy algorithm.

First, higher losses are observed for the trace from SP2 compared to that of SP1. Recall from Table 2, that SP2 has a higher coefficient of variation than SP2. The increased variation leads to an increased deviation of j from k, which causes larger data loss as well as larger opportunity loss. For the Greedy algorithm, the SP1 trace has a data loss of 13%, whereas SP2 incurs 21% data loss.

The second observation is that the Greedy algorithm does not allow a linear tradeoff between data loss and opportunity loss. Opportunity loss increases non-linearly as reduction in data loss is attempted with larger values of the safety net parameter c. For example, for SP1, a reduction in DLR from 13% to 1% increases OLR from 0% to 47%. Again, the situation is worse with SP2 due to its higher variance.

4.4. Performance of AR

Recall that the first step for this algorithm is to establish the order of the AR. To do this, we compute the *Autocorrelation Function (ACF)* and the *Partial Autocorrelation Function (PACF)* of the traces (see Fig 3). ACF shows that the coefficients of correlation between the time series and the lags of itself are decaying slowly. This means that the higher order coefficients have decaying significance. The PACF captures the exact contribution of each lag and we can see that only the first coefficient is significantly higher than the rest and only the first three coefficients are higher than 0.05. The rest are very close to zero. We therefore select the order as 3, i.e., we consider an AR(3) to model the time series of available phone memory.

Next, we wanted to check whether a global AR model could be used to represent the time series collected from any phone at any time. In Fig. 4, we plot the three coefficients of the AR(3) model by considering each 1000 samples of the SP1 trace as a separate time series. We can clearly see, that the values of the coefficients are not identical over all time series. We have therefore implemented the online approach to computing the coefficients, i.e., they are updated each time there is a new sample added to the trace.

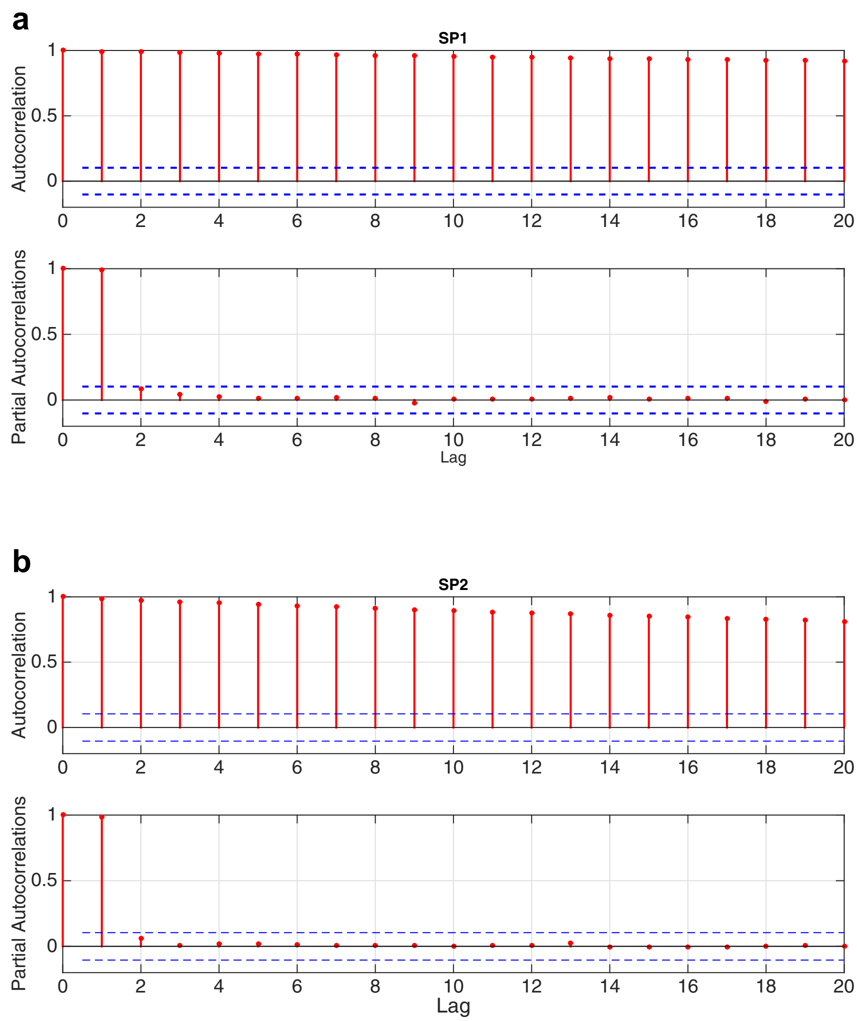

Fig. 3. Auto Correlation Function (ACF) and Partial Auto Correlation Function (PACF) of (a) SP1 (b) SP2.

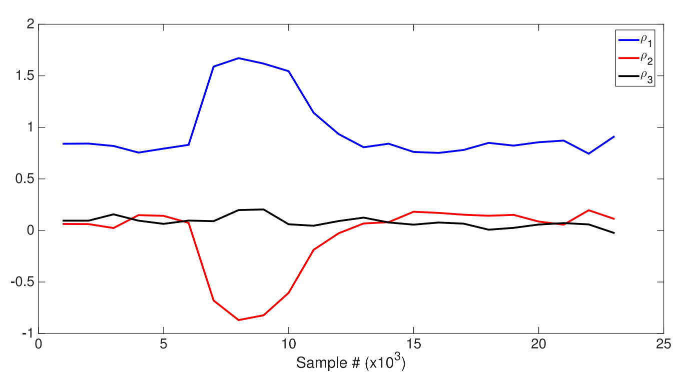

Fig. 4. Variation in AR coefficients for different parts of the time series.

Table 3
Performance of AR(3) for SP1.

Horizon (min)	RMS error of $\hat{j}$ (Chunk)	OLR (%)	DLR (%)
0.5	3.82	0.08	2.31 (2.5)
1	4.23	0.1	3.6 (4.06)
5	6.16	0.6	8.7 (10.5)
7.5	6.88	0.9	10.3 (13.05)

Finally, Table 3 shows the performance of AR(3) in terms of forecasting error, OLR, and DLR without any safety net ($c = 0$). The left-most column shows the length of the horizon in minutes. For 5-s sampling intervals, each minute equals to 12 steps for forecasting. For example, a 0.5 min horizon represents a 6-step ahead forecasting. Also note that the horizon length represents the transit time of the data in the phone, i.e., the longer the transit time, the longer the forecasting horizon, and vice versa.

The first observation we make from Table 3 is that the quality of forecasting, which is measured as root mean square (RMS) error of $\hat{j}$, deteriorates with increasing horizon length. This is a direct outcome of the fact that for $h > p$, i.e., when horizon is longer than the AR order, the forecasting is based on forecasted data. In such cases, any early forecasting error propagates further to the later forecasts. Second, the forecasting error leads to data loss and opportunity loss. The larger the forecasting error, the larger the losses. For example, for a 7.5-min transit time, a forwarding node would lose 10% of its data if AR(3) algorithm is used to load data to the phone.

Finally, we compare the performance of AR against Greedy in the last column, where the DLR values for Greedy are shown within parenthesis. We see that AR cannot outperform Greedy in any significant ways. AR achieves slightly less DLR compared to Greedy, but it does so at the expense of slight increase in OLR (recall that Greedy incurs no opportunity loss). This is a surprising outcome given the complexity and sophistication involved in AR compared to Greedy. This can be explained by the fact that the first coefficient of the AR

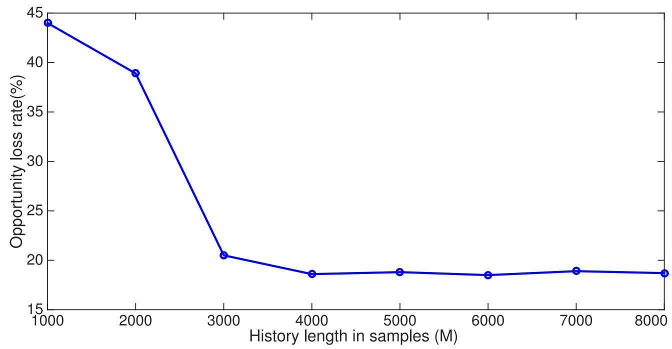

Fig. 5. Effect of history length on the performance of BD. The data loss rate has been kept 1% in this graph.

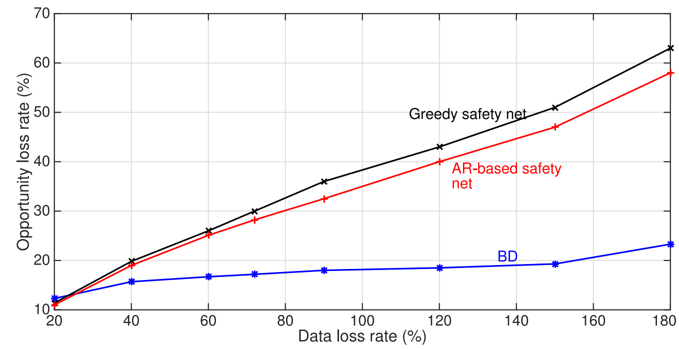

Fig. 7. Effect of the transit time (horizon) on the opportunity loss rate. The data loss rate has been kept 1% in this graph.

model of phone memory is close to 1 and all other coefficients are close to 0, which results in $\hat{j} \approx k$.

4.5. Performance of BD

The BD algorithm relies on past observations to derive its conditional probabilities, which are used to minimise the risk. We therefore first study the amount of historical data that would be necessary for this algorithm to make good decisions. Fig. 5 shows the OLR as a function of history length for a large value of c_d, which provides DLR close to 1%. As we can see, OLR decreases with increasing history, but the improvement stabilises at 4000 samples. This means that for the RAM dataset, keeping and using a history longer than 4000 samples or about 5.5 h is not useful. This is an encouraging result given the limited computational resources available in a mobile device. In our subsequent simulations, we therefore use a history of 4000 samples.

Next, to study the DLR-OLR tradeoff performance of BD, we run many simulations with varying values for the cost parameters c_d and c_o. Fig. 6 compares the tradeoff performance of BD against Greedy and AR with safety net. The superior performance of BD is seen clearly. Under data loss rate constraint, Bayesian reduces opportunity loss

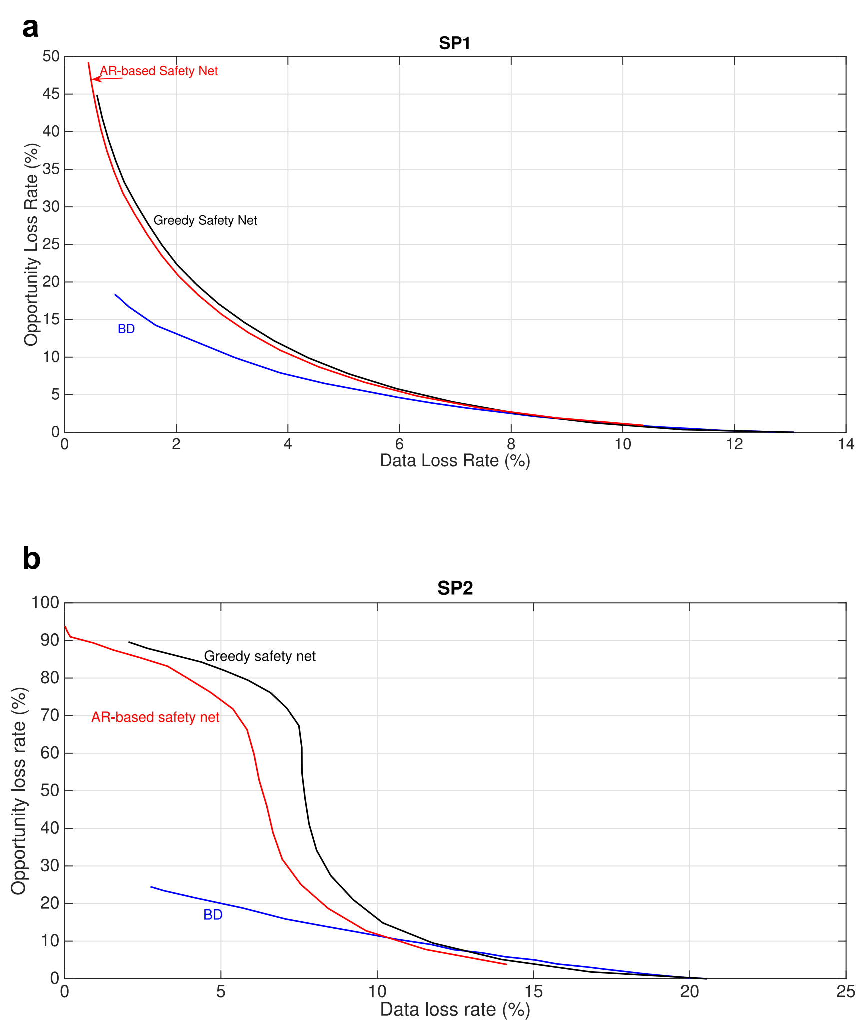

Fig. 6. Comparison of Bayesian decision against Greedy and AR for (a) SP1 (b) SP2.

significantly compared to both Greedy and AR. For example, for SP1, for a 1% data loss rate target, safety net has an opportunity loss rate of 42%, which means it can only utilise the 58% of unused phone memory. In contrast, the proposed BD algorithm has an opportunity loss of only 18%, or 82% utilisation of the unused memory opportunity. This significant increase in efficiency is the direct outcome of BD's superior capability to make decisions under uncertainty using historical observations.

Finally, the effect of horizon length (transit time) is explored in Fig. 7. We observe that for a given data loss rate target, the opportunity loss rate for both strategies increases linearly with the offload transit time. The rate of increase for BD is, however, three times slower than that of the Greedy and AR safety nets. For example, with BD, 80% of the unused phone memory can be exploited over a 12.5-min transit (150 samples ahead) while keeping the data loss rate below 1%, whereas the Greedy and AR can only exploit 50% for such long transit times.

5. Related work

Over the last decade, researchers have studied message ferrying or delay tolerant networking (DTN) for a diverse range of applications, including e-mail delivery to disconnected villages [15], restoring connectivity between partitioned mobile nodes [5,16], collecting data from a sensor network [17], and providing roadside-to-roadside (r2r) [18] data transfer service using vehicles as mobile ferries. Also in recent years, there is an increased interest in exploiting mobile user devices opportunistically as a part of offloading scheme to deliver data of cellular network users which is referred to as opportunistic traffic offloading [19–25]. Opportunistic networks have to use device storage to store data until it can be forwarded and require high throughput during contact times to transfer significant amounts of data during even short contacts. Most of mentioned works have either considered an infinite buffer [24] in the mobile device or assumed that the forwarding data is sufficiently smaller than the device's buffering capacity [23]. Other researchers [21] have assumed deterministic availability of the buffer space in the mobile device. The work in [25] focussed on dimensioning the buffering requirements in the mobile device for the optimal performance of the routing protocols used for transferring the data and [22] studied the buffer management issues arising due to carrying third party data in the personal device. While these works are not concerned with the type of memory, Coll-Perales et al. [26] and Zhao et al. [27] have proposed dynamic use of both RAM and internal memory in a mobile device for optimising the energy consumption and data access time. In term of throughput requirement, of the three types of memory available on a device, DRAM-based RAM technology provides the highest throughput [9–11]. None of these works addressed the efficiency and transparency challenge in opportunistic use of phone storage which is the focus of this paper. The closest work reported is our own work [28], which proposed the use of auto-regression model to predict the minimum amount of unused memory as a function of transit time. However, [28] did not propose or evaluate any decision strategy to study the opportunity vs. data loss tradeoff. This paper proposes a novel strategy, referred to as Bayesian decision framework, for efficient exploitation of unused phone memory for data offloading in cellular networks.

6. Conclusion and future work

Personal mobile devices with large RAM capacity are increasingly considered as effective data forwarding entities in opportunistic networks. However, because the primary purpose of RAM is to support user applications, usage of RAM for opportunistic networking must remain transparent to the user applications. One way to guarantee this transparency is to allow priority access to the RAM to user applications, even if the RAM is currently occupied by network data yet to be forwarded. This will require smart algorithms to decide how much data should be moved when a contact opportunity arises between two devices. This decision making is challenging because it has to maximise the data transferred while minimising the probability that this data will be overwritten when applications claim priority access to RAM. We have proposed three decision algorithms of different sophistications and compared their performances using real RAM traces from two smartphones. A surprising finding is that autoregression forecasting, albeit have been used successfully in many domains to predict future events, cannot outperform the simplest algorithm that greedily occupies all of the RAM that is found unused at the time of contact. This is because autoregression cannot predict the future RAM usage well using the time series. We show that Bayesian inference can effectively minimise the risk of data loss by using the knowledge from past observations and hence significantly outperforms the greedy approach as well as autoregression forecasting.

In this paper we only considered 'raw' data transfer, i.e., no error correction was explicitly considered for the decision making. When data loss is at risk, it may be useful to consider some form of error correction including network coding, which may allow to recover some lost data units. How to optimise decision making in the presence of such error correction would be an interesting research to follow up.

References

[1] L. Pelusi, A. Passarella, M. Conti, Opportunistic networking: data forwarding in disconnected mobile ad hoc networks, IEEE Commun. Mag. 44 (11) (2006) 134–141, doi:10.1109/MCOM.2006.248176.

[2] J. Wu, S. Yang, F. Dai, Logarithmic store-carry-forward routing in mobile ad hoc networks, IEEE Trans. Parallel Distrib. Syst. 18 (6) (2007) 735–748, doi:10.1109/TPDS.2007.1061.

[3] T. Spyropoulos, A. Picu, Opportunistic Routing, John Wiley & Sons, Inc., 419–452. 10.1002/9781118511305.ch11

[4] K. Wei, X. Liang, K. Xu, A survey of social-aware routing protocols in delay tolerant networks: applications, taxonomy and design-related issues, IEEE Commun. Surv. Tutor. 16 (1) (2014) 556–578, doi:10.1109/SURV.2013.042313.00103.

[5] W. Zhao, M. Ammar, E. Zegura, A message ferrying approach for data delivery in sparse mobile ad hoc networks, in: Proceedings of the MOBIHOC, Tokyo, Japan, 2004, pp. 187–198.

[6] R.C. Shah, S. Roy, S. Jain, W. Brunette, Data mules: modeling and analysis of a three-tier architecture for sparse sensor networks, Ad Hoc Netw. 1 (2–3) (2003) 215–233. Sensor Network Protocols and Applications. http://dx.doi.org/10.1016/S1570-8705(03)00003-9

[7] W. Zhao, Y. Chen, M. Ammar, M. Corner, B. Levine, E. Zegura, Capacity enhancement using throwboxes in dtns, in: Proceedings of the 2006 IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS), 2006, pp. 31–40, doi:10.1109/MOBHOC.2006.278570.

[8] S. Jain, K. Fall, R. Patra, Routing in a delay tolerant network, SIGCOMM Comput. Commun. Rev., 34, ACM, 2004, pp. 145–158.

[9] Samsung now producing four gigabit LPDDR3 mobile dram, using 20nm-class process technology, 2013, http://www.businesswire.com/news/home/20130429006782/en/Samsung-Producing-Gigabit-LPDDR3-Mobile-DRAM-20nm-class Samsung#.VTTJbFxtt95. accessed: 20.03.2015.

[10] JEDEC releases LPDDR4 standard for low power memory devices, 2014, http://www.jedec.org/news/pressreleases/jedec-releases-lpddr4-standard-low-power-memory-devices. accessed: 20.03.2015.

[11] Choosing the right NAND, http://www.micron.com/products/nand-flash/choosing-the-right-nand. accessed: 20.03.2015.

[12] C. Boldrini, K. Lee, M. nen, J. Ott, E. Pagani, Opportunistic networks, Comput. Commun. 48 (0) (2014) 1–4. http://dx.doi.org/10.1016/j.comcom.2014.04.007.

[13] J.O. Berger, Statistical Decision Theory and Bayesian Analysis, Springer Science & Business Media, 2013.

[14] T. Mitchell, Machine Learning, McGraw Hill, 1997.

[15] A.S. Pentland, R. Fletcher, A. Hasson, DakNet: rethinking connectivity in developing nations, Computer 37 (1) (2004) 78–83.

[16] C. Liu, T. He, K.-W. Lee, K. Leung, A. Swami, Dynamic control of data ferries under partial observations, in: Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), Sydney, Australia, 2010, pp. 1–6.

[17] I. Jawhar, M. Ammar, S. Zhang, J. Wu, N. Mohamed, Ferry-based linear wireless sensor networks, in: Proceedings of the IEEE Global Communications Conference (GLOBECOM), Atlanta, USA, 2013, pp. 304–309.

[18] A. Mansy, M. Ammar, E. Zegura, Reliable roadside-to-roadside data transfer using vehicular traffic, in: Proceedings of the IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS), 2007, pp. 1–6.

[19] F. Benbadis, F. Rebecchi, F. Cosnier, M. Sammarco, M. Dias de Amorim, V. Conan, Demo: opportunistic communications to alleviate cellular infrastructures: the FP7-moto approach, in: Proceedings of the 9th ACM MobiCom Workshop on Challenged Networks (CHANTS '14), ACM, New York, NY, USA, 2014, pp. 85–88, doi:10.1145/2645672.2645690.

[20] F. Rebecchi, M. Dias de Amorim, V. Conan, A. Passarella, R. Bruno, M. Conti, Data offloading techniques in cellular networks: a survey, IEEE Commun. Surv. Tutor. 17 (2) (2014) 580–603, doi:10.1109/COMST.2014.2369742.
[21] Y. Li, G. Su, P. Hui, D. Jin, L. Su, L. Zeng, Multiple mobile data offloading through delay tolerant networks, in: Proceedings of the 6th ACM Workshop on Challenged Networks (CHANTS '11), ACM, New York, NY, USA, 2011, pp. 43–48, doi:10.1145/2030652.2030665.
[22] H. Chen, W. Lou, Gar: group aware cooperative routing protocol for resource-constraint opportunistic networks, Comput. Commun. 48 (0) (2014) 20–29. http://dx.doi.org/10.1016/j.comcom.2014.03.022.Opportunistic networks
[23] P. Kolios, V. Friderikos, K. Papadaki, Mechanical relaying in cellular networks with soft-QoS guarantees, in: Proceedings of the 2011 IEEE Global Telecommunications Conference (GLOBECOM), 2011, pp. 1–6, doi:10.1109/GLOCOM.2011.6133642.
[24] F. Malandrino, C. Casetti, C. Chiasserini, M. Fiore, Offloading cellular networks through its content download, in: Proceedings of the 2012 9th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2012, pp. 263–271, doi:10.1109/SECON.2012.6275786.
[25] C. Boldrini, M. Conti, A. Passarella, Exploiting users? social relations to forward data in opportunistic networks: the hibop solution, Pervasive Mob. Comput. 4 (5) (2008) 633–657. http://dx.doi.org/10.1016/j.pmcj.2008.04.003.
[26] B. Coll-Perales, J. Gozalvez, V. Friderikos, Energy-efficient opportunistic forwarding in multi-hop cellular networks using device-to-device communications, Trans. Emerg. Telecommun. Technol. (2014), doi:10.1002/ett.2855.
[27] B. Zhao, V. Friderikos, Optimal stopping for energy efficiency with delay constraints in cognitive radio networks, in: Proceedings of the 2012 IEEE 23rd International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC), 2012, pp. 820–825, doi:10.1109/PIMRC.2012.6362897.
[28] S.A. Hoseini, A. Fotouhi, M. Hassan, C.T. Chou, M. Ammar, A message ferrying approach to low-cost backhaul in cellular networks, in: Proceedings of the 9th ACM MobiCom Workshop (CHANTS '14), 2014, pp. 25–30.