

# Design, implementation and performance evaluation of a publish-subscribe architecture for intermittently connected 802.15.4 networks

Donato Battaglino  
Università di Roma "Tor Vergata"  
Rome, Italy  
donato.battaglino@uniroma2.it

Andrea Detti  
Università di Roma "Tor Vergata"  
Rome, Italy  
andrea.detti@uniroma2.it

Andrea Bragagnini  
Telecom Italia  
Turin, Italy  
andrea.bragagnini@telecomitalia.it

Lorenzo Bracciale  
Università di Roma "Tor Vergata"  
Rome, Italy  
lorenzo.bracciale@uniroma2.it

Giuseppe Bianchi  
Università di Roma "Tor Vergata"  
Rome, Italy  
giuseppe.bianchi@uniroma2.it

Maura Santina Turolla  
Telecom Italia  
Turin, Italy  
maurasantina.turolla@telecomitalia.it

Nicola Blefari Melazzi  
Università di Roma "Tor Vergata"  
Rome, Italy  
blefari@uniroma2.it

## ABSTRACT

The aim of this work is to present Campus++, a location-based publish-subscribe system for intermittently connected delay tolerant networks, exploiting IEEE 802.15.4 devices, and taking into due account the severe constraints deriving from their physical characteristics. We describe our proposed architectural model and how we implemented our solution in a real test-bed. We investigate the trade-off between reduction of delay and storage requirements when nodes are memory-constrained. We provide some insights in this trade-off and propose simple rules to dimension the number of replicas per topic. To this end, we derive analytical models and we validate them with simulations. We point out that our system can be easily adapted to operate in a fully distributed, infrastructure-less way, allowing free communications e.g. in disaster areas or in areas in which "usual" communications means are either non-existent or intentionally made unavailable.

## Categories and Subject Descriptors

C.4 [Computer Systems Organization]: Performance of systems;  
C.2.2 [Computer Systems Organization]: Computer-Communication Networks—*Network Protocols*

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

PE-WASUN'11, November 3–4, 2011, Miami, Florida, USA.  
Copyright 2011 ACM 978-1-4503-0900-4/11/11 ...\$10.00.

## General Terms

Algorithms, Performance

## Keywords

Publish subscribe, IEEE 802.15.4, Delay Tolerant Networks, Optimization, Spray and Wait

## 1. INTRODUCTION

The small size and power consumption of IEEE 802.15.4 devices allows embedding them in GSM/UMTS U-SIM cards and/or SD cards. The availability of such technology for data exchange within mobile phones is very useful to complement GSM/UMTS services, providing location-based or proximity services, such as chat and advertisements in a commercial center, configuration data, micro-payments, access control. In addition, once that we have the availability of a free communication radio link, we can enlarge the assortment of offered services, supporting not only direct data exchanges between two users within the 802.15.4 connectivity range, but also communication among intermittently connected users. This is a typical scenario of so-called Delay Tolerant Networks (DTN) [2].

We argue that a communication paradigm well suited to this service environment is publish-subscribe [4]. Besides, this paradigm can satisfy the requirements of a community of users such as the one of a university campus, complementing other services such as voice and Internet access.

In this paper we present a location-based publish-subscribe system based on a DTN technology, named CAMPUS++, that enables a community of users equipped with IEEE 802.15.4 devices (e.g., mini SD, ZSIMs [1]) to exchange messages regarding specific "Topics". We call these messages data-samples; users that send data-samples regarding a topic are publishers of that topic; users in-

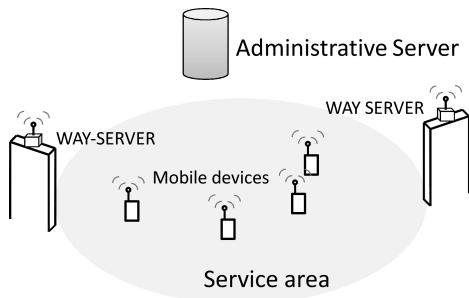
terested in receiving data-samples of a topic are subscribers of that topic. Users exploit services by running Campus++ software on their PDAs or mobile phones. The topics could have different *popularity*, i.e. different number of subscribers. The DTN is formed by nodes that use a multiple-copy routing approach [2] and have a *limited memory* to store and carry the published data-samples.

In addition to the description of our system, we observe that the presence at the same time of memory constraints and topics with different popularity issues a challenging optimization of the number of copies per data-sample. Indeed, as shown later, greedily exploiting all the memory resources available in the network nodes is counter-productive with respect to the system performance. Accordingly, we provide convenient formulae which allow to properly dimension the number of copies/replicas per data-sample. We verify the effectiveness of our analytical modeling by means of simulations. This work carries on the one presented in [12], introducing the following changes: i) it integrates contributions about system architecture, protocol and implementation description ii) it presents newer and simpler results for the analytical part described in section 4.1 iii) it provides new analysis and simulation results in section 5. We also presented a Campus++ demo in [13].

## 2. SYSTEM ARCHITECTURE AND PROTOCOL DESIGN

The Campus++ system architecture is composed of 802.15.4 mobile user devices, one or more 802.15.4 infrastructure nodes called "way-servers", and one administrative server (Figure 1).

Way-servers are placed in strategic locations crossed by users when they enter the service area. The roles of a way-server are: i) to be the publisher of control-data; ii) to provide mobile terminals with a loose system clock reference; iii) to inform the administrative server about the current status of systems parameters, such as the number of users, the number of subscribers per-topic, etc.

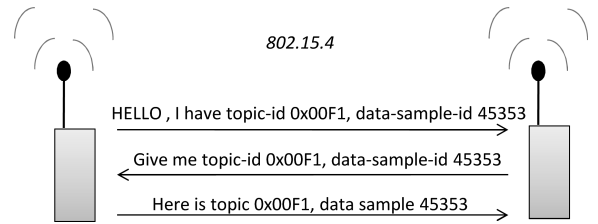


**Figure 1: System architecture, composed of an administrative server, one or more way-servers and several mobile devices**

As regards the DTN routing paradigm, we choose "Spray and Wait" [2] because of its simplicity, which suits the characteristics of IEEE 802.15.4 devices, and because of its native support for point to multi-point communication. Spray and Wait works as follows: when a new data-sample is published, it is replicated on R different nodes of the network, including the source, and this process is called spray phase. Subscribers can retrieve the data-sample when they are in direct radio contact with one of these R nodes.

We point out that "Spray and Wait" is not a fully-fledged protocol but only a routing scheme, thus, as we can see in Figure 2, we designed a publish-subscribe, receiver-driven full protocol that works as follows: a node pulls data-samples when it either wants to retrieve subscribed-to data-samples or when it wants to replicate data-samples. To this end, nodes periodically emit Hello messages,

which advertise the stored data-samples and if some of them need to be replicated [2].



**Figure 2: Example of message exchange for data-sample retrieval**

Moreover, since the available memory of nodes is limited, oldest data-samples have to be removed from the system when new data-samples are published on the same topic. This means that we need a way to distinguish newer data-samples from older ones. To support *data obsolescence* we loosely synchronize all devices with the clock reference provided by way-servers and use the publishing time to set the data-sample-id. A node that is replicating a data-sample marks it as "removable" if the node sees a neighbor node with a newer data-sample of the same topic. Removable data-samples will be indeed removed from nodes memory if space is needed for new data-samples, so that the system memory is efficiently used.

Time reference and other *data control information* are periodically distributed in the network by using the same publish-subscribe mechanism. To this end we define a special topic called "built-in topic", whose publishers are only the way-servers and whose subscribers are all user nodes. Built-in topic data-samples are replicated in an epidemic manner on all the nodes of the network. When a node periodically receives the control updates published on the built-in-topic, the node sets itself as being inside the service area and activates the DTN functionality (i.e. it starts participating to the replication process). Conversely, if a node does not receive the periodic update published on the built-in-topic, the node declares itself as out of the service area and switches off the DTN functionality.

Finally, to make possible the DTN routing we need a way to establish connections between intermittently connected nodes. Unfortunately the IEEE 802.15.4 Standard does not provide any kind of "ad-hoc" mode but it requires one and only one PAN coordinator for each personal area network (PAN). Given that security support is not (yet) considered in our scenario, and that we use unique 64bit extended addresses (i.e. MAC addresses), we do not find any reason to follow the rule of having a unique PAN coordinator. Thus, we force all nodes of the system to be PAN Coordinators.

For what concerns the implementation, we realized the DTN and publish-subscribe functionality on the TI CC2430 System on Chip (SoC)[3], which is an IEEE 802.15.4-capable device shipped with 128 KB of flash memory and 8KB of RAM.

The total code-size of the firmware is about 60 KB.

## 3. OPTIMIZATION PROBLEM DESCRIPTION

In this section, we introduce the considered scenario and assumptions, we state the delay optimization problem, and we illustrate at a glance the trade-offs that emerge in the optimization.

### 3.1 Scenario

We abstract from consider a wireless network composed of  $N$  mobile nodes that move within a service area according to a *random mobility model*. Each mobile node may have publishing and/or subscribing entities for topics; there are  $T$  topics and the  $i$ -th topic has  $S_i$  subscribing entities. Throughout the paper we refer to a publishing (subscribing) entity with the term publisher (subscriber). Moreover, we refer to the parameter  $S_i$  as *topic popularity*.

If a publisher and a subscriber for a same topic are within their radio coverage range, they can directly exchange data-samples. When this is not the case, we assume that data-samples are forwarded throughout the DTN using the “spray and wait” approach. Specifically, the data-samples for the different topics are replicated into the memories of nodes by using the so-called binary spray [2]. Binary spray spreads the replicas of a data-sample for the  $i$ -th topic as follows: the source generates  $R_i$  replicas (including the original data-sample); any node  $A$  with  $x > 1$  replicas<sup>1</sup> of the data-sample that meets a node  $B$  with no replica forwards  $\lfloor x/2 \rfloor$  of its copies to  $B$  and keeps the remaining copies.

Unlike [2], we focus on the case of nodes whose memory made available for the DTN forwarding process is bounded. Specifically, we assume that each node may store at most  $C_n$  data-samples. This sets forth the additional condition that a data-sample is replicated only whenever a neighbor node has at least one memory slot available. The introduction of a memory bound in a topic-based system (where data-samples are persistently generated) implies that a suitable mechanism must be introduced to control the data obsolescence such the one described in section 2. In what follows, for simplicity, we idealize such data obsolescence mechanism by assuming that all the data-samples for a given topic are removed from the memory when a new one is published for that same topic.

Finally, we assume that the time needed to exchange data-sample among neighbor nodes is negligible, and that a contact among nodes lasts for a sufficient amount of time to exchange all the data-samples required by the DTN forwarding rules.

### 3.2 Problem Statement and definitions

We recall that the main characterizing feature of the spray and wait protocol is the number of replicas  $R_i$  that are disseminated during the spray phase for each data-sample published to the  $i$ -th topic. In this work, we aim at globally optimizing said number of per topic replicas  $\mathbf{R} = [R_1 \cdots R_T]$ , for minimizing the average time  $D$  between the publication of a data-sample and its reception by subscribers. We call this delay the *mean delivery delay*. Obviously, this delay depends on the different topic popularity  $S_i$ . It is readily expressed as the weighted sum of the mean delays  $D_i$  perceived by subscribers of topic  $i$ , where the weights are the ratio between the number of subscribers for topic  $i$  and the total number of subscribers in the system<sup>2</sup>. In formula:

$$D = \sum_{i=1}^T \frac{S_i}{\sum_{j=1}^T S_j} \cdot D_i \quad (1)$$

<sup>1</sup>Note that when a node, because of the spray operation, holds  $x > 1$  replicas for a data-sample, all the  $x - 1$  copies do not account in the memory occupation of the node, i.e. only one copy of the data-sample is actually stored in the node memory, together with a counter that keeps track of the number of local replicas.

<sup>2</sup>We note that every node may be a subscriber for one or more topics, so the number of subscriptions could be greater than the number of nodes. Throughout this paper we consider one subscriber per subscription, so also the number of subscribers could be greater than the number of nodes.

It is convenient to express the number of replicas  $R_i$  in terms of two parameters which we refer to as *memory utilization* ( $\rho$ ) and *sharing factor* ( $\phi_i$ ). We define the memory utilization  $\rho$  as the ratio between the network-wide memory space used to store data-samples  $C_u = \sum_i R_i$  and the overall available memory  $C_{tot} = NC_n$  where  $N$  is the number of nodes in the network:

$$\rho = \frac{C_u}{C_{tot}} = \frac{\sum_{i=1}^T R_i}{NC_n} \quad (2)$$

Moreover, for each topic  $i$  we define sharing factor  $\phi_i$  as the fraction of the used memory space  $C_u$  assigned to topic  $i$  for its replicas:

$$\phi_i = \frac{R_i}{C_u} = \frac{R_i}{\sum_{j=1}^T R_j} \quad (3)$$

The number of replicas deployed for a given topic  $i$  is readily expressed in terms of these two above defined parameters as:

$$R_i = \rho \cdot \phi_i \cdot C_{tot} = \rho \cdot \phi_i \cdot NC_n \quad (4)$$

The problem thus consists in determining the optimal set of parameters  $\rho$  and  $\{\phi_i\}$  so that the delay  $D$  given in equation (1) is minimized, under the condition that no node is able to locally store more than  $C_n$  distinct replicas.

### 3.3 Discussion

At a first glance, the need to account for the memory utilization  $\rho$  may not be evident. Indeed, the per topic average delay  $D_i$  clearly depends on the number of deployed replicas  $R_i$ , and without memory constraints the delay performance would improve with a larger number of replicas. It could therefore naively appears that the sum of the replicas should be set to the maximum value allotted by the global memory constrains, i.e.,  $\rho = 1$ . However, a closer look reveals that this is not the case. Indeed, the optimal choice derives from trade-offs that involve both the memory utilization and the sharing of the memory space:

- **memory utilization** - as we increase the overall number of replicas in the system, the probability that the subscriber meets a node having one of these replicas increases and that tends to reduce the delivery delay. Conversely, as we increase the overall number of replicas in the system, the probability of finding a node with a free memory space decreases. Consequently, the duration of the spray phase is longer, the time required to spread in the system a given number of replicas increases and that tends to increase the delivery delay.
- **memory sharing** - let us consider a situation where each topic has the same number of replicas and hence all topics fairly share the system memory. As we increase the number of replicas for a given topic  $i$ , we have to decrement the replicas for another topic  $j$ , so as to maintain constant the memory occupancy. In doing so, the delivery delay of topic  $i$  decreases and the delivery delay of topic  $j$  increases. This delay unfairness could reduce the overall mean delivery delay ( $D$ ), when the number of subscribers  $S_i$  of topic  $i$  is greater than  $S_j$ . However, an excessive unfairness could cause so high delay penalty for subscribers of topic  $j$  as to vanish the benefits of delay reduction obtained by subscribers of topic  $i$ .

## 4. PROBLEM RESOLUTION

In the general case of different topic popularity, a delay optimization jointly targeting the parameters  $\rho$  and  $\{\phi_i\}$  appears nu-

merically cumbersome. Therefore, we resorted to a sub-optimal *sequential* approach which consists in two subsequent steps:

1. **memory utilization optimization:** first, we derive the memory utilization  $\rho$  which minimizes delay in the assumption that all the topics have the same popularity and, hence,  $\phi_i = 1/T$ ;
2. **sharing factor optimization:** second, we derive the sharing factors  $\phi_i$ , by assuming that the available memory is the one derived by the previous memory utilization optimization.

To simplify the analytical computation we consider exponentially distributed inter-meeting time between any pair of nodes. This implies that each node sees another (uniform distributed) random node every  $1/\lambda$  seconds (in mean).

#### 4.1 Memory Utilization Optimization

Under the assumption that all the topics have the same popularity and deploy the same number of per topic replica  $R = R_i$ , the subscribers of the different topics experience the same delay that is equal to the mean delivery delay, i.e.  $D = D_i$ . It therefore suffices to study the delivery delay for a single topic, by determining an analytic formula (eq. 10) which expresses the delivery delay as a function of the memory utilization, and numerically find the optimal value of  $\rho$  that provides the minimum delay value.

To make the computation of  $D$  analytically tractable we resort to the following assumptions:

- we approximate the binary spray mechanisms as an epidemic diffusion of data-sample replicas that stops when the target number of replicas  $R_i$  is reached.
- we assume only one spray at a time; in other words, during the spray phase of the considered topic, the amount of memory used for data-sample replicas of other topics does not change. This is a good approximation for the case of infrequently emissions of data-samples, whereas it should be considered as a worst case if data-samples production becomes frequent.

We model the evolution of spray and wait using the continuous time Absorption Markov Chain represented in figure 3. The state variable of the Markov chain is the number of deployed replicas of the considered topic. When a data-sample is first produced, the chain starts with state 1, meaning that only the producer has a copy of the data-sample. Whenever the producer "meets" another node, the chain moves to state 2, meaning that two nodes in the network carry a data-sample replica. When the chain reaches state  $R$ , meaning that the data-sample is copied over  $R$  nodes, the so-approximated spray phase stops.

In addition, a special state called "absorption" state models the reception of the data-sample by a selected subscriber for that topic, hereafter referred to as *target* subscriber. As such, the average data-sample delivery delay can be simply computed by determining the absorption time in the Markov chain.

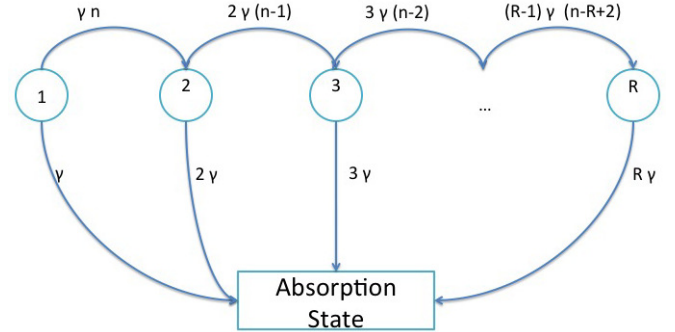
When the selected topic replication process starts, some nodes could have their memory full and so they cannot accept any more replicas. We refer to the probability that a node has at least one free place in its memory as  $f_R$ , and we conveniently call

$$n = (N - 2)f_R \quad (5)$$

the number of nodes with at least one free place in memory ( $N - 2$  are the number of nodes seen by a replica node without considering the subscriber node).

If we call  $\lambda$  the inter-meeting time frequency, we can also conveniently define the frequency of which a node encounters a specific node in the network as:

$$\gamma = \frac{\lambda}{N - 1} \quad (6)$$



**Figure 3: Markov chain representing the number of replicas in the system for a given data-sample.**

Each state  $i \in (1 : R - 1)$  in the chain has two outgoing transition rates: one towards the state  $i + 1$ , and the other towards the absorption state. These transition rates are derived and justified as follows:

- **transition rate  $\{i \rightarrow i + 1\}$ .** When the chain is in state  $i$ , a transition to state  $i + 1$  occurs whenever one of the  $i$  replica nodes encounters any of the remaining nodes except the target subscriber. Since the rate of at which one node encounters one another (specific) node is  $\gamma$ ,  $i$  are the replica nodes, and  $n - i + 1$  are the nodes that both do not hold the replica and that had at least one free space in its memory when the spray phase started, we conclude that the transition rate  $\{i \rightarrow i + 1\}$  is expressed by the product  $i\gamma(n - i + 1)$ .
- **transition rate  $\{i \rightarrow \text{absorption}\}$ .** Transition to the absorption state occurs when the target subscriber meets any of the  $i$  nodes holding the data-sample. In formula, this is expressed by  $i\gamma$ .

Using this chain, the mean delivery delay  $D$  of the considered topic can be calculated as the delay between the state 1 and the absorption state (state 0). We define  $\Delta_i$  and  $\delta_{i,j}$  respectively the mean time the system remains in the state  $i$  and the average time needed to move from the state  $i$  to  $j$ . The state equations can be written as:

$$\begin{aligned} \delta_{1,0} &= \Delta_1 + \frac{n}{n+1}\delta_{2,0} \\ \delta_{2,0} &= \Delta_2 + \frac{n-1}{n}\delta_{3,0} \\ \delta_{3,0} &= \Delta_3 + \frac{n-2}{n-1}\delta_{4,0} \\ &\dots \\ \delta_{R,0} &= \Delta_R \end{aligned} \quad (7)$$

The mean time  $\Delta_i$  the system remains in the state  $i$  is given by

the inverse of the outbound transition rates from each state:

$$\begin{aligned}\Delta_1 &= \frac{1}{\gamma(n+1)} \\ \Delta_2 &= \frac{1}{2\gamma(n)} \\ \Delta_3 &= \frac{1}{3\gamma(n-1)} \\ &\dots \\ \Delta_R &= \frac{1}{R\gamma}\end{aligned}\quad (8)$$

To conclude the analysis, it remains to derive the probability  $f_R$  used in 5 for deriving  $n$  from the number total number of nodes in the network  $n$ . This is defined as the probability that an encountered node has a free memory slot to accommodate the replica. We remark that  $f_R$  does not change with the chain evolution, and can be computed by quantifying the probability that a node has at least one free memory slot *before the start of the spray phase*, i.e. when there are only  $R(T-1)$  replicas in the system generated by the remaining topics. Specifically, note that the replicas generated for the considered topic do not affect the probability  $f_R$ . Indeed, nodes which already hold a replica for the considered topic are not accounted in the term  $f_R$ , but are accounted in the complementary term  $(n-i+1)$  within the state transition rate. It follows that  $f_R$  can be well approximated by:

$$\begin{aligned}f_R &= 1 - \left(\frac{N-1}{N}\right)^{CN-R(T-1)} \\ &= 1 - \left(1 - \frac{1}{N}\right)^{-(N)\left(\frac{R(T-1)}{N} - C\right)} \\ &\approx 1 - e^{-\frac{R(T-1)}{N} - C}\end{aligned}\quad (9)$$

This formula is obtained considering that before the start of the spray there are  $C_{tot} - R(T-1)$  memory slots available for the considered topic, distributed on node memories. Considering *free memory slots* as "tokens" randomly assigned to nodes, the probability that a node has at least a free memory slot is the complementary probability that the node has not been assigned any free space.

Solving equations and expressing the  $i$ -th Harmonic number as  $H_i$ , we obtain the following expression for the mean delivery delay  $D = \delta_{1,0}$  as a function of the parameter  $R$ :

$$D = \frac{H_R - 1}{(n+1)\gamma} + \frac{1}{\gamma R}\quad (10)$$

which can be rewritten in terms of the memory utilization  $\rho$  (and numerically minimized with respect to such unique variable) by simply substituting:

$$R = \rho \cdot \frac{C_{tot}}{T} = \rho \cdot \frac{N \cdot C_n}{T}$$

## 4.2 Sharing Factor Optimization

In this section we derive the optimal sharing factors  $\phi_i$ , under the assumption that the number of memory slots used in the network is fixed to the value  $C_u = \rho C_{tot}$ , where  $\rho$  is the memory utilization value which minimizes expression 10. Although (10) provides a delay expression which accounts for both the spray and the wait phases, to achieve a closed-form solution, similar to [2] we approximate the delay by neglecting the spray phase<sup>3</sup>. In this case, as shown in [2] and as a corollary of the previous analysis (i.e.

<sup>3</sup>Note that a more precise analysis, though necessarily addressed via numerical means, may be developed by leveraging the results derived in the previous section instead of using the approximation (11). In fact, it is easy to recognize that equation (10) is readily extended to the general case of different per topic delays  $D_i$  versus the

considering  $D \approx \delta_{R,0} = \Delta_R$ ) the mean delivery delay for topic  $i$  reduces to the very simple expression:

$$D_i = \frac{1}{\gamma R_i} = \frac{1}{\gamma C_u \phi_i}\quad (11)$$

Consequently, to compute the sharing factors  $\phi_i$  we need to solve the following constrained optimization:

$$\min_{\phi} \sum_{i=1}^T \frac{S_i}{\gamma C_u \phi_i \sum_j S_j}\quad (12)$$

$$s.t. \quad \sum_i \phi_i = 1\quad (13)$$

A closed form solution can be accomplished by means of a Lagrange approach. As a result, we obtain the following simple result:

$$\phi_i = \frac{\sqrt{S_i}}{\sqrt{S_1} + \dots + \sqrt{S_T}}\quad (14)$$

This result states that, for a relatively large number of nodes compared to the number of deployed replicas, i.e. when the spray time is small with respect to the wait time and hence approximation (11) is tight, the number of deployed replicas per topic is proportional to the square root of the topic popularity, namely the number of topic subscribers.

## 5. PERFORMANCE EVALUATION

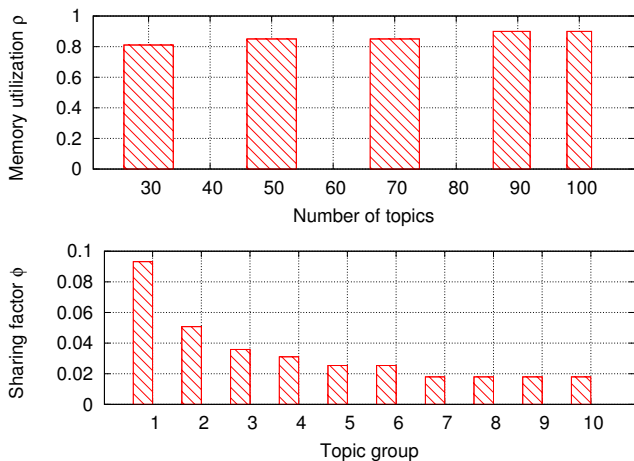
To validate the effectiveness of the proposed optimization and the modeling results, we developed an ad-hoc event-driven simulator. The simulator has been written from scratch and its code is available at [5]. As reference scenario, we consider  $N = 100$  nodes that move in an area of  $500 \times 500 \text{ m}^2$ . The radio coverage range is set to 50m, simulation duration is 500000 seconds. Every node has enough memory to store up to 10 different data-samples. We used movement traces in which the mean inter-meeting time between a specific pair of nodes is 1400s (i.e.  $N/\lambda = 1400$ , the inter-meeting time between any pair of nodes hence being  $1/\lambda = 14$ s).

In the simulator data-samples are generated as follows. For each topic, a first data-sample is generated by a randomly chosen publisher node at the start of the simulation (initial random offsets being used to avoid synchronized start). Then, every topic proceeds in parallel by independently generating subsequent data-samples. Each new data-sample is generated by a topic only when all the subscribers have received the previous data-sample. Specifically, after a 300s supplementary time gap at the end of the previous data-sample delivery phase, a new random publisher is selected for the considered topic.

We carried out the three kinds of performance evaluation, aimed at providing insight about the following aspects:

- effectiveness of the overall optimization (memory utilization + sharing factor) with respect to a not-optimized system;
- performance trade-offs related to the memory utilization;
- performance trade-offs related to the sharing factor.

corresponding  $\phi_i$ , as it depends only upon the  $\phi_i$  value for the considered topic. Such dependence is indeed confined to the exponent in equation (9), where we note that  $C_{tot} - R(T-1)$ , in the general case of different popularity, can be rewritten as  $C_{tot} - C_u + C_u \phi_i$ .



**Figure 4: Memory utilization configuration versus the number of topics and sharing factors in case of 30 topics**

## 5.1 Optimization Effectiveness

We consider nodes moving according to the Random WayPoint (RWP) model [11] with constant speed of 1 m/s and a constant pause time of 5 s. We considered 5 scenarios with different number of topics, i.e.  $T = 30, 50, 70, 90, 100$  topics. The topics have different popularities; we group the topics in 10 different groups of equal size and set the popularity of each group according to a Zipf distribution [6], with parameter  $\alpha = 1.8$ . The topics of the same group have the same popularity (a topic of the first group has 27 subscribers, a topic of the last group has 1 subscriber).

The subscriptions are uniformly assigned to nodes, so that nodes without any subscriptions are only used for store-carry-forward DTN operations.

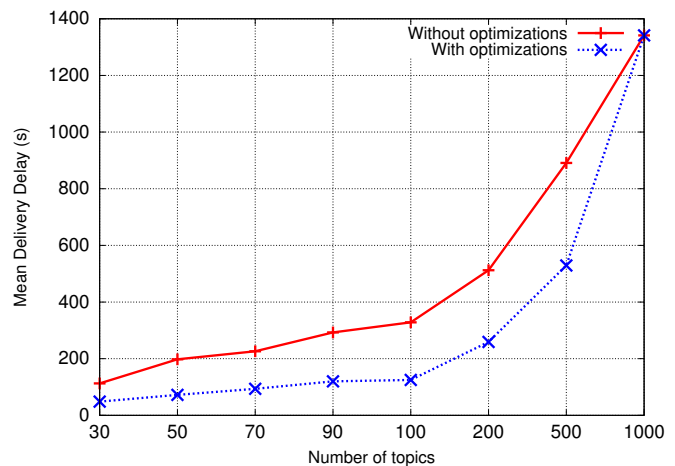
The upper plot of the figure 4 shows the values of the memory utilization  $\rho$  resulting from the optimization versus the different number of topics. We observe that, varying the number of topics, the optimal memory utilization remains in the interval to 0.80-0.9; this tells us that it is not convenient to use all the available memory.

The lower plot of the figure 4 shows the sharing factors  $\{\phi_i\}$  of the different topics in case of 30 topics. Those topics are divided in 10 groups (3 topics for each group) and popularities follow the previously described zipf distribution. We observe that the sharing factors are proportional to the topic popularities and this proportion depends on the square root of the number of subscriptions as stated in eq. (14).

In figure 5 we report the comparison of the mean delivery delay for an optimized a non optimized system. In the non-optimized system, all the available memory is used ( $\rho = 1$ ) and the memory spaces is equally distributed among topics (i.e.  $\phi_i = 1/T$ ).

We observe that the performance improvement provided by the sub-optimal configuration increases with the number of topics. However, even if it is not reported in the plot, this improvement does not always grow as the number of topic. As an example, if we consider the extreme case of 1000 topics, each topic has the possibility of deploying only 1 replica, thus the optimization returns the same configuration of the non-optimized system (i.e. 1 replica per-topic) and the delay performances are equals.

In general, we argue that the presented optimization is not effective in cases of very little or very large amount of topics (respect to the number of nodes an their capacity), but in the cases of



**Figure 5: Mean delivery delay with and without optimal configuration varying the number of the topics**

“moderate” number the presented optimization provides a valuable performance improvement. In fact:

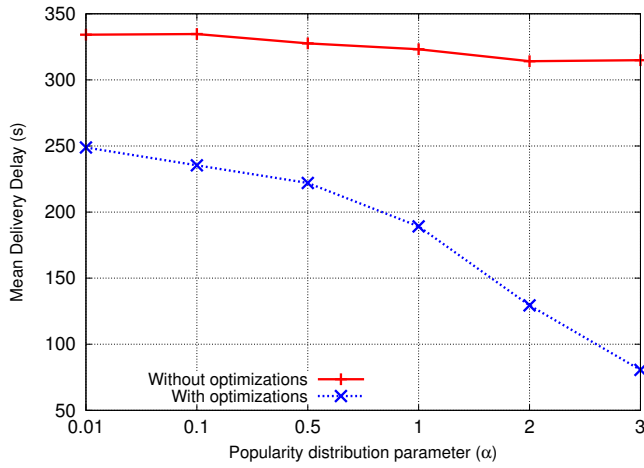
- in case of very few amount of topic, both the optimized and non-optimized systems put a large amount of replicas for each topic. This results in good delay performance of the non-optimized system so the improvement carried by the optimization is very limited (see for example the case of 20 topic in figure 7).
- in case of many topics, as for the case  $T = 1000, C_n = 10, N = 100$ , the optimization is not effective because the only viable solution to diffuse all the data-samples is to use only one replica per topic (i.e., no replication).
- in case of a moderate number of topics when the previous conditions do not occurs, the presented optimization provides valuable performance improvements.

In figure 6 we plot the comparison of the mean delivery delay of optimized and non optimized system varying the distribution of the popularity. In particular, we varied the distribution parameter  $\alpha$  previously described so that for  $\alpha = 0.01$  we have almost a flat popularity (about 5 subscribers for all the topics) and for  $\alpha = 3$  we have a very popular topic groups (with 39 subscribers) while the others are unpopular (from 3 to 1 subscribers for each of the remaining groups). As we expected, as  $\alpha$  grows the optimization becomes more and more effective and the gain compared to the unoptimized case grows as well. As  $\alpha$  decreases, the gain due to the sharing factor optimization decreases and tend to vanish when  $\alpha$  is very small so that for  $\alpha = 0.01$  remains only the effects of the memory utilization optimization.

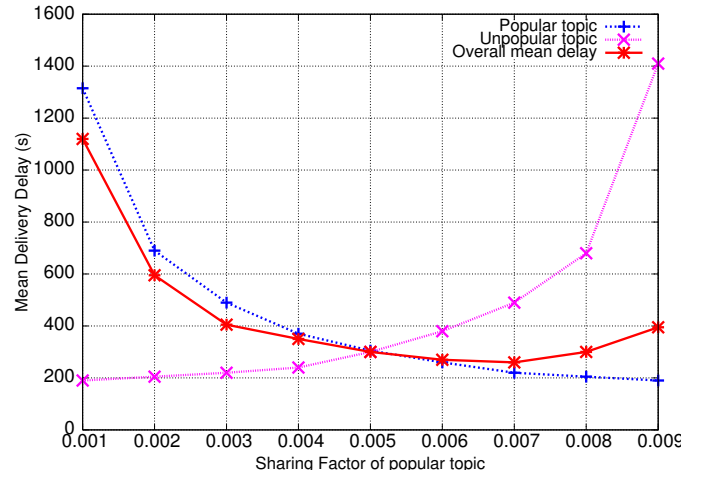
## 5.2 Memory Utilization Trade-offs

In this section we analyze the performance trade-offs related to the memory utilization.

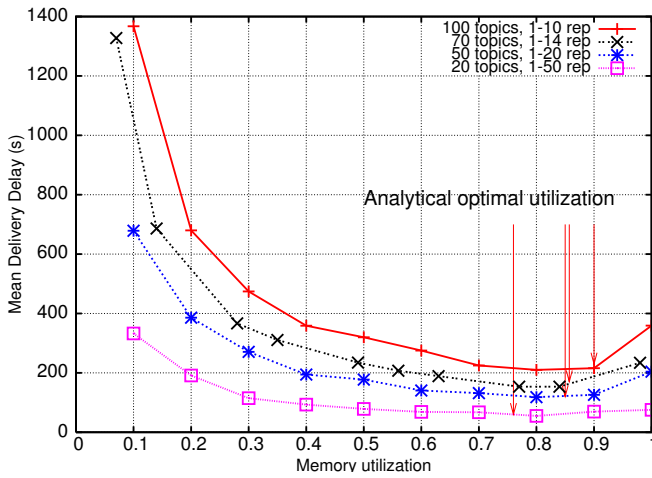
We consider scenarios where the topics have the same popularity, then  $\phi_i = 1/T$ . We feed the simulator with a trace that follows the random waypoint mobility model previously described. Figure 7 shows the mean delivery delay varying the memory utilization and the number of topics. As discussed in section 3.3, the graph confirms the presence of an optimum point after which the benefits of introducing more replicas in the system are overwhelmed by a



**Figure 6: Mean delivery delay with and without optimal configuration varying the popularity distribution parameter  $\alpha$**



**Figure 8: Mean delivery delay for the case of 100 topics equally divided in two groups, named popular and unpopular**



**Figure 7: Mean delivery delay varying the number of topics and memory utilization  $\rho$**

longer spray phase. The optimum point varies with the number of topics and the performance worsening due to memory saturation increases with the number of topics, as we can see for instance by comparing the difference between optimal memory utilization and  $\rho = 1$  for the cases 100 topics and 20 topics.

The reason is due to the different number of replicas per topic in these two cases, that respectively varies in the ranges 1-10 and 1-50. When we add more replicas in the system, the spray time increases as well. During the spray process, the first replicas are diffused more rapidly, while the latest ones are diffused more slowly. In fact, data-samples can be replicated only on nodes with available memory *and* that are not already holding a replica for that data-sample. However, when the spray time become significantly high, several replicas are already displaced in the case of 20 topics, while just few replicas are spread for the case 100 topics. For this reason the performance gap between the optimum point and the full memory occupation is smaller in the case of 20 topics and bigger in the case of 100 topics.

The graph also shows by arrows the values of memory utilization (i.e. values of the x-axis) returned by the utilization optimization

(sec. 4.1). As we can see, the optimization succeeds in finding the optimum groups are the ones presented point in all the presented cases.

### 5.3 Sharing Factor Trade-offs

In this section we analyze the performance trade-offs related to the sharing factor.

We consider a scenario where we have 100 topics equally divided into two groups, named "popular" and "unpopular": topics belonging to the popular group are have 50 subscribers, while topics belonging to the unpopular group have 10 subscriber.

Figure 8 represents the delivery delay of popular and unpopular topics and the overall average delivery delay ( $D$ ) varying the sharing factor of the popular topics while keeping  $\rho = 1$ . Each point  $x$  of the x-axis represents the sharing factor for one popular topic ( $1/1000 \leq x \leq 9/1000$ ); in that point each unpopular topic has a sharing factor equal to  $(10 - x)/1000$ . We observe that the minimum of the average delivery delay is achieved with a sharing factor of 7/1000 for popular topic, and 3/1000 to unpopular ones. This result confirms the one given by formula (14) that finds the optimal value in the point 0.00690983.

## 6. RELATED WORK

Several DTN routing strategies have been proposed so far. If we classify these strategies according to the number or replicas introduced in the network, we have single-copy routing schemes, where only one single copy of a given message is routed and forwarded, and multiple-copy routing schemes, where multiple copies of the same message are routed and forwarded independently.

The single-copy routing scheme proposed in [7] belongs to the first category. In this work, authors propose a routing scheme called "seek and focus" where a single copy of the message is passed by nodes according to an utility function that depends from the latest time a node encounters the destination. In the seek phase, if the utility around the node holding the data is "low", the node hands over the data to a neighbor node, with probability  $p$  until a node with a high utility is found. When this node is found, the focus phase starts and nodes use utility-based forwarding to deliver the message to the final destination.

With regard to the multiple-copy routing schemes, one possible approach is to distribute a message to all or almost all the nodes



in the network. This is the case of [8] where authors propose the so called epidemic routing. This approach consists in distributing messages to node called “carriers” that are within a connected portions of ad-hoc networks. During their movements, carriers will come into contact with another connected portion of the network thus “infecting” additional islands of connected nodes.

The probabilistic routing approach proposed in [9] called PROPHET (Routing Protocol using History of Encounters and Transitivity) aims to overcoming the two main limitations of the epidemic routing approach: the assumption of infinite buffer space and bandwidth in the nodes and the inability of predict user movements.

To overcome such limitations, PROPHET exploits a forwarding strategy based on a probabilistic metric, the “delivery predictability”: nodes that are often encountered have a high delivery predictability and delivery predictability reflects the transitive property, according to which if node  $a$  frequently encounters node  $b$ , and node  $b$  frequently encounters node  $c$ , then node  $c$  probably is a good node to forward messages destined for node  $a$ .

There exists also a multiple-copy routing version of the single-copy routing scheme proposed in [7]. This scheme has been presented by the same authors in [2], it is called as “spray and wait” and it is the one we used in this paper. In the same work, the authors propose a second scheme, denoted as “spray and focus”, which consists of two different phases as well: i) the spray phase, which is the same as the spray phase in the “spray and wait” scheme, ii) the focus phase, during which each of the  $L$  message copy is independently routed according to the single-copy scheme proposed in [7]. In addition, they provide some insight on how to choose the number of replica to spray in the network to achieve a required expected delay (expressed as a multiple of the optimum minimum delay).

For what concern DTN routing used with the publish-subscribe communication paradigm, we cite SocialCast [10]. This work relies on the notion of utility  $U$  of a node  $N$  with respect to interest  $I$ , that represents how good a carrier  $N$  is to carry messages matching the interest  $I$ . The basic assumption is that hosts which have same interest spend time co-located, so that the SocialCast routing aims at exploiting as carrier for messages hosts which have been co-located often with the interested. Kalman filter forecasting techniques are used to predict the future evolution of the movement based on previous observations on some attributes characterizing social behavior.

## 7. CONCLUSIONS

In this work we presented system architecture and the protocol design of Campus++, a location-based publish subscribe system for intermittently connected networks. We tackled the problem of how to optimize the mean delivery delay of DTN messages. We considered scenarios where i) the DTN nodes make available only a limited memory space for routing purposes, and ii) topics have different popularity, i.e. different number of subscribers. We focused on spray-and-wait routing scheme and therefore the only configuration parameter that could be optimized is the number of replicas per topic.

Given that providing an holistic approach results too complex, we split the optimization problem in two parts and we provided answers to these basic questions: i) which is the optimal amount of available memory to use for replication purposes (memory utilization), and ii) which is the right way to share this amount of memory among topics (sharing factor). In doing so, albeit we derived a sub-optimal solution, we derived handily and effective formulas. Indeed, with the exclusion of extreme cases of too few or too many

topics, the optimization provides a reduction of the mean delivery delay up to 60%, with respect to a non-optimized system

Finally, we argue that our optimization results can be used not only in case of spray-and-wait routing but also in cases of other multiple-copy routing schemes, which have the number of replicas as input parameter.

## 8. REFERENCES

- [1] M. Turolla, E. Alessio, *ZSIM enabling innovative services to improve quality of life*, white paper on line available at [www.zigbee.org/imwp/download.asp?ContentID=10403](http://www.zigbee.org/imwp/download.asp?ContentID=10403)
- [2] T. Spyropoulos, K. Psounis, C. S. Raghavendra, *Efficient routing in intermittently connected mobile networks: the multiple-copy case*, IEEE/ACM Transactions on Networking (TON), v.16 n.1, p.77-90, February 2008
- [3] CC2430 web site, <http://focus.ti.com/docs/prod/folders/print/cc2430.html>
- [4] P. T. Eugster, P. A. Felber, R. Guerraoui, A. Kermarrec, *The Many Faces of Publish/Subscribe*, in ACM Computing Surveys, V. 35 - I. 2, 2003
- [5] <http://stud.netgroup.uniroma2.it/lorenzo/campus++/simulator.tar.bz2>
- [6] S. Voulgaris, E. Rivière, A.M. Kermarrec, M. Van Steen, *Sub-2-sub: Self-organizing content-based publish and subscribe for dynamic and large scale collaborative networks*, Research Report RR5772, INRIA, Rennes, France, December 2005.
- [7] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, *Single-copy routing in intermittently connected mobile networks*, in Proc. IEEE Conf. Sensor and Ad Hoc Communications and Networks (SECON), 2004
- [8] A. Vahdat and D. Becker, *Epidemic routing for partially connected ad hoc networks*, Duke Univ., Durham, NC, Tech. Rep. CS-200006, Apr. 2000.
- [9] A. Lindgren, A. Doria, and O. Schelen, *Probabilistic routing in intermittently connected networks*, SIGMOBILE Mobile Comput. Commun. Rev., vol. 7, no. 3, 2003.
- [10] P. Costa, C. Mascolo, M. Musolesi, G. P. Picco, *Socially-Aware Routing for Publish-Subscribe in Delay-Tolerant Mobile Ad Hoc Networks*, in IEEE Journal of Selected Areas in Communication (JSAC), vol. 26, no. 5, pp. 748-760, June 2008
- [11] T. Camp, J. Boleng, V. Davies, *A Survey of Mobility Models for Ad Hoc Network Research*, Wiley, Wireless Communications & Mobile Computing, vol. 2, 2002
- [12] L. Bracciale, D. Battaglino, A. Detti, G. Bianchi, N. Blefari-Melazzi, A. Bragagnini, M. S. Turolla: *Delay Performance of a Publish Subscribe system deployed over a Memory-Constrained, Delay Tolerant Network*, in the proceeding of Med Hoc Net 2011
- [13] D. Battaglino, L. Bracciale, A. Detti, F. Lo Piccolo, A. Bragagnini, M.S. Turolla, N. Blefari Melazzi: *Campus++: a publish-subscribe architecture for intermittently connected 802.15.4 networks* Demo Session. To be presented in SECON, June, 2011, Salt Lake City, Utah, USA