Publish/Subscribe Architecture for Mobile Ad hoc Networks

Cristiano G. Rezende
Federal University of Minas Gerais
Belo Horizonte, Brazil
rezende@dcc.ufmg.br

Bruno P. S. Rocha
Federal University of Minas Gerais
Belo Horizonte, Brazil
bpontes@dcc.ufmg.br

Antônio A. F. Loureiro
Federal University of Minas Gerais
Belo Horizonte, Brazil
loureiro@dcc.ufmg.br

ABSTRACT

Publish/Subscribe architectures have been widely studied and applied in wired networks. However, their deployment on mobile ad hoc networks still presents a lot of challenges. This work proposes and analyzes a solution for such networks using the nodes' movement to disseminate publications to the whole network with few transmissions. Our proposal does not build dissemination trees which incur in a high cost to keep them updated due to constant changes in the topology, and does not even require beacons exchanges in order to sustain a neighborhood table. Our experiments show that we are able to achieve better results than a gossip-based algorithm and other solutions found in the literature.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design

General Terms

Distributed Algorithms

Keywords

Publish/Subscribe, MANETs, mobile computing

1. INTRODUCTION

Publish/Subscribe (P/S) is a paradigm which has already been intensively studied. It offers a totally decoupling way of communicating since the destination address is not needed and the parts involved in the process may not be on-line at the same time.

Several proposals made a deep study of P/S architectures for fixed networks [4, 15]. However, the use of this paradigm in mobile ad hoc networks (MANETs) offers many new challenges, manly due to the highly dynamic topology and the scarce resources. Nevertheless, architectures based on this paradigm can use their time and space decoupling properties to solve many problems faced on MANETs, like intermittent connection and heterogeneity of wireless interfaces.

Several existing solutions consider the availability of a powerful infrastructure which does not exist in most MANETs (or incurs spending a lot of limited resources, like energy and bandwidth, to keep it). The main problem is how to make the produced information reach all interested consumers with as few as possible transmissions.

This work is based on the idea that mobility increases the connectivity of MANETs [7]. Nodes will be responsible for disseminating locally received publications to different areas of the network after they move. Therefore, with sufficient movement, publications will be able to travel the whole network and reach interested nodes spread through it.

The proposed solution is suitable to asynchronous applications which can tolerate delays of dozens of minutes. Examples of such applications include e-mail, database synchronization between a mobile terminal and a central database, and certain types of event notifications [7]. Thus, the main contribution of this work is the solution of the delivery task achieving high match rates with few transmissions.

The rest of this paper is organized as follows. Section 2 presents some state of the art solutions. Section 3 describes in details the architecture. Experiments and their results are presented in Section 4. Finally, Section 5 summarizes the paper and describes some ongoing and future work.

2. RELATED WORK

There are several proposals of Publish/Subscribe architectures for wired networks. In [6], Eugster et al. make an extensive analysis of the P/S paradigm and compare it to other paradigms based on the three decoupling dimensions of the P/S paradigm: space, time and synchronization. Carzaniga et al. [4] have studied deeply this problem using modeling and developing of an expressive and scalable event notification service for wide-area networks called Siena. Other proposals [2, 15] also managed to discuss P/S architectures for wired networks, focusing on the need of scalability for event notification services developed on P/S architectures.

Even though the P/S problem is already widely covered in wired networks, the wireless environment presents many other challenges (e.g., limited resources, disconnection, mobility), which need further research in order to find new adequate solutions. Cugola et al. [5] discuss about how P/S paradigm can address these challenges that often are raised by mobile applications. Most proposals that make use of the P/S architecture in wireless networks are actually adaptations of existing architectures for wired environment to

SAC'08 March 16-20, 2008, Fortaleza, Ceará, Brazil

handle mobile producers and consumers connected to the network via access points [3, 10]. The idea is to have a broker, composed by more powerful computer devices connected to a wired network, which is responsible to manage incoming subscriptions and publications and to handle eventual disconnections and movement of mobile devices.

Huang et al. [9] studied the problem of applying P/S paradigm to MANETs. They describe different algorithms to build dissemination trees used to deliver publications to the proper interested consumers. However, that approach has the drawback that trees have to be constantly rebuilt due to the highly dynamic topology of a MANET. Baehni et al. [1] make use of the mobility, inherent to MANETs, to improve the performance of a topic-based publish/subscribe architecture. The main disadvantages are that the solution relies on a network where a high percentage of nodes have the same interests and the need of constant beacon exchanges in order to keep the nodes' neighborhood table updated.

3. ARCHITECTURE

The idea of this work is to use mobility to notify interested subscribers of messages sent by publishers in a mobile ad hoc network. The goal is to deliver publications to all interested parties using the minimum amount of broadcasts (messages). Even though its expressiveness, filters, merging techniques and other factors are of major importance to event notification services, the proposed solution focuses only on the delivery task of such services. An important assumption is that mobile devices are able to know if they are moving or not [12, 13].

This work develops a content-based publish/subscribe architecture for mobile ad hoc networks called PSAMANET. It is content-based because signatures describe the interested content and publications are the information itself structured in a way that its content is easily read. Since expressiveness is out of our scope, the publications are positive numbers varying until a predefined value R. Therefore, subscriptions define a range of interested values, $[v_{min}, v_{max}]$.

The MANET considered in this work is composed of homogeneous nodes which can perform both operations: subscribe and publish. Every subscription/publication is sent through broadcast, hence, every node within its radio range receives the message. Duplicated subscriptions and publications are ignored.

Subscriptions are sent to neighbors in two moments: when they are created and every time a node stops after it had moved. Publications are broadcast in four moments: (i) when they are created, (ii) when a node stops, (iii) when a node has a publication which matches a new incoming subscription, and (iv) when a node receives a publication which matches any stored, still valid, subscription. Therefore, a publication can run through the network in the following steps: first it is broadcast by the publisher itself, then all neighbors check if they have any matching subscription (if they do, they instantly forward it). After that, if any of these neighbors move, as soon as they stop, they broadcast the publication to a different area of the network. In this way, the nodes' movement helps to disseminate a publication to the whole area of the network. In the following subsections, we explain the details of how publications and subscriptions are handled.

3.1 Publication Buffer

An important data structure of PSAMANET is the Publication Buffer, PB. It is a FIFO queue where every received publication is stored. Therefore, whenever a publication is broadcast by a node, all its neighbors store it in their own PB. This stored publications are forwarded (broadcast again) on the four moments mentioned before.

The PB is limited to PB_{size} publications. If a publication is received and the PB is already full, the oldest (first in the queue) publication is removed and then the new one is placed at the end of the queue. Therefore, the PB stores the PB_{size} more recent received publications.

The value of PB_{size} has to be set based on a trade-off between the time a publication still lasts in the network (increases the number of reached nodes) and the number of times it is sent (impacts the number of messages sent). Through some preliminary experiments, it could be observed that to achieve reasonable results in both metrics (coverage and messages sent), PB_{size} should assume higher or lower values, respectively. A solution to this problem was to set a high PB_{size} and a second value which specifies an upper bound to the number of publications forwarded each time, PB_{max}. Therefore, if at any given time, more than PB_{max} publications have to be sent (n, for instance), then PB_{max} publications of the n possibilities are randomly chosen (uniformly) to be be forwarded. In this way, it is possible to increase the time a publication lasts in the network and still limit the number of messages sent.

3.2 Subscription Table

Every node also holds a Subscription Table, ST, which stores the node's subscriptions. There are two kinds of subscriptions: own, from the node itself, and foreign, from neighbors. It is a simple list with an unlimited size. This is not a drawback since subscriptions do not stay for a long time in the ST, as it will be explained later. A subscription is broadcast in two situations: (i) when a node first wants to subscribe for a range of values, a subscription is created, locally stored and then broadcast to all its neighbors, and (ii) every time a node stops, it forwards its own subscriptions.

An important observation is that if subscriptions are never removed from ST, in time every node of the network will be interested in every publication. For this reason, when a node moves, it cleans its ST leaving only its own subscriptions. Furthermore, every foreign subscription is valid only for a defined amount of time S_{ttl}. Every time the ST is used, it first checks for expired subscriptions and remove them. These measures are fundamental in order to adapt PSAMANET to the highly dynamic topology of MANETs. Dropped expired subscriptions may go back to node's ST if any neighbor sends it again. In this case, the subscription will not be considered a duplicated subscription.

Another point about the algorithm is that not every neighbor needs to receive the subscription. If more than one node forwards the same publication it will be a waste of resources (e.g., bandwidth, energy). Therefore, when a node receives any foreign subscription, it has a S_{prob} probability of accepting it or not. Only accepted subscriptions will be stored in the ST and later checked for publications that match.

4. EXPERIMENTS AND RESULTS

PSAMANET was implemented on the NS-2 simulator [14] using the parameters described in Table 1. The network

Parameter	**Value**
`Radio Propagation Model`	FreeSpace
`RXThresh_`	7.69113e-08 (50m)
`Routing`	DumbAgent
`Number of Nodes`	200
`Area`	600m × 606m
`Simulation Time`	3600s

Table 1: NS-2 parameters

Parameter	**Value**
`Speed`	[1.5,4.0]m/s (normally)
`Pause`	[0, 600]s (uniformly)

Table 2: Random Way Point parameters

density is such that a node has an average of four neighbors. This scenario can represent a campus with students moving in different ways (walking, running), each one carrying a handheld. The movement model used was the Random Way Point [11] using the parameters in Table 2.

Related to subscriptions and publications, the experiments considered that each node subscribes to two different ranges (not necessarily disjoint). Those ranges were randomly generated by choosing uniformly a positive number smaller or equal to R, as the minimum value. The maximum value is always the minimum value plus a constant range S_{range}. The S_{range} was set to a value which makes a publication to be of the interest to 20% of the nodes. Therefore, in this experiment setup, S_{range} is set to be $R/400$ because there are 200 nodes and two subscriptions per node. A total of 1000 publications are published during the first third of the simulation (20 minutes). These publications were randomly distributed among the 200 nodes of the network.

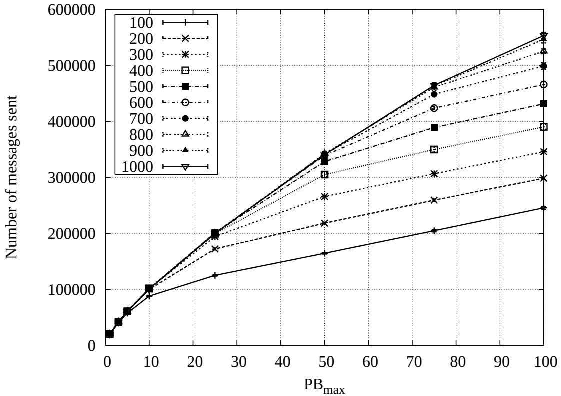
The experiments were divided into two different phases: the first one for tuning the parameters (PB_{size}, PB_{max}, S_{ttl} and S_{prob}) and the second to the performance analysis, as described below.

4.1 Tuning

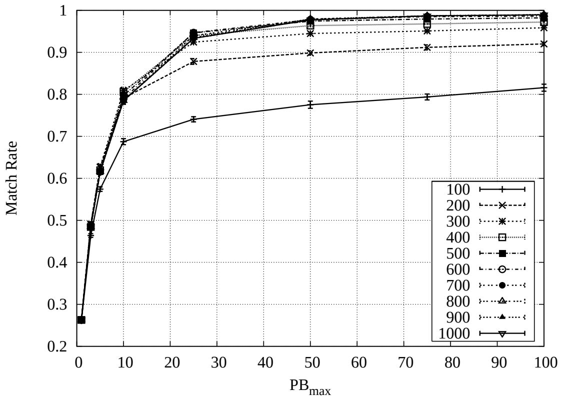
Every tuning experiment ran over 10 different instances of the mentioned scenario. This number was enough to have small confidence intervals using the Student's t-distribution at a 95% confidence level.

Varying all four parameters would lead to a massive number of configurations. Therefore, we chose to first evaluate the parameters related to the Publication Buffer and choose the most suitable parameters to then evaluate the Subscription Table parameters.

As mentioned above, PB_{size} influences how long a publication lasts in the network (the longer it lasts, more interested nodes it reaches), and PB_{max} how many times a publication is sent. Figure 1 shows the results of the tuning experiments with S_{ttl} and S_{prob} set to 300 and 0.4, respectively. The different lines are for different values of PB_{size}. There are 1000 publications during each simulation, therefore, $PB_{size} = 1000$ actually means we used an unlimited buffer. It is interesting to notice that for small values of PB_{max} the match rate for different PB_{size} are almost the same. This happens because even if a publication lasts longer in the PB it has a small possibility of being broadcast in areas different from where they were received. However, as PB_{max} increases, PB_{size} has a greater impact on the match rate. Nevertheless, large PB_{max} values resulted in close match rates for PB_{size} greater then or equal to 500. This means that with a buffer size equals to half of the publications of one hour, the results would be the same as with an unlimited buffer. This is an important observation since large Publication Buffers would incur in spending a large amount of resources, usually scarce on mobile devices. In order to try to evaluate configurations with a small number of messages and others with high match rates, we chose to evaluate the subscriptions parameters with four PB configurations, with (PB_{max}, PB_{size}) equals to: (5, 200), (10, 200), (25, 300), (50, 500).



(a) PB_{max} × Messages Sent for different PB_{size}

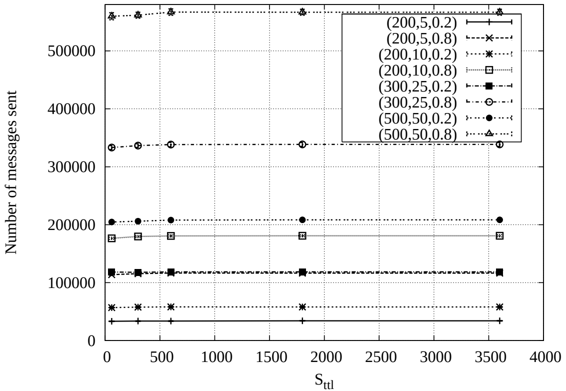


(b) PB_{max} × Match Rate for different PB_{size}

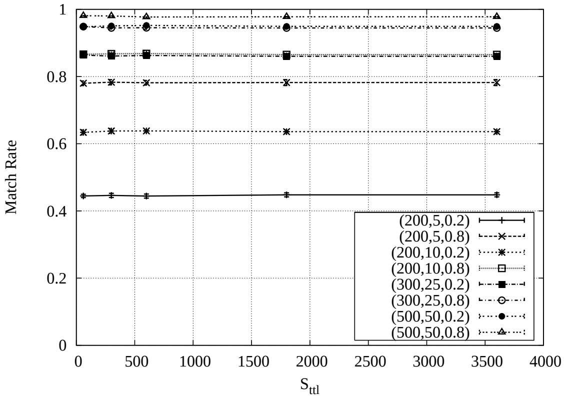
Figure 1: Tuning Experiment for Publication Buffer

Tuning the Subscriptions Table parameters, the first observation was that the S_{ttl} does not have a great impact on the number of messages sent nor on the match rate, as it can be seen in Figure 2. This happens due to the fact that nodes which have moved clear their ST of foreign subscriptions, even before they are expired. Perhaps on more static topologies, the S_{ttl} would have an impact on the architecture performance. We chose to use foreign subscriptions that last for 60s, in order to have a better adaptation on more static topologies where nodes will have many subscriptions expired before they move.

Analyzing further the behavior of the ST, the probability S_{prob} has a great impact in the number of messages sent and in the match rate, as shown in Figure 3. This was

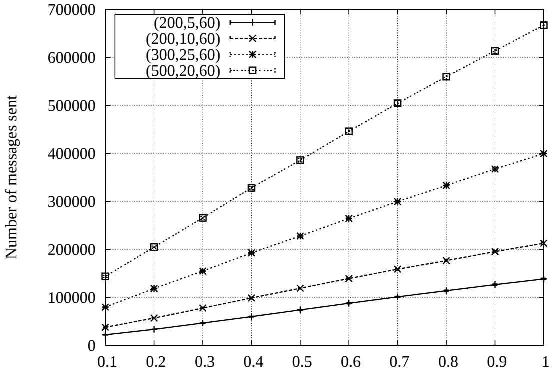


(a) S_{ttl} × Messages Sent for different configurations (PB_{size},PB_{max},S_{prob})

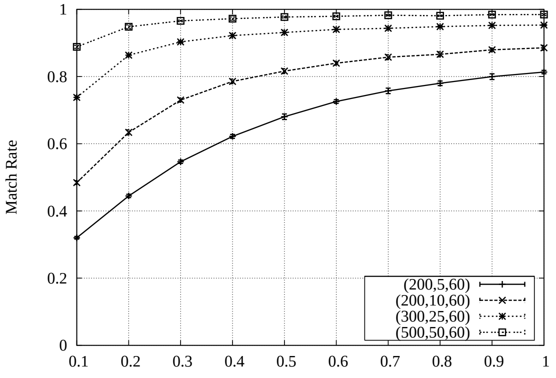


(b) S_{ttl} × Match Rate for different configurations (PB_{size},PB_{max},S_{prob})

Figure 2: Tuning Experiment for S_{ttl} of Subscriptions Table



(a) S_{prob} × Messages Sent for different configurations (PB_{size},PB_{max},S_{ttl})



(b) S_{prob} × Match Rate for different configurations (PB_{size},PB_{max},S_{ttl})

Figure 3: Tuning Experiment S_{prob} of Subscriptions Table

expected since it defines the number of neighbors which are aware of each node's interests. However, even though the number of messages grows linearly, the match rate grows more in a log way, almost converging in some configurations. The results also show that even with high probabilities of accepting subscriptions, the match rate is limited based on the PB configuration. Another observation is that due to the increase patterns of match rate and messages sent, it is possible to reduce the number of messages sent by using smaller values of S_{prob} and still have similar match rates.

4.2 Performance Analysis

According to the observed results, we have chosen to evaluate seven different configurations of the PSAMANET. They are described in Table 3 and all of them use S_{ttl} equals to 60 seconds. As a baseline, we used a gossip-based routing algorithm for ad hoc networks [8]. In this case every subscriber stores (and does not broadcast) its subscriptions. The publisher broadcasts its publications, and whenever a node receives any publication it has a probability of G_{prob} to forward it. When $G_{prob} = 1$, the algorithm behaves as a flooding routing algorithm.

The setup for the second phase of experiments is the same as before but now the experiments are run for 33 different executions. We use this number of runs in order to use the z-distribution to calculate the confidence intervals at a 95% confidence level.

Figure 4 shows the results for the Gossip-based solution. It had a poor result because of the lack of ability to handle the constant changes of the network topology. The large confidence intervals is also an evidence that this solution is highly dependable on the topology, since publications reach only nodes on the same connected component as the publisher. Table 4 shows the results of the seven configurations of PSAMANET. It is clear that PSAMANET is a better solution than the Gossip-based one. The best match rate reached by the Gossiping was of 54% sending 110k messages with $G_{prob} = 1.0$ (Flooding). The configuration `II` had a match rate of about 79% sending 11k less messages. With just 9k more messages than Flooding, we had a match rate of 86%. To achieve a match rate of 95% PSAMANET had to send more than 200k messages. It is important to notice that depending on the application, the proposed architecture can be adapted to attend the service requirements.

Another important observation is that, as opposed to the work in [1], it was possible to have high match rates with only 20% of the network interested for each publication and without the need of exchanging beacon messages to update a neighborhood table.

Configuration	PB_{size}	PB_{max}	S_{prob}
I	200	10	0.1
II	200	10	0.4
III	200	10	0.9
IV	300	25	0.2
V	300	25	0.4
VI	500	50	0.2
VII	500	50	0.4

Table 3: PSAMANET Configurations

Configuration	Messages Sent	Match Rate
I	37,869 ± 234	0.4896 ± 0.0036
II	99,375 ± 632	0.7892 ± 0.0032
III	195,729 ± 991	0.8774 ± 0.0033
IV	119,013 ± 931	0.8659 ± 0.0035
V	193,974 ± 1,181	0.9242 ± 0.0024
VI	207,611 ± 1,508	0.9539 ± 0.0020
VII	330,740 ± 2,137	0.9741 ± 0.0017

Table 4: PSAMANET Results

5. CONCLUSIONS AND FUTURE WORK

This work presents a solution to the problem of developing a Publish/Subscribe Architecture for Mobile Ad hoc Networks (PSAMANET), which properly adapts to the highly dynamic topology of such networks using nodes' movement to disseminate publications. The solution considers a totally asynchronous communication in such a way that end-to-end delays of minutes are not a problem.

The proposed solution is based on the idea that moving nodes can connect distant regions of the network with fewer transmissions. This is done by using a Publication Buffer to store received publications to be later broadcast after a node stops. Node's interest is informed to neighbors, which forward matching publications. Subscriptions are also broadcast when a node stops in a new area of the network, in order to let the new neighbors to be aware of its interests.

The results showed a much better performance than the baseline. For instance, it was possible to reach a 46% higher match rate sending 10% less messages. Besides that, the proposed architecture can be adapted to different QoS requirements achieving match rates above 97%. Therefore, P/S delay-tolerant services can make use of mobility not only to transmit fewer messages but also to achieve higher match rates.

Further investigation of the impact of the movement model and intensity is being currently studied. Another ongoing work is the analysis of PSAMANET behavior as a complex network and observe properties like degree distribution and diameter. As a future work, it is possible to study the P/S paradigm in MANETs for real-time applications and also try to use mobility without incurring in a high delay.

6. REFERENCES

[1] S. Baehni, C. S. Chhabra, and R. Guerraoui. Frugal Event Dissemination in a Mobile Environment. In *Middleware*, pages 205–224, 2005.

[2] L. F. Cabrera, M. B. Jones, and M. Theimer. Herald: Achieving a global event notification service. In *HOTOS '01*, page 87, Washington, DC, USA, 2001. IEEE Computer Society.

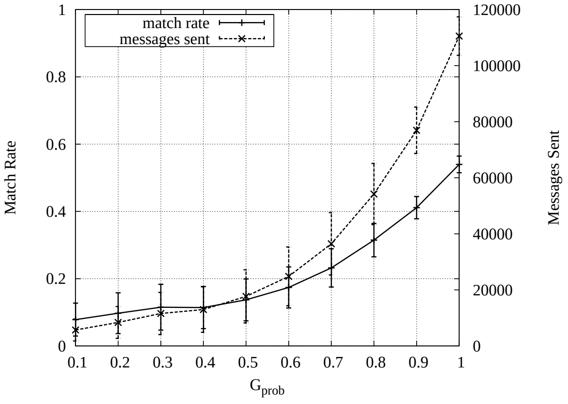


Figure 4: Gossip-based Solution Results

[3] M. Caporuscio, A. Carzaniga, and A. L. Wolf. Design and evaluation of a support service for mobile, wireless publish/subscribe applications. *IEEE Transactions on Software Engineering*, 29(12):1059–1071, 2003.

[4] A. Carzaniga, D. S. Rosenblum, and A. L. Wolf. Design and evaluation of a wide-area event notification service. *ACM Trans. Comput. Syst.*, 19(3):332–383, 2001.

[5] G. Cugola and H.-A. Jacobsen. Using publish/subscribe middleware for mobile systems. *SIGMOBILE Mob. Comput. Commun. Rev.*, 6(4):25–33, 2002.

[6] P. T. Eugster, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec. The many faces of publish/subscribe. *ACM Comput. Surv.*, 35(2):114–131, 2003.

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

[8] Z. J. Haas, J. Y. Halpern, and L. Li. Gossip-based ad hoc routing. *IEEE/ACM Trans. Netw.*, 14(3):479–491, 2006.

[9] Y. Huang and H. Garcia-Molina. Publish/subscribe tree construction in wireless ad-hoc networks. In *MDM '03*, pages 122–140, London, UK, 2003. Springer-Verlag.

[10] Y. Huang and H. Garcia-Molina. Publish/subscribe in a mobile environment. *Wirel. Netw.*, 10(6):643–652, 2004.

[11] D. B. Johnson and D. A. Maltz. Dynamic source routing in ad hoc wireless networks. In Imielinski and Korth, editors, *Mobile Computing*, volume 353. Kluwer Academic Publishers, 1996.

[12] M. Kim and B. Noble. Mobile network estimation. In *MobiCom '01*, pages 298–309, New York, NY, USA, 2001. ACM Press.

[13] R. Narasimhan and D. Cox. Speed Estimation in Wireless Systems Using Wavelets. *IEEE Transactions on Communications*, 47(9):1357–1364, 1999.

[14] ns 2. The network simulator. http://www.isi.edu/nsnam/ns.

[15] A. I. T. Rowstron, A.-M. Kermarrec, M. Castro, and P. Druschel. Scribe: The design of a large-scale event notification infrastructure. In *NGC '01*, pages 30–43, London, UK, 2001. Springer-Verlag.