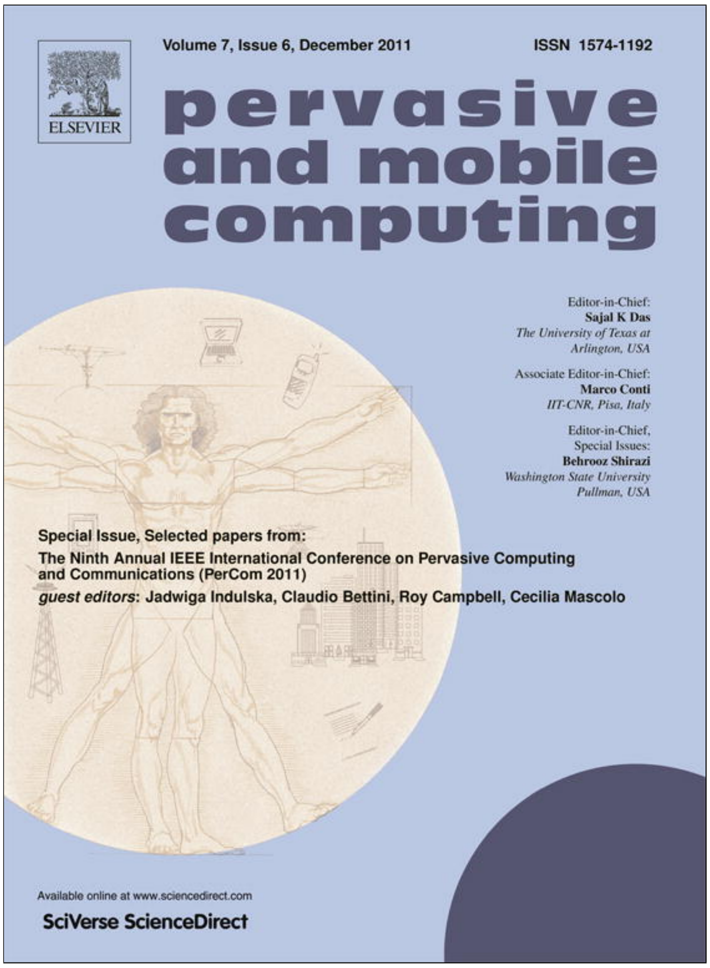


Volume 7, Issue 6, December 2011 ISSN 1574-1192

ELSEVIER

pervasive and mobile computing

Editor-in-Chief:
Sajal K Das
The University of Texas at Arlington, USA

Associate Editor-in-Chief:
Marco Conti
IIT-CNR, Pisa, Italy

Editor-in-Chief,
Special Issues:
Behrooz Shirazi
Washington State University Pullman, USA

Special Issue, Selected papers from:
The Ninth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom 2011)
guest editors: Jadwiga Indulska, Claudio Bettini, Roy Campbell, Cecilia Mascolo

Available online at www.sciencedirect.com
SciVerse ScienceDirect

Pervasive and Mobile Computing 7 (2011) 671–689

Contents lists available at SciVerse ScienceDirect

Pervasive and Mobile Computing

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

Floating content for probabilistic information sharing

Jörg Ott [a,*], Esa Hyytiä [a], Pasi Lassila [a], Jussi Kangasharju [b], Sougata Santra [b]

[a] *Department of Communications and Networking, Aalto University School of Electrical Engineering, Finland*

[b] *Helsinki Institute for Information Technology, University of Helsinki, Finland*

ARTICLE INFO

Article history:
Available online 29 September 2011

Keywords:
Location-based services
Opportunistic networking
Delay-tolerant networking
Geo-tagging
Content sharing

ABSTRACT

People are increasingly using online social networks for maintaining contact with friends and colleagues irrespective of their physical location. While such services are essential to overcome distances, using infrastructure services for location-based services may not be desirable. In contrast, we design and analyze a fully distributed variant of an ephemeral content sharing service, solely dependent on the mobile devices in the vicinity using principles of opportunistic networking. The result is a *best effort* service for *floating content* in which content is created locally, its availability is geographically limited and its lifetime and spreading depends on interested nodes being available. In this paper, we present our system design, algorithms, and protocol specification in detail. A set of real world experiments is then used to assess the achievable transmission rates and transmission ranges in such a system. We validate our previous analytical results and assess the performance of floating content in general and especially of different replication and deletion strategies by means of extensive simulations using a map-based mobility model in downtown Helsinki.

1. Introduction

Online social network applications for sharing content, opinions, and experiences (Facebook, Flickr, YouTube, Twitter, etc.) are widespread means for people to foster their relationships irrespective of physical distance. Increasing mobile Internet use has made sharing experiences from mobile devices popular, as the authors witness daily from their acquaintances, e.g., on Facebook. Context- and location-aware services, such as digital graffiti [1], and (to some extent) Google Maps and Google Earth, have been constructed around mobile users, however, relying on services in the network infrastructure: to store and maintain data as well as to determine geographic proximity. While network-based social applications are essential to overcome distances and connect people around the world, relying on infrastructure services for location-aware services may often not be desirable:

- *Location privacy* concerns arise since the location needs to be communicated at least at some level of accuracy to obtain the right context information.
- *Content privacy* issues occur since shared information will be available at some "central" location and thus easily subject to censorship (a mixed blessing).
- *Connectivity* to the infrastructure becomes a prerequisite that is often limiting, especially for traveling users who may face high roaming charges, unavailability of data services, or simply no network coverage.
- *Geographic validity*: locally relevant content may be of little concern to the rest of the world; storing it in some well-accessible location may not be of much use.

[*] Corresponding author. Tel.: +358 9 4702 2460; fax: +358 9 4702 2474.
E-mail address: jo@netlab.tkk.fi (J. Ott).

1574-1192/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.pmcj.2011.09.001

- *Temporal validity*: Information may only be valid for a limited amount of time; yet, centrally stored content rarely is tagged using expiry information, leading to the content never being deleted—yet, quite frequently never being read either (WORN, write-once, read never).
- *Large pieces of content* may overload wireless access links unnecessarily, in particular if the content is unlikely to be sought after globally.
- *User identification* of some kind is usually applied to limit the amount of data posted and creates some sense of responsibility towards the service provider.

In this paper, we propose and analyze a fully distributed variant of an ephemeral content sharing service, solely dependent on the mobile devices in the vicinity using principles of opportunistic networking, built upon our earlier reported work-in-progress [2] and extending [3] to explore more diverse parameter settings and perform a more thorough evaluation of floating content. Specifically:

- We perform extensive analysis of the buffer zone issue of floating content to determine how content replication and deletion should be performed. We consider several replication and deletion functions and determine that choice of a deletion function has more impact on overall performance than the replication function.
- We perform real-world measurements of WLAN throughput in urban settings to get accurate estimates of network parameters for our simulations.
- We also sketch a programming model and application examples for floating content.

The results strongly concur with our earlier findings and affirm the technical feasibility of floating content.

With the floating content concept, any user may create content and define its geographic *origin*, and *validity radius* – that together define the *anchor zone* in which content is meaningful – plus an *expiration time*.[1] The creator's device starts disseminating the content to its neighbors within the validity radius, as do other nodes, reducing the replication and increasing the deletion probability as a function of the Euclidean distance from the origin. As long as there are enough supportive mobile devices around in the anchor zone to replicate and store a piece of content, it *floats*. Other mobile nodes interested in this piece of content – because of its location or because its metadata matches what they are looking for – will be able to obtain a copy when they get ''in range'', i.e., enter its anchor zone, and have a copy disseminated to them.[2] When the node density becomes too low (even temporarily), the content will disappear (it *sinks*).

The net result is a *best effort* service for *floating content* in which: (1) information dissemination is geographically limited; (2) the lifetime and spreading of information depends on interested nodes being available in the anchor zone of a particular content item; (3) traffic can only be created and caused locally, thus limiting DoS effects; and (4) content can only be added, not deleted, so that security issues are kept outside the system.[3] We expect that such simple mechanisms allow creating more sophisticated services on top.

2. Related work

Concepts similar to our floating content have been introduced in research already a few years ago, such as in [1,4,6–14]. Some, e.g., [4,8,12], focus on distributing content over an ad-hoc or DTN-like network, using the wireless network only as a kind of a cache for Internet content. They do not consider the case of managing content *purely in the wireless domain*. Others, e.g., [9,10], consider a purely wireless case, but focus on directed communications between users. In contrast, our work is not about directed communications, but about content floating in a particular location, i.e., the applications are different.

Work in [6,11] presents models for hovering information, which bears similarities with our floating content model. However, our work differs from theirs in an important way. They focus on defining the general model, whereas our focus is on *evaluating the general feasibility* of floating content systems. By feasibility, we understand aspects such as node density, required communication range, etc. Our work explores the parameter space of floating content systems to identify under which conditions such systems are feasible.

Work in a recent publication [13] presents a system concept very similar to our work in [2]. Their focus is mainly on implementation aspects and protocols and hence complementary to the work in this paper with a focus on evaluating the operating conditions for floating content systems. Moreover, a similar scheme has been considered in the context of VANETs, e.g. in [15,16]. The former studies the storage capability of such a system in terms of mean time to information loss. Several different scenarios have been considered by means of simulation experiments ranging from a one-way traffic setting to a city-wide scenario based on real vehicle traffic traces in San Francisco (SF Cab). The latter has focused on estimating the node density and dynamically adapting the number of carriers for a piece of content within a given geographic area. Location-based storage and retrieval of events was also explored using (fixed) infrastructure nodes, e.g., by creating a well-connected peer-to-peer overlay that takes locality into account when choosing nodes to store information [17].

[1] Moreover, content may be tagged with metadata (e.g., content channels [4]) to support filtering and selective forwarding and replication.

[2] They could also obtain a copy by means of one-hop queries to their neighbors, but we defer this to future study.

[3] We expect that the validity of local information be checked at low cost, e.g., in the real world by validating that there is really free beer in a particular bar and posting annotations in case a ''story'' is not true. In addition, within groups content postings may be signed leveraging the users' identities in other domains (e.g., of SIM cards [5]).

Another very recent paper [14] presents a system called Locus. Conceptually, the work only reiterates already published concepts on floating content [2,6,11,13,18,19]. Locus allows remote nodes to query for information placed in a given anchor zone, which makes it different from related work in this area. However, we believe this property is actually a severe weakness, since it allows for remote nodes to perform denial-of-service attacks anywhere in the network. In pure floating content work [2,6,11,13,18,19] such attacks require physical presence in the targeted anchor zone, making them far less vulnerable, if not even impervious, to these kinds of attacks. This paper performs an extensive evaluation of floating content in diverse urban settings and taking more key parameters into account.

The floating content concept is an example of delay tolerant networks. The fundamental properties of routing in such networks have been recently analyzed, see [20] and the references therein. However, the floating content service requires modeling the notion of information availability in the anchor zone. We have developed analytical models to this end in [19], and validated the result of the so-called non-spatial model in a more complex mobility scenario [3].

3. Floating content design

We assume that all users are mobile nodes and that there is no supporting infrastructure for the system.[4] They use mobile phones or similar devices to communicate, so that ample storage capacity is available of which a sufficient fraction is set aside for floating content. The devices use wireless interfaces (e.g., Bluetooth or WLAN) for ad-hoc communication. Real-world traces have shown that median contact durations between mobile users, e.g., range from 15 s or less as in RollerNet [21] to some 120 s in conference environments [22].[5] Experiments using Bluetooth for inter-device communication yielded some 10–15 s channel setup delay and a net data rate of some 50 KB/s [22]. Thus, exchanging data up to several megabytes per contact appears feasible so that floating modestly sized pieces of content could be supported (from text messages to photos); it also suggests that the communication capacity during a contact is likely to become the system bottleneck and mobile nodes may not need to reserve a lot of storage capacity for floating content. We carried out experiments using ad-hoc WLAN between different mobile devices to determine reasonable simulation parameters and report them in Section 5.

The devices also need to be location-aware, e.g., by using GPS, triangulation-based methods using WLAN access points or cellular base stations, or any other method offering reasonable accuracy; suitable location APIs are commonly available in modern smart phones. Since the floating content system is probabilistic, there are no strict limits on the accuracy of positioning techniques; nodes are only required to agree on basic measurement parameters and the overall operation to determine the extent of anchor zones. Nodes need roughly synchronized clocks to time out content items; both GPS and cellular networks may provide local time.

Mobile users originate ("post") content items with a defined anchor zone and TTL and must reside in the anchor zone when posting. Other users are interested in these items[6] and accept copies to store and (probabilistically) further replicate within the anchor zone. We explicitly allow information items to disappear from the system and provide no guarantees about their availability. If no (or too few) nodes are around to replicate a content item and the creator leaves the anchor zone, items may disappear. Once the lifetime of an item expires, it will be deleted by all nodes.

3.1. System operation

A node generates an information item I of size $s(I)$ with a certain lifetime (TTL) and assigns an anchor zone defined by its center P and two radii, r and a, as shown in Fig. 1: r defines the *replication range* within which nodes always try to replicate the item to other nodes they encounter; a defines the *availability range* within which the content item is still kept around with limited probability, while outside a no copies of the item are to be found.

In the simple case of $r = a$, if two nodes A and B meet inside the anchor zone of an item I, and A has I while B does not, then A replicates item I to B. Since replication is based purely on the location of nodes, every node in the anchor zone should have a copy of the item. Nodes leaving the anchor zone are free to delete their copy of the item.

In practice, the replication and deletion works as follows. Consider node A having item I, with an anchor zone defined by center point P and radii r and a. Let h denote the distance of node A from point P. When node A meets another node B, A will replicate item I to B with probability $p_r(h)$:

$$p_r(h) = \begin{cases} 1 & \text{if } h \le r \\ R(h) & \text{if } r < h \le a \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where $R(h) \in [0, 1]$ is some (decreasing) function that gives the probability of replication outside the replication range but within the availability range. We define the deletion probability $p_d(h)$ in a similar way, $D(h) \in [0, 1]$:

$$p_d(h) = \begin{cases} 0 & \text{if } h \le r \\ D(h) & \text{if } r < h \le a \\ 1 & \text{otherwise.} \end{cases} \tag{2}$$

[4] Content dissemination from fixed access points would lead us towards the PodNet model [4] paired with geographically limited distribution.

[5] See, e.g., the CRAWDAD archive for a collection of contact and mobility traces at http://crawdad.cs.dartmouth.edu/.

[6] The content can be organized into groups or channels similar to [4], but we restrict our considerations in this paper to a single channel.

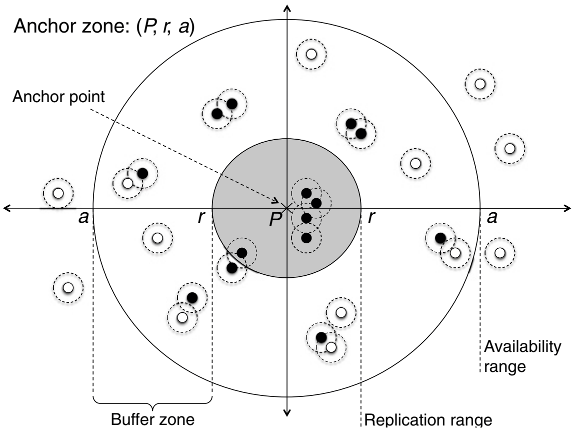


Fig. 1. An anchor zone of an item, mobile nodes and their communication ranges: the content item gets replicated across and deleted from nodes as a function of the distance from the anchor point. The probability of a node carrying an item (black nodes) tends to 1 inside the anchor zone's replication range r and decreases until, after an availability threshold a, no more copies are found.

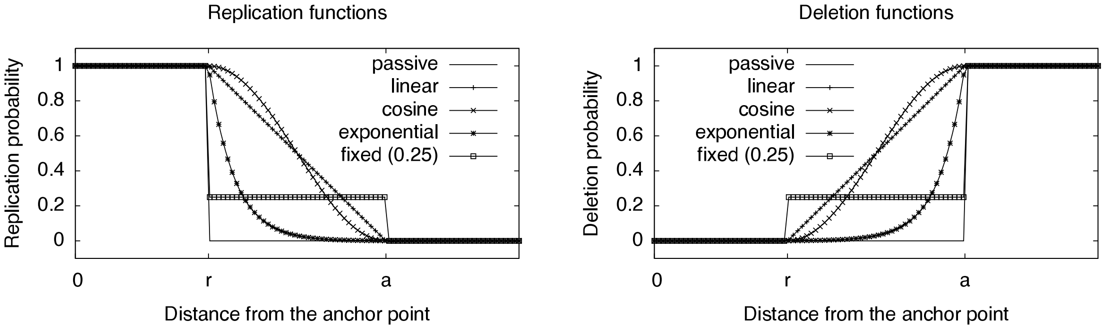


Fig. 2. Different replication (left) and deletion functions (right).

The deletion function serves the purpose of early prioritization to prevent buffers from filling up; it is evaluated upon each encounter with another node. If there is still a need to free buffer space the oldest messages are discarded.

These definitions yield the area outside the replication but within the availability range of the anchor zone as a *buffer zone* that offers smooth degradation of availability over distance. This definition of an anchor zone with two ranges is beneficial, because it provides protection against items disappearing when nodes move outside the replication range of the anchor zone for a brief moment and then return. In essence, the buffer zone provides a degree of hysteresis in item replication and deletion, analogous to control systems. Beyond the availability distance a (see Fig. 1) copies are deleted (immediately or upon encountering the next node, depending on the implementation).

3.2. Buffer zone

As noted above, the buffer zone (r, a) serves the purpose of allowing for a gradual reduction of information item availability on mobile nodes with increasing h and avoiding oscillation. The buffer zone is characterized by replication and deletion functions, $R(h)$ and $D(h)$. In a simple case, the buffer zone is entirely passive, i.e., content items are neither replicated nor deleted, i.e., $R(h) = D(h) = 0$ (*none*). In addition to the passive operation, we define three dynamic functions – *linear*, *exponential (exp.)*, and *cosine* – as well as one static – *fixed* – behavior, with C_r and C_d being the replication and deletion constants, respectively. The resulting functions are summarized in Table 1. The functions are illustrated in Fig. 2 with $C_e = 7$ as this yields $R(a) < 10^{-3}$ close to zero, yet still leaving chances for replication/deletion across the buffer zone. Note that setting $C_r = C_d = 0$ yields the passive buffer zone. If $r = a$, $R(h)$ and $D(h)$ become immaterial; this leads to a hard borderline between deletion and replication.

Fig. 3 depicts a content item with its creation at its anchor point and the anchor zone radii r and a for linear replication and deletion functions. It also shows all the locations at which copies of the message were replicated and deleted. Note that the deletion points also include those points where the copies of the message were discarded when its lifetime expired; therefore, those points can also be found inside the replication range. Deletion locations well outside the anchor zone can be found because the message deletion only occurred upon timeout or when another node was met (*upon-encounter* deletion, see Section 3.3). This example shows that 12 copies of a content item were available inside the replication range when its lifetime expired.

Table 1
Replication and deletion functions for $h \in (r, a)$.

Type	Replication function	Deletion function
None	$R(h) = 0$	$D(h) = 0$
Fixed	$R(h) = C_r$	$D(h) = C_d$
Linear	$R(h) = 1 - \frac{h-r}{a-r}$	$D(h) = \frac{h-r}{a-r} - 1$
Exp.	$R(h) = e^{-C_e \frac{h-r}{a-r}}$	$D(h) = e^{-C_e\left(1-\frac{h-r}{a-r}\right)}$
Cosine	$R(h) = \frac{1}{2}\left(\cos\left(\pi\frac{h-r}{a-r}\right) + 1\right)$	$D(h) = -\frac{1}{2}\left(\cos\left(\pi\frac{h-r}{a-r}\right) - 1\right)$

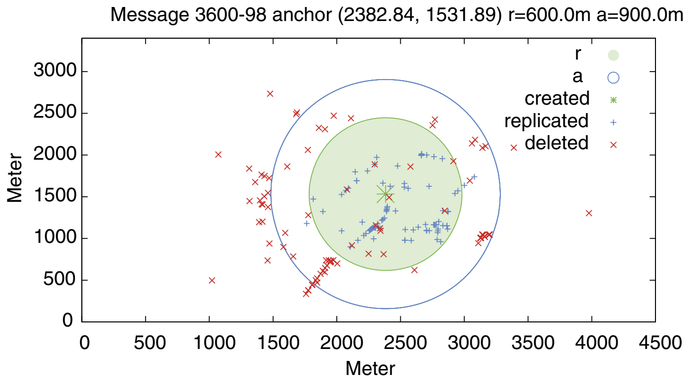


Fig. 3. Illustration of the buffer zone showing the anchor point (creation) of a content item, its replication (r) and availability range (a) as well as the locations where the item was replicated and deleted, assuming linear replication and deletion functions.

Our simulations in this paper explore two different sets of scenarios: (1) the hard border without buffer zone, i.e., $r = a$, for studying the basic feasibility of floating content, and (2) the impact of the buffer zone with all the different behaviors defined above (covering all permutations) to understand their impact on floating performance.

3.3. Floating content protocol

A floating content message m is identified by a unique message id Id_m and carries its anchor zone spec, i.e. anchor point (P_m, r_m, a_m); and its lifetime T_m in its headers; the content item I_m of size $s(I_m)$ is in the message body. We use a 4-phase protocol for exchanging floating content messages:

(1) *Neighbor discovery*: Nodes continuously send neighbor discovery beacons via IP multicast to discover peers within radio range.
(2) *Offering content*: After discovering a peer, i.e. receiving a beacon, a node sends a summary message that includes a vector of the content items available for replication, i.e., a list of all items for which $p_r(h) > 0$ at the node's present location. Per item, the vector contains the item's identification Id_m and its size $s(I_m)$. The summary may be limited to what fits into one MTU size packet (or otherwise to a message sufficiently small to be exchanged efficiently so that most of the per-contact communication capacity is left to exchange content); if a node has more content to share, the list of content is spread across multiple summary messages in a round-robin fashion.

Requesting content: As soon as a device is aware of what a neighbor has to offer, it requests all or a subset of the items for those items for which evaluating $p_r(h)$ so suggests. It prioritizes the order in which elements should be replicated according to the replication policy. This policy determines in which order messages are replicated when two nodes meet. We define seven such policies: FIFO preserves the order in which messages were created/received at a node, RND randomizes the order, and three further algorithms use ascending order by anchor zone size ($A_m = a_m$), floating volume ($V_m = A_m \times s(I_m)$), and the product of volume and TTL ($V_m \times T_m$); we abbreviate these algorithms as *Smallest Area First (SAF)*, *Smallest Volume First (SVF)*, and *Smallest Total resource consumption First (STF)*. SAF, SVF, and STF consider the impact of the anchor zone size linearly; however, the covered area grows to the square. We therefore also define SVF2 ($V'_m = A_m^2 \times s(I_m)$) and STF2 ($V'_m \times T_m$) to take this into account.[7]

Content transfer: Messages containing the requested items are then exchanged until the batch is completed or nodes lose contact. In the former case, the receiver discards incomplete messages; in the latter, the protocol returns to phase (2).

All phases are fully bidirectional so that message exchanges take place in both directions simultaneously. The protocol does not restrict message exchange to two nodes at a time (even though some link layer technologies may). Beaconing is continued throughout the message exchange process so that new nodes may be discovered while a node is already

[7] There is no need for SAF2 because SAF has the anchor zone size as the only parameter.

exchanging messages with another. The protocol phases and the incremental summary messages allow nodes to append messages they just received to the list of those offered to other nodes while already exchanging messages.

As an independent operation, nodes delete messages following one of two deletion policies when $p_d(h)$ suggests deleting a message m. Recall that, with $D(h) = 0$, no messages are deleted within a. Outside a, *immediate* deletion leads to m being discarded once a node leaves a, whereas *upon-encounter* deletion performs this action only when the next node is met, allowing a node to wander out of a and back in while keeping m. From an implementation perspective, the *upon-encounter* deletion policy is more sensible as it is triggered by an external event. Deletion takes place before messages are offered to other nodes.

While our protocol specification operates as described above, we make two simplifications for the simulations described in Section 6: (1) We let the simulator determine instantaneously when two nodes are in radio range and thus do not need any explicit beaconing; this also implies that there is no delay (in the order of a beaconing interval) before recognizing available peers. (2) We do not explicitly model the exchange of summary messages in the simulations because these are tiny compared to the floating content messages; their additional transmission delay would be marginal (also compared to the beaconing interval).

3.4. Operational considerations

We allow the user generating a content item to define the extent of the anchor zone. We only require the user to be in the anchor zone at the time of creation, but do not impose any limits on the anchor zone size. This may naturally cause problems because there is no incentive for users to limit the anchor zone. It would thus be easy to spam the system by inserting items with "infinite" anchor zones, limited in practice only by numeric representation of the radius in the protocol messages. But flooding items throughout the network is problematic, since it is known that flooding quickly exhausts the system resources (both channel and buffer capacity).

Without infrastructure and with potentially sparse and frequently disconnected node populations, accounting or reputation mechanisms like [23] do not appear feasible, e.g., due to their vulnerability against the Sybil attack [24] because of the lack of central authentication of identities. We therefore define a simple mechanism for resource management to discourage unlimited content distribution: at any given point, we prioritize items inversely with respect to (a) their expected resource consumption and (b) the distance from their anchor. We realize (a) by means of the five resource-aware replication policies SAF, SVF, SVF2, STF, and STF2 that take into account the anchor zone size A_m, $A_m \times s(I_m)$, $A_m^2 \times s(I_m)$, $A_m \times s(I_m) \times T_m$, and $A_m^2 \times s(I_m) \times T_m$, respectively, that we evaluate below. We realize (b) by means of five different functions for $p_r(h)$ and $p_d(h)$.

When a node needs to replicate or store more items than it is able to, it prefers items with the smallest anchor zones and evicts the items with the largest anchor zones. It is possible to post items with large anchor zones, but their availability is likely to be quite low whenever there is congestion.

It is obviously possible for a spammer to move and create items with small anchor zones everywhere. We provide no mechanisms to guard against this. Instead, we consider the effort of having to move around to be a sufficient deterrent to most spammers. Moreover, making replication priority a function of the content lifetime (as in the STF policy) ensures that content will not stay around infinitely so that a spammer would have to revisit each place regularly.

Our approach has the advantage that it does not require security infrastructure or any degree of mutual trust. Of course, nodes could spam their immediate neighborhood or perform DoS attacks, but they could achieve the latter using a physical layer interference anyway.

4. Analytical model

The floating content service has been studied analytically in [19]. A fundamental objective of this work has been to derive the so-called *criticality condition* guaranteeing the availability of the information within the anchor zone with a high probability. This depends heavily on the mobility patterns of the users, especially inside the anchor zone. Next we present the so-called non-spatial model from [19].

4.1. Non-spatial black-box model

Consider the anchor zone just as a locale where the nodes enter and then spend some time and finally exit. Inside the locale, the nodes are exchanging information between each other. We assume that nodes move randomly within the locale and the time spent there is relatively long so that the exact points of entrance and exit have no bearing. The population is assumed "well mixed" so that the proportion of information-carrying nodes is constant everywhere.

During the sojourn time, a node randomly encounters other nodes. Consider a system consisting solely of two nodes staying permanently in the locale and denote by ν the frequency at which they come in contact with each other, i.e., within each others' transmission ranges. Now, if the total population of nodes in the locale is N, then there are $(1/2)N(N-1) \approx (1/2)N^2$ pairs and the total rate of encounters is $(1/2)N^2\nu$. The fraction $2p(1-p)$ of these encounters are such that a node without the information gets it, and the total rate of such events is $p(1-p)N^2\nu$. This is the rate at which the size of the population of information-carrying nodes tends to increase. Let $1/\mu$ denote the node's mean sojourn time in the anchor zone. Then the total exit rate of nodes is $N\mu$, and the exit rate of information-carrying nodes is $Np\mu$, so that their net growth rate is

$$N\frac{d}{dt}p = N^2p(1-p)\nu - Np\mu. \tag{3}$$

In equilibrium, the two terms on the right hand side are equal leading to the stationary value. Existence of a positive solution $p^* > 0$ requires that,

$$N\nu/\mu > 1. \tag{4}$$

This is the *criticality condition*. Note that the quantity on the left hand side is the *average number of encounters a randomly chosen node experiences during its sojourn time*. By considering the sign of the derivative (3) we see that the solution is stable. Below criticality, the derivative is negative and the solution is driven to $p = 0$, i.e., the information cannot be sustained even in the black-box model.

4.2. *Applicability of the model*

The black-box model for information exchange is a highly abstract model capturing only the essential elements. The crucial assumptions are (1) the fluid limit approximation and (2) a well-mixed mobility pattern. The fluid limit approximation means that the anchor zone is populated with a large number of nodes. In this case, the criticality condition ensures that information remains available. In practice, the number of information-carrying nodes, *Np*, can be small and the information will be eventually lost due to stochastic fluctuations in the system. The well mixed mobility pattern assumption implies that all the nodes inside the anchor zone, with and without the information, are equally likely to encounter each other at any given point of time and independently of any past encounters.

In the black-box model, we have chosen to ignore the spatial aspects of the information exchange process. For example, in practice one can expect that the probability of a node carrying the information is smaller near the boundary than at the center of the anchor zone. We also note that several key parameters are not explicitly present in (3). For example, the transmission range *d* affects directly the encounter rate ν, i.e., $\nu = \nu(d)$. Analysis of a more detailed model capturing the spatial aspects of the information exchange is challenging, and can only be done for some elementary models. For an example, see [19].

In this work, we apply the above black-box model, as it already captures the fundamental elements of the system, and characterizes the criticality condition, above which the information remains available in the anchor zone with a high probability. In particular, we estimate the three key parameters in (4) from the simulation experiments, and then compare how the criticality quantity $N\nu/\mu$ correlates with the observed average information life time. The similarity turns out to be very good, confirming the applicability of the theoretical model in these more complex mobility scenarios.

5. Network transmission capacity

In order to get a clearer picture about how much data can be transmitted during a contact in a floating content scenario – to gain an initial understanding of the feasibility of floating content in practice and to properly parameterize our simulation scenario – we performed measurements with real 802.11 devices and measured both the effective transmission ranges and throughputs. Similar measurements were already carried out in the past, see e.g., [22,25,26], but some of the results are already well over 5 years old and the underlying technology has evolved significantly, for example through the proliferation of 802.11g/n devices. Comparing our results to previous work, in particular [25] which reports on similar experiments, we can see that both the effective transmission ranges and throughputs have increased significantly since 2004 when those results were published. Our results roughly agree with those in [22] in terms of radio range but exceed their reported transmission capacity. We present our initial measurement methodology and results below and draw conclusions for our simulation setup. Because of the significant evolution in quantitative results, we believe that a more thorough study is warranted.

5.1. *Phase 1: minimal interference*

We divided the experiments into two phases. In the first phase we performed measurements in controlled environments, as interference-free as possible. In the second phase described in the next subsection, we performed measurements in normal urban settings like shopping malls and bus terminals, to get a better picture of real-world parameters.

The first phase covers both outdoor and indoor environments. We performed our outdoor experiments in a football field, on the outskirts of Helsinki. We scanned the area for network interference and, contrary to our expectations, there was some network interference along the edges of the field. However, the signal strength of such interference was minor and we decided to ignore it in our analysis. We did our indoor experiments inside a long corridor inside the university. There were several other wireless networks operating in the building, but we carried out our tests at night, when interfering traffic was minimal to non-existent.

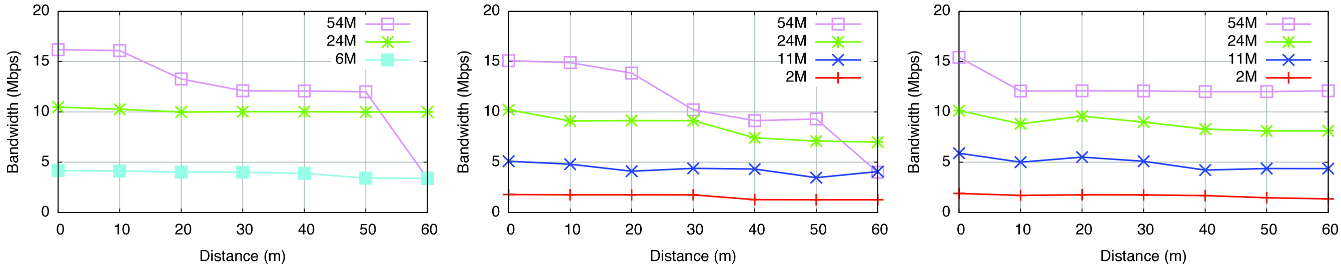


Fig. 4. Throughput of 802.11g as a function of transmission distance, RTS/CTS off. From left to right, TCP outdoors, TCP indoors, UDP indoors.

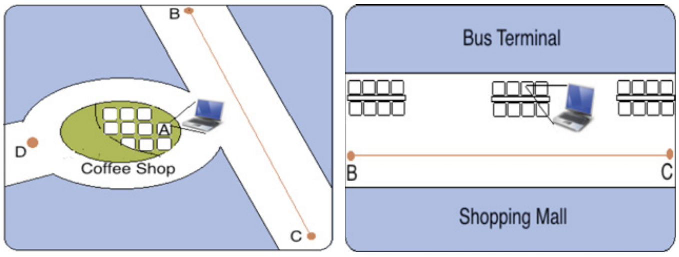


Fig. 5. Measurement locations in phase 2. Location "coffee shop" (left), "bus terminal" (right).

The measurements were carried out on channel 1, with transmission power set at 100 mW with power savings off. Two laptops were used to carry out the experiments, a Macbook with a Broadcom wireless card, supporting 802.11a/b/g/n and an Asus netbook with an Atheros card supporting 802.11b/g. We used 802.11g in our measurements. Both systems were running Ubuntu 2.6.32-24-generic. We used iperf to generate traffic and collect the throughput data. We measured performance both with RTS/CTS on and off, but report results only for the case of RTS/CTS off. With RTS/CTS turned on, the throughput was generally slightly lower for UDP and roughly one half for TCP compared to the throughput with RTS/CTS off.

The results are shown in Fig. 4. Each graph shows the throughput for different nominal data rates as a function of the transmission distance. In each experiment, the server was fixed at a particular location, while the client was moved 10 m after each reading. Each step was repeated several times, and the results shown are the average readings at each location. As can be seen, lower data rates have rather long, stable transmission ranges, whereas higher data rates tend to drop off sharply after a certain distance. Outdoor throughputs are slightly higher than indoor throughputs and we attribute this difference to the interference of other wireless networks in the building, as well as the more complex fading environment for the electromagnetic signals. Indoor graphs in particular for UDP do not show a drop in throughput, but in the outdoor experiments this was observed to happen after around 80 m (not shown).

The results show that under "ideal" conditions, transmission ranges of up to 80–100 m with a stable throughput are possible. Higher data rates start to fall off around 30–50 m for TCP. Actual throughputs are around 30%–50% of the nominal data rates, which agrees with the results for 802.11b presented in [25]. However, our effective transmission ranges are much longer than those reported in [25]. We attribute this to the difference between 802.11b, used in [25], and 802.11g that we used, which agrees with the findings in [22].

5.2. Phase 2: lively urban places

In the second phase, we focused on real-world scenarios. We used a mixture of devices, laptops, netbooks, and Nokia N800 Internet tablets and performed the measurements in crowded areas, like shopping malls and bus terminals. All locations had a lot of people around and several active wireless networks causing interference.

Fig. 5 depicts two of the locations we used. Location 1 is a coffee shop. The server was placed at point A, points B and C are two ends of a path in direct line of sight with the server. The distance between A and B is roughly equal to distance between A and C and is approximately 40 m. Point D is behind the coffee shop and has no direct line of sight to the server. The distance between A and D is approximately 30 m. Location 2 is an indoor Bus Terminal with a wide corridor separating the platforms from a shopping mall. The server was placed at point A. Points B and C are two ends of a path, in direct line of sight with the server. The distance between A and B is roughly equal to distance A to C and is approximately 45 m. We had selected these locations because of the large number of people present and the large amount of interfering wireless networks (more than 10 were observed at some times).

We used four Nokia N800 Internet tablets with Intersil Prism54 chip sets, p54 drivers, 802.11b, running OS 2007. We used a custom-built software – *tcpx* [27], a TCP client–server pair for configurable load generation and fine-grained performance tracking – for measuring TCP throughput. These measurements were performed at the "Bus Terminal" location. The second set of experiments was performed using the laptops used in phase one (using 802.11g). The laptop experiments were carried out in both locations.

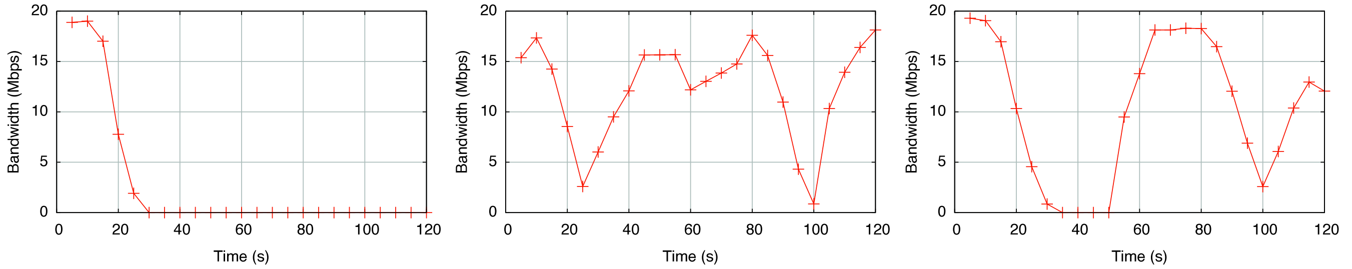


Fig. 6. Throughput as a function of time in phase 2. From left to right scenario 1, 2, and 3, respectively. See text for explanation of movement patterns.

We experimented with many different scenarios, placement of devices, and user mobility and report three representative cases in this paper[8]:

- *Scenario* 1: Location 1 and laptops, one node fixed at point A, another moves along AD and beyond.
- *Scenario* 2: Location 1 and laptops, one node fixed at point A, another moves A–B–C–A always within the line of sight without any physical obstacle except the people moving along the path.
- *Scenario* 3: Location 2 and laptops, one node fixed at point A, another moves A–B–C–A always within the line of sight without any physical obstacle except the people moving along the path.

Fig. 6 shows the resulting throughput for these three cases. Note that we do not show results for Nokia N800 here because they were limited to 802.11b and we were only able to get throughputs of up to 1.13 Mbps; however, they exhibited similar behavior in terms of transmission range, with reliable range being just a bit shorter.

Fig. 6 shows the three scenarios. In scenario 1 on the left, one node is fixed at point A and another moves from A, walks around the coffee shop to point D and continues beyond D. Point D is reached at around 20 s, marking a sharp drop in throughput, indicating that the devices are beyond communication range (recall that the distance A–D was about 30 m, without line of sight). In scenario 2, shown in the middle, one node is fixed at point A, another moves A–B–C–A. Point B is reached at around 20 s and point C at around 100 s, both visible as significant drops in throughput. Although connection was maintained throughout the whole experiment, the distance of 40 m between A and B and A and C is likely to be near the maximum usable transmission range. In scenario 3, shown on the right, the movement pattern was again A–B–C–A, but the location was the bus terminal. Again, we see similar behavior to scenario 2, with the exception that the connection was lost near point B (around 30–45 s). Recall that the distance A–B was approximately 45 m, again indicating that a range of about 40 m seems to be the maximum usable range in crowded indoor areas with good line of sight visibility.

Comparing to phase 1 results with an unpopulated and relatively interference-free indoor environment, we see a clear drop of effective communication range, from well over 60 m (length of the corridor; true range likely a bit higher as shown by the outdoor experiments) to around 40 m (similar to [22]). Walking speeds does not seem to affect the achievable data rate. This becomes apparent in the middle and right graphs of Fig. 6. Because of the movement pattern, between 40 and 80 s the two devices are relatively close to each other and we observe throughputs similar to the static cases over short distances shown in Fig. 4. Our findings concur with previous results, in particular [25], and extend them to 802.11g, confirming the brief results presented in [22].

As a practical take-home message for floating content, we can conclude that effective transmission ranges in realistic settings are likely to be below 40 m at high data rates, possibly less depending on line of sight. However, when connection can be established, we can expect quite good throughput numbers, 15–20 Mbps for 54 Mbps nominal data rates, even in environments with a lot of interference from networks and people walking around. Lower data rates will yield higher reachability ranges and outdoor operation yields a farther reach than indoors. Assuming a walking speed of 1 m/s, two people meeting head-on would remain in contact for about 30 s and would be able to transfer on the order of *several tens of megabytes* of data. This is clearly enough for even short videos, allowing floating to support a wide range of different types of content.

For our simulation setup – that models outdoor movement – we decide to be conservative and choose just 2 Mbit/s effective throughput data rate as is easily achieved across all our scenarios and is well within the rate range of 800 kbit/s–4 Mbit/s reported in [22]. For the radio range, we choose 50 m to reflect the outdoor nature of our setup and yet accounting for the many obstacles that would be found in the real world. For reference, we also include simulations with only 10 m radio range.

6. Simulation environment

While our initial evaluations [2,19] focused on basic feasibility in a static and two idealized mobility settings (using random-waypoint and Manhattan mobility), for this paper, we choose the more sophisticated Helsinki City Scenario

[8] Those not reported yielded results leading to similar conclusions.

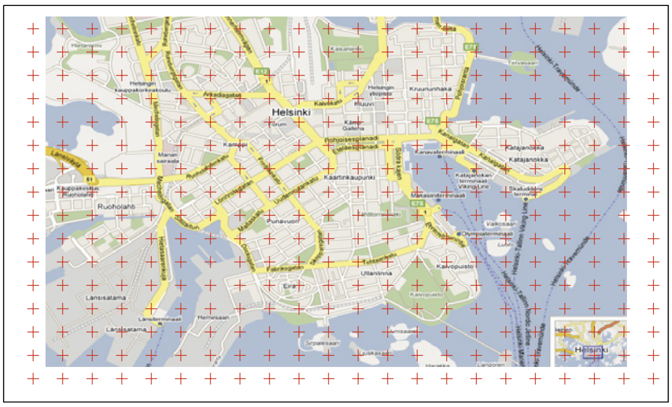


Fig. 7. Simulation area and reference anchor locations every 200 m.

Table 2
Overview of simulation scenarios.

Reference	Radio range (m)	# nodes	Ped's	Cars	Trams
$S10\ (r, a)$	10	126	80	40	6
$S50\ (r, a)$	50	126	80	40	6
$M10\ (r, a)$	10	252	160	80	12
$M50\ (r, a)$	50	252	160	80	12
$L10\ (r, a)$	10	504	320	160	24
$L50\ (r, a)$	50	504	320	160	24

(HCS) [28] based upon a city map (4500 m × 3400 m) of downtown Helsinki for evaluation (see Fig. 7). HCS models two types of nodes: most roam the city area following streets and walkways when moving to randomly chosen points on the map following a shortest path using pedestrian (0.5–1.5 m/s) or car (10–50 km/h) speeds; some follow a set of three predefined routes as trams with their own characteristic speed (25–35 km/h).

We consider this kind of tourist-style mobility of restless users as we are only interested in their local interaction within different anchor zones and do not try to achieve city-wide routing functionality. As a consequence, nodes can be treated as "new" once they have left an anchor zone and, therefore, daily routines and social relations would not add to the model (beyond possible density fluctuations over the day) so that we deliberately focus on simpler modeling.

We use the ONE simulator [28] for which we implemented:

(1) a *FloatingApplication* class initiating content postings at different locations with configurable parameters (a, r, TTL, size);

(2) a *FloatingRouter* that implements the seven replication and two deletion policies described in Section 3.3 above; and

(3) two classes *FloatingAppReporter* and *ContactConditionReport* that provide information about the node density and encounter frequencies as input parameters to our analytical model. The class *FloatingMessageReport* delivers statistics about creation, replication, and deletion of floating messages.

We create six different mobility scenarios as shown in Table 2: As noted above, we use a 2 Mbit/s data rate for the wireless links for 10 m (Bluetooth) and 50 m (WLAN ad-hoc mode) radio ranges and vary the number of nodes from 126 (small) to 252 (medium) to 504 (large scenario). While we carry out our initial feasibility studies for both radio ranges, we focus on 50 m for in-depth evaluation. Unless stated otherwise, we use 5 MB buffer space per node, except for trams that always use 50 MB.

We conduct most of our simulations with two different anchor zone sizes: $a = r \in \{200\text{ m}, 500\text{ m}\}$. To assess the feasibility of floating content in downtown Helsinki, we choose fixed anchor locations every 200 m horizontally and vertically across the entire simulation area. The resulting 352 anchor points are shown as crosses in Fig. 7.

Unless stated otherwise, we run all simulations over a period of 24 h of simulated time, use a message TTL (T_{ttl}) of one hour and correspondingly a cooldown period of 1 h. We report the mean values for a given scenario (a) across all anchor zones in the entire simulation area, using an average weighted by the number of messages generated per anchor zone and (b) restricted to a *core area* of 1.8 × 1.8 km^2 that includes diverse terrain (including water front) but where we do not expect any boundary effects due to simulation area limits.

Fig. 8 shows the mean number of node contacts per anchor zone per hour for the scenario $S50(500, 500)$ (that we will also use in Section 7.2). The core area is indicated as a square on the map. Expectedly, across all scenarios (not shown explicitly here), the contact density increases with radio range, the number of nodes, and the anchor zone size from some 230 contacts/h for the densest spot in $S10(200, 200)$ to 17,500 contacts/h in $L50(500, 500)$.

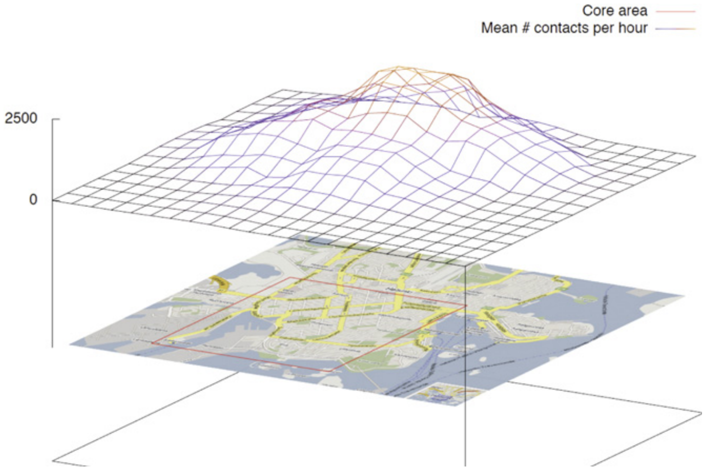


Fig. 8. $S50(500, 500)$: Mean number of contacts per hour per anchor zone.

7. Evaluation

We carry out two series of simulations: an initial set of simulations uses minimal message sizes and sufficiently large buffers to establish under which of the above mobility models floating content is feasible and how it matches to the predictions of our analytical model Section 7.1. We then evaluate the floating content concepts and different algorithms for more diverse parameter settings Section 7.2.

7.1. Feasibility considerations

We run simulations individually for each of the 352 anchor points with two different radii for the six mobility scenarios introduced above. Individual runs are used to isolate anchor zones from the impact of neighboring ones. In these scenarios, we use tiny messages so that message transfers are never aborted because nodes move out of range. Each node generates a message when it comes within $(1/2)r$ of the anchor point, but message generation per node is limited to one message per 30 min. This yields from 1350 messages for $Sx(200, 200)$ to 9350 messages for $Lx(500, 500)$ per 24 h simulation time in the busiest spots. Next we consider the spatial distribution of the probability that content floats for up to one hour for all $S10$ scenarios. We additionally validate criticality condition (4) that defines a condition under which a message will stay available with a high probability. To this end, the parameters for the mean number of customers N, the mean sojourn time $1/\mu$ and the mean contact rate ν for each anchor zone were also estimated from the simulation data (cf. Section 4).

Fig. 9 depicts the spatial probability that the content is available up to one hour in each anchor zone and a scatter plot between the criticality factor $N\nu/\mu$ and the floating probability. In the scatter plot, each sample corresponds to one of the 352 anchor zones. For illustration purposes, we have bounded the criticality factor to interval $[0, 10]$. In the left column in the figure $r = a = 200$ m, and in the right column $r = a = 500$ m. In both columns, the left figure shows the observed floating probability (fraction of messages that are available at the end of $T_{\text{ttl}} = 1$ h), and the right figure the corresponding scatter plot. From top to bottom, the number of nodes increases: 126 nodes, 252 nodes and 504 nodes.

When the criticality factor is less than 1, or close to one, the expectation is that the messages will not stay alive (i.e., loosely speaking, they will not float). As can be seen in the scatter plots, this indeed is the case in every scenario. Only for the cases where the criticality factor is clearly above the threshold the floating starts to occur.

For $r = a = 200$ m, the granularity of the underlying road network probably affects the results. Only with 504 nodes, the floating starts to occur in certain locations. When $r = a = 500$ m (right column) the criticality condition is well above the threshold almost everywhere, except in the areas corresponding to the Baltic sea. Similarly, the floating probability is close to 1 except in the sea.

The numerical results presented are based on the mean values over one full day, during which the mobility patterns change according to the Helsinki City Scenario (HCS) model. Therefore, e.g., the criticality quantity also varies as a function of time in each location. This, the finite system, and other stochastic fluctuations explain the randomness in the scatter plots. However, the main observation is that no floating occurs below the criticality threshold (bottom left corner in the scatter plots), and clearly above the threshold the messages start to float (top right corner), thus supporting the theoretical results obtained in [19].

Using a larger radio range yields further improvement as shown in Figs. 10 and 11, especially for larger r and a. These latter two figures compare the mean floating time as a fraction of T_{ttl} for a piece of content across all scenarios considering the entire simulation area (except for places in the sea where there are no messages generated) and only the core area, respectively. The mean values are weighted by the number of messages (running from a few tens to some 17,000 per hour). We find many scenarios for which floating is basically feasible, i.e., content availability $p(t = T_{\text{ttl}}) \gg 0$. Especially the large anchor zones show a trend towards slowly decaying $p(t)$. This is an indication that the system operation point is well above

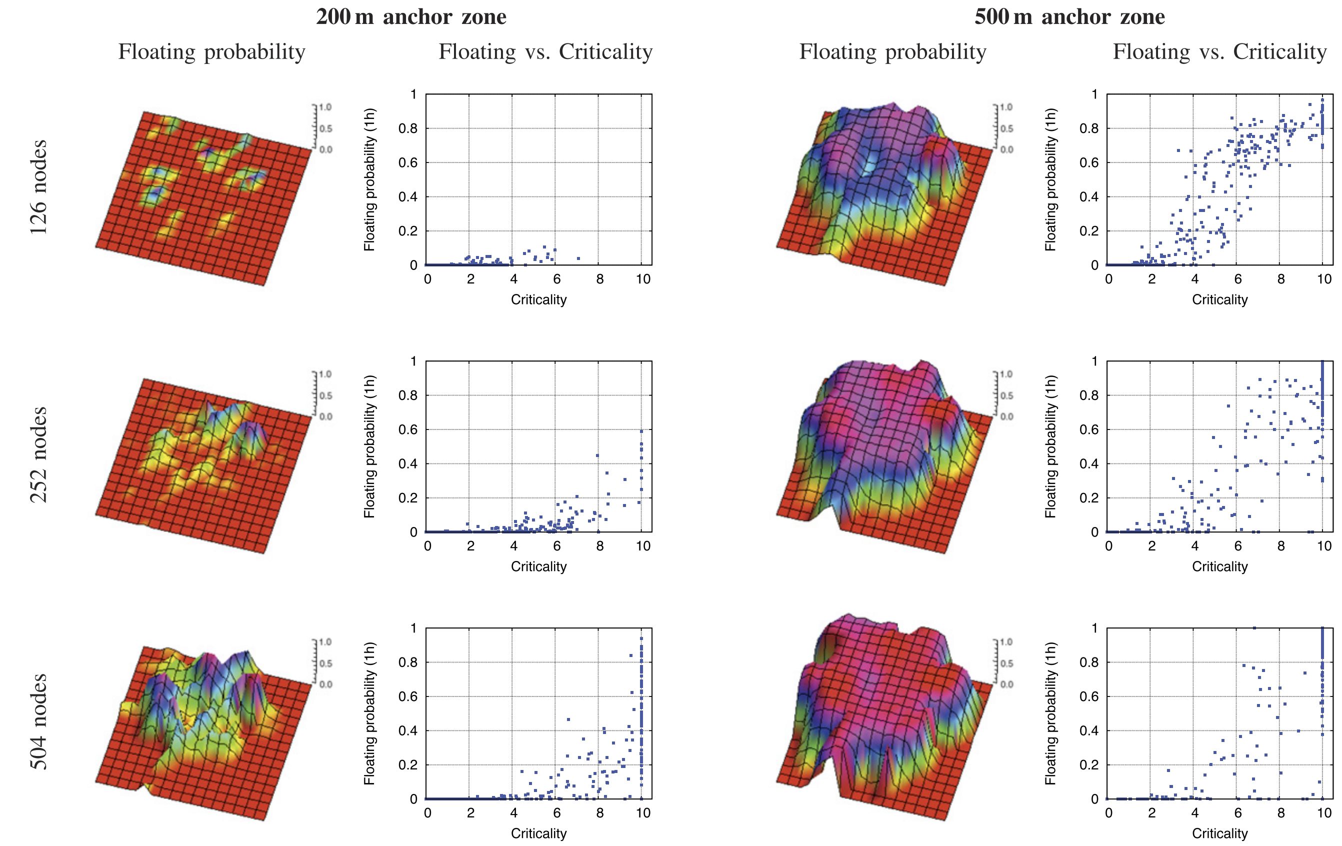


Fig. 9. The observed floating probability and the corresponding scatter plot with the criticality condition on the x-axis and the floating probability on the y-axis. The scatter plot comprises 352 anchor zone locations in each case. On the left, $r = a = 200$ m, and on the right $r = a = 500$ m. The transmit range is 10 m (Bluetooth radio). From top to bottom, the number of nodes is 126 ($S10$), 252 ($M10$) and 504 ($L10$). One can observe that no floating occurs when the criticality condition does not hold (criticality less than 1). However, as soon as the criticality quantity increases clearly above the threshold of the fluid model, i.e., from 1 towards 10, the floating starts to occur in each case. The criticality is limited to 10 in the graphs for illustration purposes.

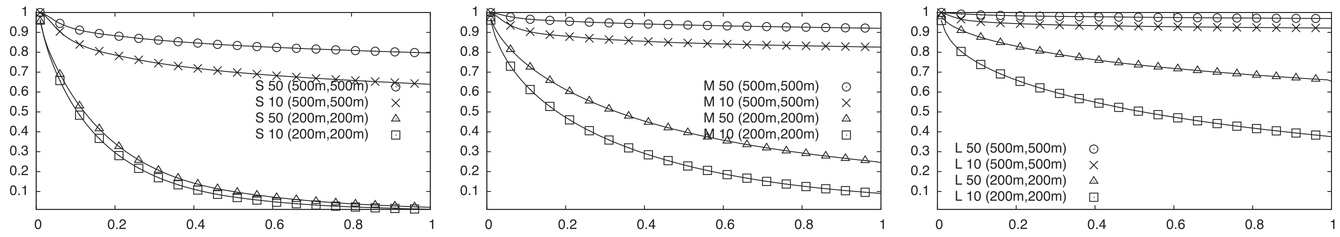


Fig. 10. Floating duration as fraction of TTL: from left to right scenarios S, M, and L.

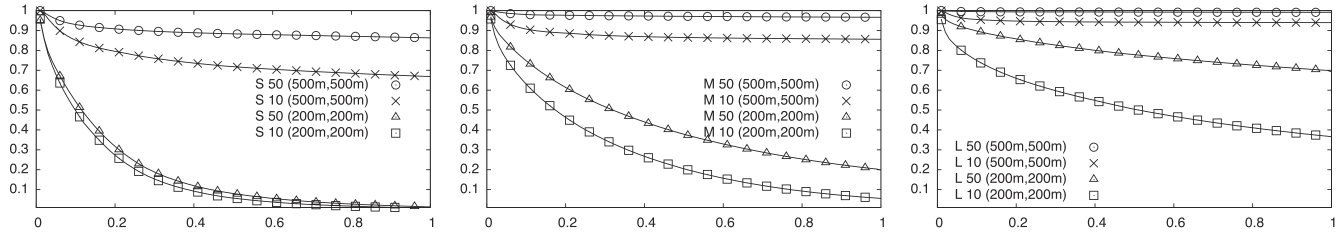


Fig. 11. Floating duration for the 1.8 × 1.8 km^2 core area excluding borders and the sea as a fraction of TTL: from left to right scenarios S, M, and L.

criticality which ensures that messages even in the simulated stochastic system will float with a high probability for a long time, say tens of hours. We also see that in those cases when messages do not float (but *sink*), the majority sinks during the first 10%–20% of their lifetime. This is an important finding as a system might be able to warn a user still close by that her message is likely to disappear soon.

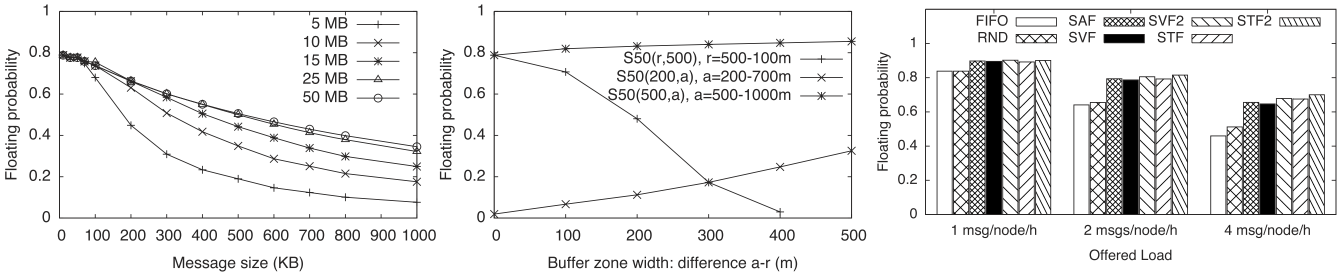


Fig. 12. Floating performance in scenario $S50(500, 500)$ as a function of (a: left) the node buffer size for different message sizes, (b: middle) when varying a and r, and (c: right) for different offered loads and replication algorithms.

7.2. Characterizing floating content performance

We choose scenario $S50(r, a)$ (with $r = a = 500$ m in most cases) for further evaluation as this provides on the one hand sufficient floating success rates for messages to realize best-effort content sharing, yet leaves room for improvement on the other hand.

One important element is understanding where limitations of floating performance come from. Given an encounter pattern, opportunistic communications can be limited by buffer size or by communication capacity when nodes encounter each other (determined by link rate and contact duration). We vary the buffer sizes available to the 120 mobile nodes and plot the floating success rate over the message size in Fig. 12(a). We find that for messages smaller than 100 KB the system is limited by the encounter capacity, for large messages buffer space becomes an issue. We also find that a 50 MB message buffer is sufficient for the system not to be buffer-limited (with the considered message sizes/message generation rate).

In a separate set of simulation runs (not shown), we chose message and buffer sizes and the link rate so that the entire buffers of two nodes could be exchanged even in the shortest possible contact (0.1 s). We found that a buffer capacity of 100 messages was sufficient to maximize floating performance (with the considered message generation rate).

The floating content algorithm provides for two distance parameters, the *replication* radius r and the *availability* radius a. While all simulations so far assumed $r = a$, we now evaluate the impact of $a > r$, i.e., the impact of a buffer zone $a - r$ in which messages are neither replicated nor deleted. In two cases, we keep r constant at $r \in \{200\text{ m}, 500\text{ m}\}$ as above and vary a in $[r, r + 500\text{ m}]$ in steps of 100 m. In the third case, we keep $a = 500$ m and vary r in $[100\text{ m}, 500\text{ m}]$. Fig. 12(b) shows the resulting floating performance plotted over the buffer zone width, the difference $a - r$.

We find that the replication radius r is, for a given node density, of crucial importance for the floating performance. Looking at the first curve, the fraction of messages floating for their TTL degrades quickly as r falls below 400 m (i.e., $a - r = 100$ m). The second curve shows that, with a small replication range $r = 200$ m, increasing the availability radius a improves data availability notably, but does not quickly reach a satisfactory level, only achieving some 33% at $a = 700$ m (i.e., $a - r = 500$ m). While the trend points upward, recall that increasing the availability range increases the resource consumption as data is spread more widely. The third curve shows that a sufficiently large replication range $r = 500$ m does not require much of a buffer zone: the floating performance increases (up to some 7% in the figure when $a = 1000$ m, i.e., $a - r = 500$ m), but not significantly and with diminishing returns. With appropriate r, a small buffer size is useful, but not much more is needed, supporting the idea of limited geographic coverage and spatial re-use of buffer and communication capacity.

We also experimented (not shown) with two different deletion policies, immediate and upon-encounter, using the same choices of r and a as shown in Fig. 12(b). We find that the floating performance shrinks by up to 0.3–0.4 when applying immediate deletion compared to upon-encounter deletion; the resulting curves for varying r and a are qualitatively similar to those in Fig. 12(b). This shows the importance of keeping content even when outside the anchor zone and supports the idea of a buffer zone. This is in line with practical implementation considerations: a node would rather act upon an external trigger (such as encountering another node) than permanently scan its buffers for data items to expire.

The expected gain from a passive buffer zone with $R(h) = 0$ (and from upon-encounter deletion) depends, obviously, on the mobility pattern. That is, a buffer zone outside the replication radius is only useful if at least some of the nodes return to the replication zone directly from the buffer area. For pedestrians, this may happen quite often in practice, while perhaps less often for vehicles.

Finally, we simulate a mixed scenario based on $S50$ with 50 MB buffers: the nodes move as before and generate content items when they are inside the *core area* at mean rates of 1 (single), 2 (double), and 4 (quad load) messages per node per hour. Each generated message is anchored at the generating node's present location. Anchor zone sizes $r = a$ are chosen from [500 m, 2000 m] in 100 m steps, message sizes from [100 KB, 1000 KB] in 100 KB steps, and TTLs from [30 min, 180 min] in 30 min steps, all uniformly distributed. We use upon-encounter deletion and simulate the seven different replication policies from Section 3.3: FIFO, RND, SAF, SVF, STF, SVF2 and STF2.

Fig. 12(c) shows the overall floating performance across all content items. We observe that the single load already appears to saturate the floating capacity of the network and that increased load leads to performance degradation. We find that those

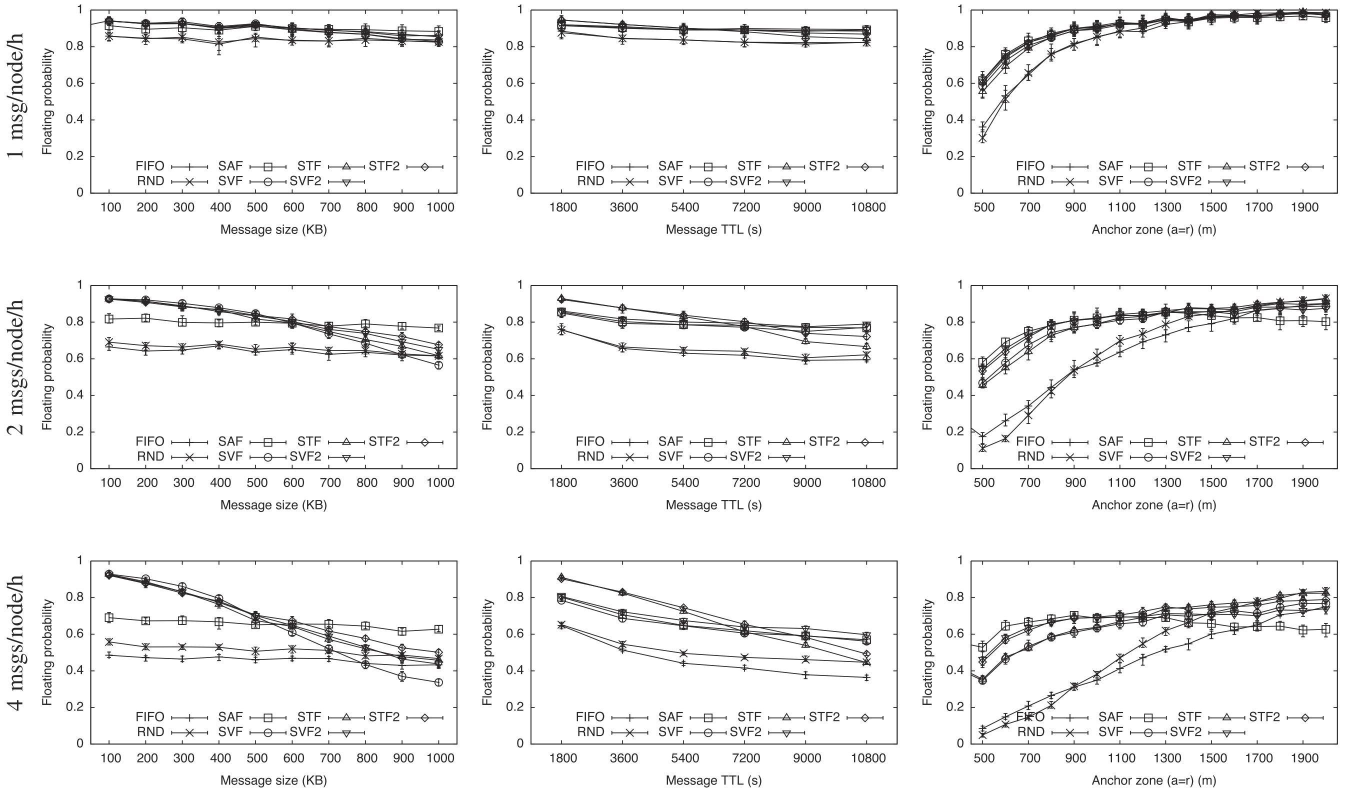


Fig. 13. Floating performance as a function of message size, TTL, and anchor zone for seven different replication algorithms.

replication algorithms taking per-message resource consumption into account perform generally better than those that do not. We also note that with less saturation, the impact of the replication algorithm is less pronounced.

We dissect the impact of the replication algorithm further in Fig. 13, plotting the floating performance as a function of message size, TTL, and anchor zone. Across all plots, we can see that the two simple replication algorithms, FIFO and RND, perform worse than those considering at least one property of the content items. The difference becomes more pronounced with increasing load and their respective prioritization becomes visible: we can influence that less resource consuming messages (smaller size or TTL) shall be preferred. The impact of conscious replication is most visible for the different anchor zone sizes. FIFO and RND do not support small anchor zones well. But the algorithms are also aware that r and a do not manage to obtain similar performance across all anchor zone sizes because their respective radii (paired with the mobility model) determines the number of contacts available for replication.

Overall, SAF (taking into account only a) performs best across all anchor zones while not adding bias towards smaller or shorter-lived messages. Adding conscious preference for smaller or short-lived content pieces comes at the expense of messages with small anchor zones, simply because we presently use a non-weighted scalar product of the message parameters for prioritization: a large (or high TTL) message with a small anchor zone will thus be treated similarly to a small (or low TTL) message with a large anchor zone. We find that considering the area (SVF2, STF2) rather than just the radius (SVF, STF) yields slightly better performance as can be seen in the summary in Fig. 12(c). The reason becomes evident in the left and middle columns of Fig. 13: while STF and STF2 (as well as SVF and SVF2, respectively) yield similar floating performance for small message sizes (left) and small TTLs (middle), STF2 and SVF2 perform better for larger and longer-lived messages because they manage to balance resource utilization better within the area in which floating content is spread.

While experimenting with different weights could be subject to future investigations, we find quite some dependency on the load (and expect similar observations for different node densities and mobility models) so that fine-tuning to a specific scenario does not appear a useful exercise. Picking STF or STF2 appears a reasonable choice as they perform best for different loads.

7.3. Buffer zone algorithms

The above evaluations have mostly assumed $r = a$, with a few controlled exceptions for which we used passive buffer zones ($R(d) = D(d) = 0$). In this subsection, we explore the impact of different choices for $R(d)$ and $D(d)$ by means of simulations for the five buffer zone replication and deletion functions introduced in Section 3.2. We use the above three different node densities, identified with letters S, M and L, with 50 m radio ranges – $S50$, $M50$, and $L50$ – so that we are able to determine the impact of the node density. All nodes have a radio range of 50 m and a buffer size of 50 MB; they use STF for replication and the encounter-based deletion policy. We consider the three different message loads introduced

Table 3
Storage capacity in relation to the offered load.

Reference	# nodes	Capacity (GB)	Single load (%)	Double load (%)	Quad load (%)
$S10(r, a)$	126	6.3	1.9	3.9	7.7
$M10(r, a)$	252	12.6	1.9	3.9	7.7
$L50(r, a)$	504	25.2	1.9	3.9	7.7

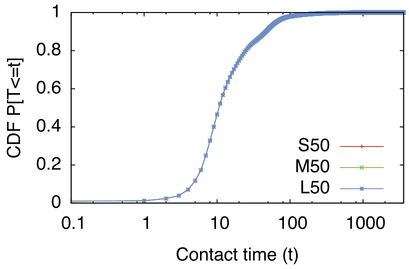


Fig. 14. Contact duration distribution for a radio range of 50 m. Note that the three curves almost exactly overlap.

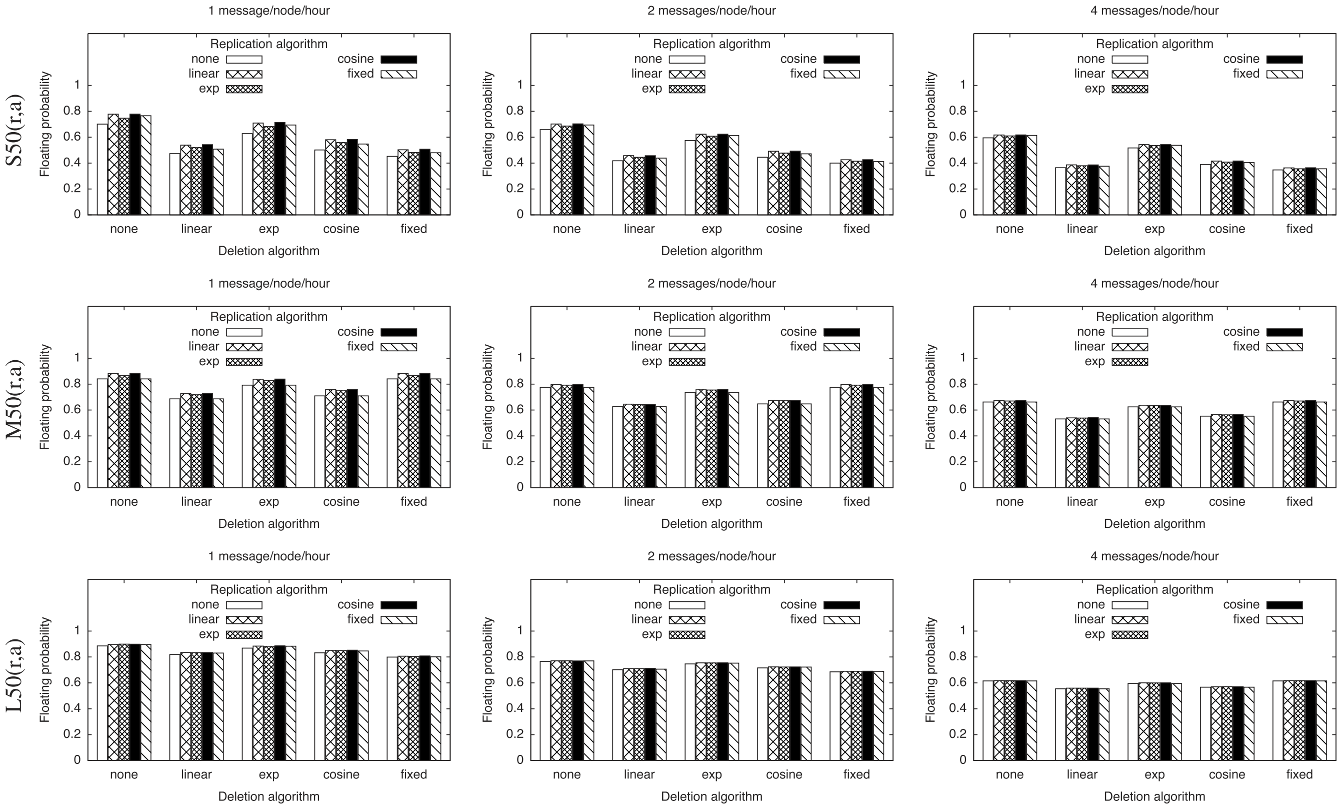


Fig. 15. Floating performance for different behaviors of the buffer zone and different node densities.

above with k messages per node per hour, $k \in \{1, 2, 4\}$, representing single, double, and quad load. For the simulations in this section, we allow anchor points across the entire simulation area. Based upon our findings above, we choose replication $r \in \{200, 300, 400, 500\}$ m and availability radii $a \in \{500, 600, \ldots, 2000\}$ m. We run the $S50$ simulations for one day (86,400 s) and the other two densities for three hours (10,800 s), with three hours of cooldown period in each case to account for the maximum TTL. We perform ten simulation runs for each of the 225 permutations of density (3), offered load (3), replication (5), and deletion policy (5).

Table 3 summarizes the node densities and offered loads, indicating the total capacity in terms of the total buffer space available when combining the storage of all the nodes at 50 MB per node. With a mean message size of 550 KB and a mean message lifetime of 6300 s, we can estimate the content volume generated per node (and in total) for the different offered loads (cf. Little's result) and set this into relation to the total storage capacity. The (average) fraction of the total capacity that would be needed to store exactly one copy of each message during its lifetime within the entire system is shown in the three right columns. Naturally, this does not change with the number of nodes as both capacity and load grow linearly with this number. However, as several copies of messages need to be stored and the nodes are not uniformly distributed in the simulation area, this leads to quite different performance. Overall, this table indicates that buffer space is unlikely the limiting factor. Fig. 14 shows the cumulative distribution function for the contact durations across all nodes for 10 simulation runs) for all three node densities and a radio range of 50 m. The mean duration is 21.4 s, the median is around 10.5 s, allowing the exchange of 2.6 MB and 5.4 MB, respectively, about 5% or 10% of the buffer. Less than 0.8% of the contact last 200 s or more and would allow exchanging all buffered content. This suggests that the link capacity is the major constraint.

Fig. 15 summarizes the floating performance across all simulation scenarios, averaged across all anchor and buffer zone sizes. Fig. 16, discussed later, provides a detailed overview of the mean performance with different deletion strategies, node density and loads as a function of the anchor and buffer zone radii (a, r).

We find that the overall floating performance is governed by the choice of deletion function rather than the replication function, especially when looking at $S50$, the top row of Fig. 15: the differences within a group (representing one deletion algorithm) are less pronounced than across groups. As can be expected from the area under the respective curves plotted in Fig. 2, the linear and cosine functions perform best when used for replication and yield bad performance when applied for deletion. Note that the fixed deletion policy is even worse for the sparsest scenario: when only few nodes are around, chances of early deletion jeopardize the chances of a message to float. This effect disappears for the two denser scenarios when sufficient nodes are around to hand replicas to. This leaves the no deletion (naturally) and exponential deletion policies as the most promising ones.

Floating performance degrades with increased message load, but also the differences between the replication functions become less pronounced. For quad load (top right in the figure), almost no difference remains between the replication functions for a given deletion function. The dominance of the deletion policy can be intuitively explained: once a message is deleted at a node, it is irrevocably gone on this node, while a replication choice yields a less clear impact: the node may move out of the anchor zone or not meet any other nodes (before its deletion function strikes) so that the gain may only be of a short-term nature.

With higher node densities, we also see first the impact of the replication functions disappear. Apparently, enough nodes are around within the replication radius so that there is no need for additional replication inside the buffer zone. The mean difference between different deletion functions also decreases, simply because enough nodes are inside the replication range and the buffer zone loses importance.

Increasing the load further contributes to equalization across different replication and deletion functions. This confirms the system design: With increased load, the contact durations get too short for two nodes to exchange all messages they have stored, so that the prioritization functions take effect and select which messages to replicate. With many nodes and many messages with overlapping anchor zones, when using STF (or SAF or SVF) those further away from their anchor points will not make it to the front of the queue for replication during a contact so that the choice of a replication probability becomes close to immaterial. Moreover, if messages do not get replicated the impact of the deletion function is reduced to the time period the messages are available on at least one node, unless the node happens to return close to or into the replication area. Therefore, while the impact of the deletion function shrinks a minor influence remains.

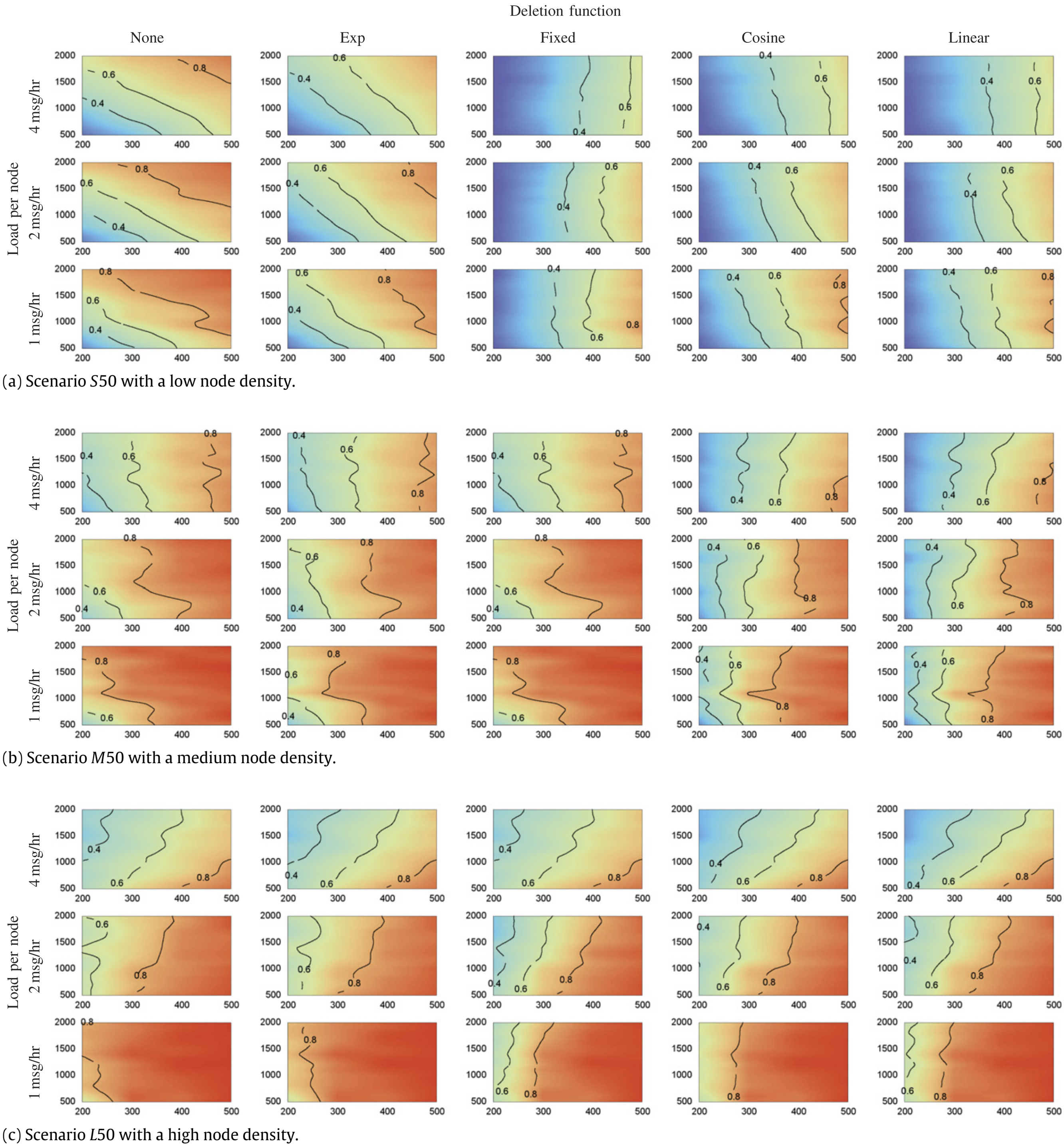
Looking at the details presented in Fig. 16, we find that the impact of different behaviors in the buffer zone and the size of the anchor zone (for both r and a) varies as a function of node density and offered load. We carried out the same simulation runs also for different replication functions and found that they have minimal effect on the floating performance. This is an important finding as it essentially reduces the number of the system parameters.

For the low density ($S50$) scenario, floating performance grows approximately monotonously (close to linearly) with increasing r until the floating probability becomes close to one. As expected, increasing a has a stronger effect for smaller than for larger r. Linear, cosine, and fixed deletion functions virtually eliminate the impact of the buffer zone. These qualitative observations hold irrespective of the offered load.

For the medium density ($M50$) scenario, growing r yields diminishing improvements in floating performance and a larger buffer zone has only an impact if r is small: for $r \in \{400, 500\}$ m, the choice of a does not appear to yield additional impact. In contrast to $S50$, only the linear and cosine deletion functions remove the impact of the buffer zone and yield lower performance. However, with increasing offered load, the positive impact of the buffer zone disappears for all deletion functions leading to a similar performance across the different deletion functions.

Finally, the high density ($L50$) scenario overturns the impact of the buffer zone: while there is no noticeable impact for a small offered load, with 2 and 4 messages per node per hour, increasing the buffer zone size reduces floating performance—an important observation that confirms our built-in DoS prevention mechanisms. We find that modest anchor zone sizes suffice and that, for lower offered loads, none and exponential deletion perform best.

Overall, these findings suggest that, for our scenarios, only low density scenarios require a buffer zone (or a wider anchor zone). If a buffer zone is used, passive operation or exponential deletion appear suitable, paired with any replication function. When the node density and load grow, the resources become scarce and congestion eventually emerges. In such a situation a buffer zone should be avoided as they rather do harm than good. Since content is linked to a geographic location, it appears sensible for a mobile node to monitor the surrounding node density and pick a buffer zone size according to these observations. Unless the node density is very high, choosing a fixed $r = 500$ m would be suitable for our scenario, but this likely depends on the details of a city's map. We leave the investigation of suitable algorithms to determine those parameters for future work.



(a) Scenario *S*50 with a low node density.

(b) Scenario *M*50 with a medium node density.

(c) Scenario *L*50 with a high node density.

Fig. 16. Floating performance with different buffer zone parameters. The variations of the replication function is not shown as it had very marginal effect on the resulting performance.

8. Applications for floating content

Fig. 17 depicts the conceptual view of a node. It has a data store on which content items from local applications and other nodes are stored. Three largely independent control loops govern the operation of the data store:

(1) The *protocol loop* is responsible for sharing items with neighboring nodes. The functionality includes peer discovery; negotiating which items to exchange, i.e., determining in which items' anchor zones both nodes are located; prioritizing the items; and exchanging them reliably. (2) The *local management loop* takes care of deleting items when the node leaves the respective anchor zone or when the items lifetime expires. It also runs the deletion function if the memory gets full. (3) The *application interaction loop* offers the interface for local applications to the data store.

We foresee a publish/subscribe-like API for applications.

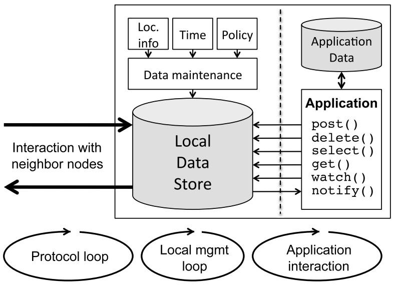


Fig. 17. System model.

post( ) creates an item and stores it in the local data store.

delete() is a purely local operation and removes an item from the local content store. The deleted item will not be replicated and the operation has no bearing on other nodes.

select( ) is a data base-style operation to retrieve a set of content items matching a given query, keyword, or metadata.

get( ) retrieves a given item from the data store.

watch() registers application interest in incoming items matching a given query, keyword, or metadata.

notify() is a callback-style operation allowing the local data store to inform an application about the reception of new items matching a previously issued *watch*( ).

We now present different application examples and show how their communication needs can be satisfied.

Graffiti: The simplest application is essentially digital graffiti, where some users post items and others read them. The application calls *post* to create new items and reading happens with *select* and *get*.

Collaborative sensing is a variant of graffiti: Consider a fully distributed sensing application, e.g., mapping of WLAN access points. Devices *post* information about discovered APs, e.g., MAC address, SSID, signal strength, etc., and implicitly their own location, using an anchor zone radius well exceeding the AP coverage. *select* and *get* allow nearby nodes to learn about APs. An application uses *watch* to ask for a *notify* when it gets close to certain APs.

Regional chat: In a chat application two (or more) users *post* messages and set appropriate filters with *watch*, e.g., matching their pseudonyms. When new messages appear, *notify* will be triggered and the application can offer the user the possibility to *post* a reply. Since all communications are targeted to specific users, *select* and *get* are not needed because *notify* will ensure that the application gets a chance to react to new messages.

Local auction: On a flea market users *post* about the (real-world) items they sell. Buyers use *select* to find items of interest and *get* for more information about an item. They can also *watch* for new items coming on sale as a passive search and then react to a *notify*. In either case, a user can *post* a reply, e.g., a bid for an item. The seller uses *watch* and *notify* to collect the bids and can *post* messages about ending the auction.

The above examples demonstrate the versatility of floating content for many different kinds of applications. The main open question in whether floating content applications will become successful relates to their probabilistic operation. As there are no guarantees about information persisting, users could find this intolerable and not be willing to invest their time, memory, or battery for these applications. On the other hand, in many cases, even probabilistic best effort communication is better than no communication at all. We believe that the only way to answer the questions about the real feasibility of the applications is to build and deploy them. Our results in this paper show that such applications are feasible from a technical point of view, but to get user acceptance, real deployments are needed.

9. Conclusion

We have presented the concept of *Floating Content*, a fully distributed variant of an ephemeral content sharing service, solely dependent on the mobile devices in the vicinity that uses principles of opportunistic networking.

Our evaluation using the Helsinki City Scenario suggests that floating content in urban environments can be feasible even with modest numbers of nodes supporting this application. Sufficiently large anchor zones are required for the node densities we investigated so that, while geographically limited sharing works, tightly constraining the availability may not be feasible unless the node density is very high. Indicatively, it seems that anchor zones should span multiple blocks in a city for sufficient replication, but a systematic exploration is for further study. We also find that – expectedly – floating content has limitations in border areas across all node densities when the movement of nodes into and out of an area is restricted, e.g., at the waterfront.

We could confirm the predictions of our analytical model for the probability that content would float in a given location. This is particularly important as the input parameters to our model can be estimated locally (possibly in cooperation with other nodes) so that a predictor could be calculated that can indicate the floating expectation to the user or an application. This is one direction of further study.

Our exploration of the impact of the buffer zone showed that their value is clearly density-dependent, calling for algorithms to dynamically assess the properties of the immediate environment to automatically choose a buffer zone. This is subject to ongoing work. In contrast to the buffer zone size, the impact of the chosen replication and deletion functions appears less important. In particular, the effect from different replication functions turned out to be insignificant suggesting that replication solely within the anchor zone is sufficient.

We based our simulations on parameter choices for communication capacity on a conservative estimate derived from a set practical experiments with different mobile devices in different settings. Yet, these experiments are limited and so are the findings from the resulting simulations. Therefore, we are targeting experimentation in diverse real-world scenarios – in the lab and in various places of Helsinki – to gain a more in-depth understanding of floating content performance in reality.

We are also well aware of the limitations of synthetic mobility models (and of HCS). Yet, we believe that macroscopic mobility and social context and interaction patterns may be of lesser significance for localized content sharing – unless selective support based upon (closed) groups or content channels come in – so that we expect our findings to hold for other scenarios as well. Nevertheless, our current and future work also includes a validation of our analytical models and our floating content algorithms for a broader set of mobility models and mobility traces with diverse node densities and system loads.

On the practical side, future efforts are to develop an Android prototype implementation as well as exploring the power consumption implications of continued location tracking as well as neighborhood sensing to determine appropriate tradeoffs for floating content operation. In the mid-term, it will be interesting to understand which more sophisticated applications could make use of such a best-effort service and how to evolve communication paradigms in support of those.

Acknowledgments

This work was supported by TEKES as part of the Future Internet program of TIVIT (Finnish Strategic Centre for Science, Technology and Innovation in the field of ICT) and by the Academy of Finland in the RESMAN project (grant no. 134363).

References

[1] S. Carter, E. Churchill, L. Denoue, J. Helfman, Digital graffiti: public annotation of multimedia content, in: Conference on Human Factors in Computing Systems, 2004, pp. 1207–1210.

[2] J. Kangasharju, J. Ott, O. Karkulahti, Floating content: information availability in urban environments, in: Proc. of IEEE Percom 2010, Work in Progress Session, March 2010.

[3] J. Ott, E. Hyytiä, P. Lassila, T. Vaegs, J. Kangasharju, Floating content: information sharing in urban areas, in: Proc. of IEEE Percom 2011, March 2011.

[4] V. Lenders, M. May, G. Karlsson, C. Wacha, Wireless ad hoc podcasting, ACM/SIGMOBILE Mobile Computing and Communications Review (2008).

[5] N. Asokan, K. Kostiainen, P. Ginzboorg, J. Ott, C. Luo, Applicability of identity-based cryptography for disruption-tolerant networking, in: Proc. 1st Intl. ACM MobiSys Workshop MobiOpp, 2007.

[6] A. Villalba Castro, G. Di Marzo Serugendo, D. Konstantas, Hovering information: self-organizing information that finds its own storage, Tech. Rep. BBKCS707, School of Computer Science and Information Systems, Birkbeck College, London, UK, November 2007.

[7] D. Corbet, D. Cutting, Ad loc: location-based infrastructure-free annotation, in: The Third International Conference on Mobile Computing and Ubiquitous Networking, ICMU 2006, 2006.

[8] I. Leontiadis, C. Mascolo, Opportunistic spatio-temporal dissemination system for vehicular networks, in: Proc. 1st Int. ACM MobiSys Workshop MobiOpp, 2007.

[9] W. Gao, Q. Li, B. Zhao, G. Cao, Multicasting in delay tolerant networks: a social network perspective, in: Proceedings of the Tenth ACM International Symposium on Mobile Ad Hoc Networking and Computing, ACM, New York, NY, USA, 2009, pp. 299–308.

[10] P. Hui, J. Crowcroft, E. Yoneki, Bubble rap: social-based forwarding in delay tolerant networks, in: Proc. ACM MobiHoc, ACM, 2008, pp. 241–250.

[11] A. Wegener, H. Hellbruck, S. Fischer, C. Schmidt, S. Fekete, AutoCast: an adaptive data dissemination protocol for traffic information systems, in: 2007 IEEE 66th Vehicular Technology Conference, VTC-2007 Fall, 2007, pp. 1947–1951.

[12] G. Karlsson, V. Lenders, M. May, Delay-tolerant broadcasting, IEEE Transactions on Broadcasting (2007).

[13] O.R. Helgason, E.A. Yavuz, S.T. Kouyoumdjieva, L. Pajevic, G. Karlsson, A mobile peer-to-peer system for opportunistic content-centric networking, in: ACM SIGCOMM MobiHeld Workshop, 2010.

[14] N. Thompson, R. Crepaldi, R. Kravets, Locus: a location-based data overlay for disruption-tolerant networks, in: Workshop on Challenged Networks, Chicago, IL, September 2010.

[15] B. Liu, B. Khorashadi, D. Ghosal, C.-N. Chuah, M. Zhang, Assessing the VANET's local information storage capability under different traffic mobility, in: Proc. of IEEE INFOCOM, 2010.

[16] E. Koukoumidis, L.-S. Peh, M. Martonosi, RegReS: adaptively maintaining a target density of regional services in opportunistic vehicular networks, in: Proc. IEEE PerCom, 2011.

[17] A. Ziotopoulos, G. de Veciana, P2P network for storage and query of a spatio-temporal flow of events, in: Proc. IEEE PerCom Workshop on Mobile Peer-to-Peer Computing, MP2P, 2011.

[18] O. Karkulahti, Distributed location-aware hovering information systems, Master's Thesis, University of Helsinki, October 2009.

[19] E. Hyytiä, J. Virtamo, P. Lassila, J. Kangasharju, J. Ott, When does content float? Characterizing availability of anchored information in opportunistic content sharing, in: IEEE INFOCOM, Shanghai, China, April 2011.

[20] P. Jacquet, B. Mans, G. Rodolakis, Information propagation speed in mobile and delay tolerant networks, CoRR, 2009..

[21] P.U. Tournoux, J. Leguay, F. Benbadis, V. Conan, M.D. de Amorim, J. Whitbeck, The accordion phenomenon: analysis, characterization, and impact on DTN routing, in: Proc. IEEE INFOCOM, April 2009.

[22] A.K. Pietiläinen, Experimenting with opportunistic networking, in: Proc. of the ACM MobiArch Workshop, June 2009.

[23] S. Buchegger, J.-Y. Le Boudec, A robust reputation system for p2p and mobile ad-hoc networks, in: Proceedings of Workshop on Economics of Peer-to-Peer Systems, Cambridge, MA, June 2004.

[24] J.R. Douceur, The Sybil attack, in: Proceedings of International Workshop on Peer-to-Peer Systems, Cambridge, MA, March 2002.

[25] G. Anastasi, E. Borgia, M. Conti, E. Gregori, Wi-Fi in ad hoc mode: a measurement study, in: Proceedings of IEEE PerCom, Orlando, FL, March 2004.

[26] R. Gass, J. Scott, C. Diot, Measurements of in-motion 802.11 networking, in: Proceedings of the Seventh IEEE Workshop on Mobile Computing Systems & Applications, IEEE Computer Society, Washington, DC, USA, 2006, pp. 69–74. [Online]. Available: http://portal.acm.org/citation.cfm?id=1158340.1158805.

[27] J. Ott, D. Kutscher, Drive-thru Internet: IEEE 802.11b for "automobile" users, in: Proceedings of IEEE Infocom, Hong Kong, March 2004.

[28] A. Keränen, J. Ott, T. Kärkkäinen, The ONE simulator for DTN protocol evaluation, in: SIMUTools '09: Proceedings of the 2nd International Conference on Simulation Tools and Techniques, New York, NY, USA: ICST, 2009.