Randomised and distributed methods for reliable peer-to-peer data communication in wireless *ad hoc* networks

H.-Z. Chou, S.-C. Wang, S.-Y. Kuo, I.-Y. Chen and S.-Y. Yuan

Abstract: Peer-to-peer (P2P) communications have attracted a great deal of attention from the network research community in recent years. However, due to the fundamental limitations of wireless environments, providing reliable data availability for P2P applications over wireless *ad hoc* networks is still a major challenge. To address the problem, a distributed and randomised scheme based on self-avoiding walks is proposed. The scheme concatenates disparate network layers, with the goal of recovering from routing failures that disrupt P2P data accessibility. In addition, a probabilistic approach is presented that explores the tradeoffs between several system parameters. Some new analysis tools, such as path coupling, are utilised which provide a better understanding of the system's operations. That the proposed concepts and techniques make a significant contribution to the design of effective and efficient P2P applications in wireless *ad hoc* networks is believed.

1 Introduction

Recent years has seen a strong demand for peer-to-peer (P2P) applications in wireless *ad hoc* networks. In a wired environment, the operation of P2P systems depends on an application-layer overlay network. However, as the P2P overlay does not reflect the underlying *ad hoc* topology, the overlay's connections create a performance bottleneck for P2P applications in *ad hoc* networks [1]. Moreover, the inherent characteristics of wireless *ad hoc* network routing protocols, such as network dynamics, unreliable wireless links and limited power supply, pose many challenges to the integration of P2P applications and the availability of P2P data access. Several packet-routing schemes have been proposed to improve fault tolerance. For example, the concept of localised route repair has been discussed in relation to *ad hoc* on-demand distance vector (AODV) and dynamic source routing (DSR) methods [2], but it has yet to be thoroughly investigated.

Theoretically, the fault-tolerance capabilities of P2P applications should be able to improve data availability over wireless *ad hoc* networks. Thus, our goal is to design application-layer fault-tolerance mechanisms for wireless mobile environments. Specifically, we consider P2P data communication problems, in which each node (peer) already has a number of objects and it issues queries to search for other objects of interest. Nodes make their own decisions about which peers to connect to, and which to query for objects. Although objects can be replicated, their contents may become obsolete without regular updates. In real-world situations, the types of objects may be diversified, for example, numerical data, a certain event or a duplicable service code. Likewise, queries may consist of file names, serial numbers or elaborate Boolean predicates [3].

Computer networks are usually composed of a layered architecture that simplifies network design and implementation. However, the current physical layer is only suitable for single-hop wireless networks, that is, it only tries to optimise receiving and transmitting primitives, which is inefficient for the relay-based communications in wireless *ad hoc* networks. Additionally, current hop-to-hop packet routing involves a great deal of unnecessary queuing and contention management at a node, and the layered design creates a bottleneck that impedes performance. As cross-layer optimisation is essential for efficient and robust *ad hoc* wireless networking, the proposed scheme has a cross-layer design base, similar to the concept proposed by Ramanathan [4]. In brief, the forwarding and routing functions are moved to the physical layer, leaving the application layer to help repair disrupted P2P data access routes.

We address the problem of re-routing data objects when regional malfunctions occur in an *ad hoc* network. This subject is attracting more attention because of some emerging applications, such as military surveillance and emergency rescue operations. Consider a large-scale, dynamic, wireless environment, such as a sensor network, in which power depletion is the primary factor affecting the operational lifetime and overall performance of the application [5]. To improve the network's efficiency and robustness, the local-control design principle is employed. Distributed algorithms based only on local information are better than centralised algorithms for constructing and maintaining the virtual infrastructure of a wireless *ad hoc* network. Ideally, locally made decisions should collectively ensure certain global properties. Moreover, Barbosa e Oliveira *et al.* [6] showed that unstructured P2P overlaying is

doi:10.1049/iet-com:20060262

Paper first received 4th May 2006 and in revised form 21st February 2007

H.-Z. Chou, S.-C. Wang and S.-Y. Kuo are with the Electrical Engineering Department, National Taiwan University, Taipei, Taiwan

S.-Y. Kuo is also with the Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan

I.-Y. Chen is with the Department of Computer Science and Information Engineering, National Taipei University of Technology, Taipei, Taiwan

S.-Y. Yuan is with the Department of Communications Engineering, Feng Chia University, Taichung, Taiwan

E-mail: sykuo@cc.ee.ntu.edu.tw

much more resistant to failures. The primary objective of our work is to facilitate the recovery of disrupted routes. As we only have local knowledge about each node, probing a network's characteristics by random walks yields favourable results [7].

In wireless *ad hoc* networks, random-walk-based techniques have the following advantages over flooding-based and centralised opposite-based methods: (1) they avoid excessive traffic (e.g. duplication transmission and control messages) across the whole network; (2) they are robust in dynamic situations; and (3) they are scalable because of their simplicity and random nature. Nonetheless, most of the proposed random walk approaches for P2P applications on the Internet cannot be applied to our problem directly. The recently proposed rumour routing algorithm [8] and the sticky search algorithm [9] are al-most similar to our approach. In the former, the nodes record information about interesting events and corresponding paths. Then, if a query agent propagated by a random walk intersects with an event path, it exploits the stored information to efficiently route itself to the location of the event [8]. Shakkottai [9] proposed an asymptotic method to evaluate the intersection probability of a 'sticky search', which is formulated as Brownian motions in a lattice network [9]. He claimed that the probability of two Brownian motions failing to intersect decreases over time. These approaches, however, may cause unnecessary memory and power consumption because of their self-looping nature. To avoid being restricted to specific regions, our repair scheme adopts a self-avoiding walk, which does not select areas in the path record. We believe that self-avoiding walks can provide better scalability in wireless *ad hoc* networks.

Our contribution in this paper is 2-fold. First, we propose a distributed randomised scheme based on self-avoiding walks and a cross-layer design that increases the reliability of P2P data access in a wireless mobile environment. Second, we present probabilistic analyses and simulations to demonstrate the efficacy of the proposed scheme. To the best of our knowledge, no other work has performed similar analyses to characterise the problem of P2P access in *ad hoc* networks.

The remainder of the paper is organised as follows. In Section 2, we describe the system model and the proposed repair scheme. Section 3 provides probabilistic analyses of several performance issues. Section 4 details the results of our simulations of the proposed scheme, and Section 5 contains a discussion of the scheme. Then, in Section 6, we present our conclusions and indicate the direction of future research.

2 Proposed method

The proposed method recovers disrupted routes in wireless *ad hoc* networks by performing self-avoiding walks, and then transmits undelivered target objects via another path. The most important design principles for *ad hoc* networks are efficient use of power, resilience in dynamic situations, resistance to failure and scalability. Furthermore, we introduce the concept of damage limitation. The region to be repaired, which is determined by the transmission radius of the nodes and the length of the walks, is called the 'repair-region' throughout this paper (Basically, this region is dependent on the specific requirements of the upper application and object distribution. Moreover, it should be self-adjusting. For brevity, we do not discuss these issues in this section.).

2.1 System model

We consider a wireless *ad hoc* network consisting of n mobile devices, each of which is assigned a unique identifier and equipped with an omni-directional antenna for wireless communications. The network can be represented as an undirected graph $G = (V, E)$, where $|V| = n$, each node in V corresponds to a mobile device and the edges in E correspond to possible connections between the nodes. We assume that all nodes are equal in terms of their transmission radius, R, and that each radio propagation channel is symmetric, that is., if $(u, v) \in E$, nodes u and v are neighbours. Consequently, the graph G can be modelled as a unit-disk graph. Let $|u, v|$ denote the hop-distance between nodes u and v. For a node u, the set $N^h(u) = \{v \in V\colon |u, v| \le h\}$ is called the h-hop neighbourhood of u. As global knowledge of the network topology is not available, techniques, such as periodic updates via beacons, can be used to determine whether an edge from a node to a neighbour is still valid.

In this paper, a network is assumed to be a relaxed asynchronous model, that is, the upper bounds of the execution speed, message transmission delay, and clock draft rate are known [10]. Although perfect synchronisation is impossible, these requirements are achievable based on several reasonable assumptions. As in previous works, such as [11], all nodes are synchronized in rounds (As mentioned in the literature, rational synchronisation can be achieved using extra facilities, for example, with the help of global positioning system signals or other beacons. Given sufficient precision, the work in [10] presents a design and analysis of round-based self-stabilising protocols for wireless *ad hoc* networks.) and wireless communications are modelled as a one-to-all local-broadcast operation. In other words, when a node transmits a message m, all neighbours in the node's transmission radius can receive it. We assume that local-broadcast operations can be accomplished within a constant time period t_{lb}.

The query–response mechanism of P2P data communications works as follows. We assume that all nodes participate in a P2P application, and each node is seen as a data source containing various objects. Any node can issue a query to search for an object o. As the query is propagated over the network, any node(s) possessing the target object will respond to the querying node. As more than one node may have the target object, the upper P2P application must identify at least one node with an acceptable cost and then send a request, q, to it to retrieve the target object. Tradeoffs between the average response time and the traffic cost incurred by queries are crucial and have been investigated extensively [12, 13].

2.2 Single repair of single object

We propose a distributed randomised scheme, called single repair of a single object (SRSO), which is scalable and has a lower recovery overhead than hierarchical or centralised approaches. To reduce the maintenance costs of self-avoiding walks, the memory and communication requirements should be considered. In addition, a timeout mechanism is employed so that receiver nodes do not have to wait too long for a message. SRSO works as follows. A querying node issues a request q to retrieve an object o within a certain timeframe. If o remains undelivered at the end of this period, some nodes will launch self-avoiding walks in an attempt to repair a lost connection. If the walks encounter a crossing point, a valid path is identified and the repair operation is considered finished. As mentioned earlier, the scheme runs in discrete rounds. First we consider a simple scenario in which a node u issues a request q to a node v that possesses a target object o. Let T_u denote the local clock value of node u and Δ_σ denote the upper bound of the clock drift. Initially, node u starts the timer with timeframe t_{ou}, which relates to the hop distance between node u and node v. As longer paths are more likely to fail and should have longer timeframes, the asymptotic behaviour of t_{ou} can be expressed as $O(|u, v| \cdot t_{\text{lb}})$ (This period also correlates with the underlying failure-detection mechanism and practical considerations, such as the probability of a successful transmission and signal interference.). SRSO comprises two phases: the failure-detection phase and the local-repair phase.

2.2.1 Phase I. Failure detection: A timestamp with an initial value of $(T_u + t_{ou})$ is attached to the request q. Once node v receives q, the backward timeout value $t_{ov} = \max\{0, \ (T_u + t_{ou} + \Delta_\sigma) - T_v\}$ is attached to the object o. Note that this value is modified by each relaying node along the path from v to u. For example, when node j receives a message from node i (either all or a portion of an object o), it subtracts the value $(|T_j - T_i| + \Delta_\sigma)$ from the remaining time. The timer of node j is then set to this new timeout period. If node j receives an acknowledgment from the next relaying node before the time expires, the timer will stop. However, if o is not delivered to node u and the timer of some 'blocked' intermediate node r expires, the randomised local-repair procedure will be triggered.

2.2.2 Phase II. Local-repair (rpc): When a timeout period expires, node u and the blocked node r launch a local-repair procedure to construct new independent paths. The procedure comprises κ paths identified by κ walks, which are indexed from rw_1 to rw_κ. Each walk rw_i is directed by a probing node, denoted as cp(rw_i), which typically changes for each new round. Note that, for each walk, the number of nodes that the probing node must communicate with depends on the minimum memory requirements and the maximum number of potential simultaneous random walks. Self-avoiding walks are ideal in such circumstances. Specifically, rw_u and rw_r are represented by a list of tuples $\langle id, index\rangle$, where 'index' is associated with the round in which the node the identifier 'id' acts as the probing node. Obviously, each tuple's 'id' is unique. If two complementary paths cross, a detour is found to re-route the object o, and the repair operation is accomplished (This detour might be obsolete due to topology changes; here, we assume that object delivery can be completed successfully in most cases.). Fig. 1 shows a simple illustration of the local-repair procedure with κ=2. The walks launched by nodes u and r are denoted as rw_u and rw_r, respectively; and the probing nodes cp(rw_u) and cp(rw_r) are initialised as u and r, respectively. Each probing node switches to newly added nodes (if any). The point where two random walks (depicted by the dotted lines with arrows in the figure) intersect is called a crossing. Note that when new nodes selected by different random walks, a crossing occurs if the boundaries of the random walks are separated by a distance of at most R. Since the decisions made by a random walk in each round are based exclusively on the information available in its vicinity, the scheme is resilient and highly scalable. The pseudo-code of SRSO is presented in Fig. 2. To avoid over-complex notations and possible confusion, both rpc and $\{rw_i| \ i = 1, 2, \ldots, \kappa\}$ denote the corresponding node set.

In Fig. 2, t_{oi} denotes the timeout setting allocated for the local repair operation. The function $f_{tw}(\text{cp}(rw_i))$ assigns each l-hop neighbour a network topology-based value (The value of l is a system parameter. In practice, choosing a proper value will increase the probability of finding a crossing. However, we simply set $l = 1$ here. Further details are given in Section 3.). Then r_i is selected from $N^l(\text{cp}(rw_i))$ based on the assigned values. In line 5, c_i assigns a non-zero constant σ to every self-loop probability. Finally, if cp(rw_i) selects a new participating node that is already in the path record of rw_i, which means that a cycle exists in the walk, then rw_i executes a popping operation. The popped node (also called a top) is the node with the highest round index. In the popping operation illustrated in Fig. 3, the subscripts of the nodes are labelled in index order. These settings are based on the self-avoiding property discussed above. Additionally, to limit power consumption, an upper bound L imposes a length constraint on each walk.

For ease of analysis, we assume that self-avoiding walks launched from nodes i and j execute the pseudo-code alternately. Specifically, in round t (when t is odd), cp(rw_i) tries to select a new participating node and cp(rw_j) performs the same step when t is even. Note that the performance of SRSO can be further enhanced if the walks are applied to some virtual infrastructure (e.g. a dominating set) or with

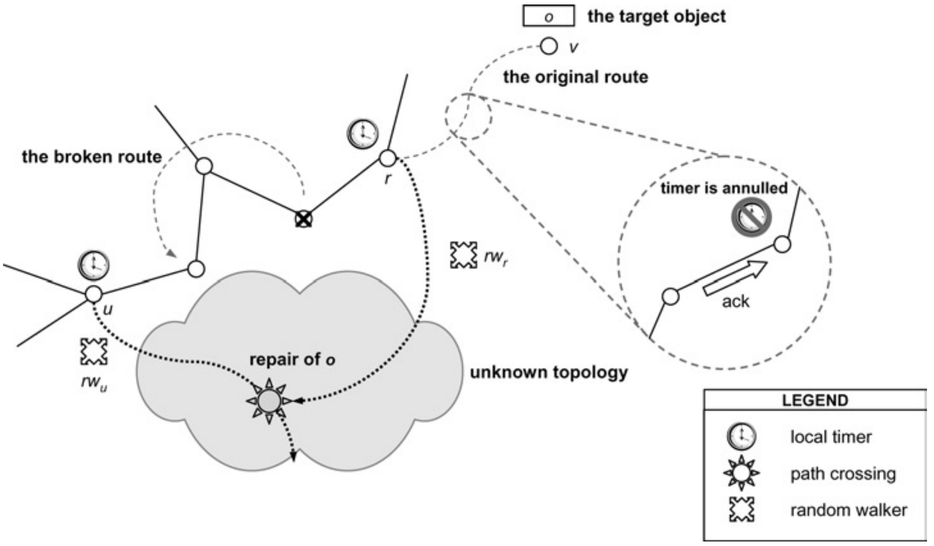


Fig. 1 *Simple scenario of local-repair via self-avoiding walks*

```
Local-Repair Scheme for Random Walker $rw_i$
on alarm of $t_{oi}$ do {
  repeat {
    for the node on behalf of $cp(rw_i)$ do {
      pick $c_i \in \{0,1\}$ and $r_i \in N^l(cp(rw_i))$ according to $f_{tw}(cp(rw_i))$;
      if($c_i = 0$) then $rw_i \leftarrow rw_i$;
      else {
        if($|r_i \in rw_i|$) then $rw_i \leftarrow$pop($cp(rw_i)$);
        else {
          if ($|rw_i| < L$) then $rw_i \leftarrow rw_i \cup \{r_i\}$;
        }
        update $rpc$ if necessary;
      }
    }
  } until (a crossing occurs in $rpc$)   /*local-repair successes*/
}
re-start the local-repair procedure if necessary;   /*a failure is detected*/
```

Fig. 2 *Pseudo-code of SRSO*

specific goals (e.g. object replication). However, to facilitate wider applicability, we do not assume any specific optimised scheme for SRSO in this paper.

3 System analysis

3.1 Analysis of system parameters

We begin by considering a local-repair scheme, which is similar to a traditional random walk over an n-node lattice graph. Specifically, we assume that SRSO does not perform a popping operation. Thus, the application of SRSO can be described by previous studies [9, 14]. Formally, if n is large enough, the random walk can be approximated as a Brownian motion with an exponentially distributed timeout setting. Shakkottai [9] derived the probability derived the probability $P_f(t)$ that no intersection would be found for k Brownian motions with independent timeout settings, where t denotes the corresponding time steps and $k \geq 2$. For $k = 2$. P_f(t) satisfies the following inequality [9]

$$P_f(t) \leq (1 - e^{-2/t}) + \frac{c\Gamma(3/8, 2/t)}{t^{5/8}}$$

where $\Gamma(\cdot,\cdot)$ is an incomplete Gamma function.

Note that the value of Γ (3/8, 2/t) approximates to a constant 2.37 as. $t \to \infty$. Therefore, the probability of finding a crossing for rw_i and rw_j scales asymptotically as. $\Theta(e^{-2/t} - t^{-5/8})$ asymptotically. In this sense, a disrupted route is likely to be repaired as t increases. Furthermore, the number of distinct nodes traversed by the search process scales as $\Theta(t/\log t)$ [14]; therefore each walk only needs to keep the tuples of the distinct nodes it visits. In this paper, we assume that t is bounded by $\Theta(n)$ Actually, the simulations reported in Section 4.2 show that n time steps are usually sufficient to finish a repair operation. Hence, the asymptotic behaviour of the memory requirement of our local repair scheme, L, can be estimated by. $\Theta(n/\log n)$.

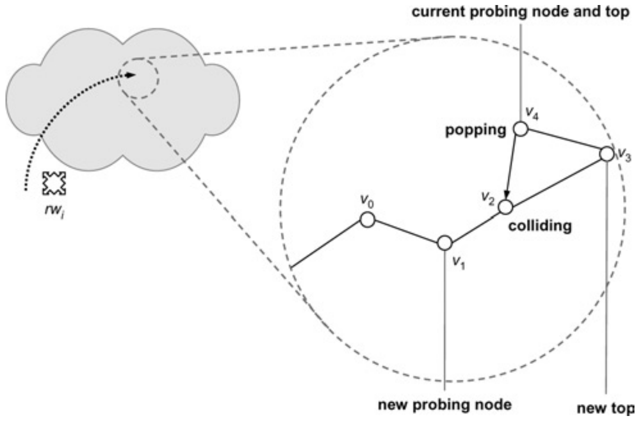


Fig. 3 *Illustration of popping a node from a walk*

In these networks, message communications are the main cause of power depletion [15]. Thus, the communication cost (and by extension, power consumption) is a major concern. Let N_{lb} denote the number of trials before a packet is successfully transmitted in one hop, and let P_{lb} denote the probability of a successful transmission when a random access scheme is used. The value of P_{lb} is mostly determined by the underlying MAC protocol. As the expected value of N_{lb} is represented as $E[N_{\text{lb}}] = \sum_{i=1}^{\infty} iP_{\text{lb}} (1 - P_{\text{lb}})^{i-1} = 1/P_{\text{lb}}$, we anticipate that the number of packet transmissions needed to diffuse information inside a random walk (e.g. for a newly added node) will be $O(L/P_{\text{lb}})$. For a CSMA/CA reservation scheme with RTS and CTS frames (i.e. IEEE 802.11 or a similar protocol), the analysis of P_{lb} can be found in [16].

The scheme discussed here does not include a popping operation because the analysis of connections between rounds and L is much more complicated with the self-avoiding property. However, it is clear that the execution time of the local-repair procedure is dependent on a rapidly mixing time. In the following section, we introduce a sophisticated technique called path coupling to evaluate the mixing time.

3.2 Analysis of the convergence rate with Markov chain-based techniques

For a randomised algorithm $\mathcal{R}$, the Chapman–Kolmogorov equation is usually employed to calculate the time needed to absorb the Markov chain associated with $\mathcal{R}$. Because of the stochastic behaviour of our SRSO scheme, it is very difficult to determine the inter-relationships between the link-level error probability and hop-level random walks, particularly when the walks involve multiple repairs. However, as we are primarily interested in the probabilistic upper bound of the completion time, we employ a novel technique called path-coupling to evaluate the convergence time [17]. Specifically, a path coupling derives the convergence rate of $\mathcal{R}$ for a set of legal configurations $\mathcal{L}$. Generally, the coupling time is used as an upper bound of a Markov chain's mixing time, that is, the time at which $\mathcal{L}$ will be reached with high probability (ε-absorption time [18]). The application of the algorithm $\mathcal{R}$ can be seen as applying a Markov chain over the state space Ω such that the probability of going from configuration x to another configuration y, denoted by $P(x, y)$, depends solely on the state of the current system. Hence, the algorithm $\mathcal{R}$ can be characterised by the transition matrix $\mathcal{M}$ on $\Omega \times \Omega$. If the (ε-approximate) mixing time is bounded by a polynomial $\ln(\varepsilon^{-1})$ and the size of Ω, a Markov chain is said to be rapidly mixing.

As noted earlier, the state space Ω of our model can be described by all the combinations of *rpc* and the state of each node can be expressed as $s_i = (r_1, r_2, \ldots, r_\kappa)$, where $r_j \in \{0, 1\}$. The case $r_j = 1$ indicates that node i belongs to the random walk j; otherwise, node i is outside of the random walk j. Consequently, each possible configuration in Ω is expressed as $(s_1, s_2, \ldots, s_n)$. Note that (1) the number of possible states of s_i is 2^κ, thus the size of Ω is $2^{\kappa n}$; (2) a change in *rpc* may cause some movement from one configuration to another in Ω; and (3) $\mathcal{L}$ consists of all scenarios in which a crossing occurs between the random walks. We observe that there is a positive probability that rw_i and rw_j will cross at another node because of a predefined sequence of popping and probing operations. Such a crossing indicates a further legal configuration. In other words, is strongly connected. As shown in Fig. 4, random walks rw_u and rw_r are launched by nodes u and r, respectively. If $rw_u = \{u, rw_{(u)}, v_1, v_2\}$ and $rw_r = \{r, rw_{(r)}, v_4, v_3\}$, there exists a legal configuration $\mathcal{L}_1 = \{u, rw_{(u)}, v_1, v_2, v_3, v_4, rw_{(u)}, r\}$. Because the sequence of operations pop(v_2), pop(v_3), probing(v_7), probing(v_5), and probing(v_6) is performed interactively by rw_u and rw_r, another legal configuration $\mathcal{L}_2 = \{u, rw_{(u)}, v_1, v_7, v_6, v_5, v_4, rw_{(u)}, r\}$ can be obtained. Moreover, $\mathcal{L}$ denotes the states with a non-zero probability in the stationary distribution of $\mathcal{M}$. Unlike many previous studies, in this work, the state space does not refer to the walk distribution over the communication graph G. Instead, we prove that the process of local-repair is a rapidly mixing operation over Ω (i.e. it converges to $\mathcal{L}$ in a time-efficient manner). For brevity, the upper bound of L is assumed to be large enough to cover the repair region, and every self-loop probability σ is set to $1/2$ (this technique is also known as a lazy chain) to ensure the aperiodicity of $\mathcal{M}$.

Proposition 1: There exists a subset S of $\Omega \times \Omega$, an integer-valued metric δ on $\Omega \times \Omega$ in the range $\{0, 2, 4, \ldots, 2n\}$ and a coupling defined on S for SRSO. Moreover, the transition matrix $\mathcal{M}$ is rapidly mixing with a value $0 < \beta < 1$, which is communication-graph-dependent.

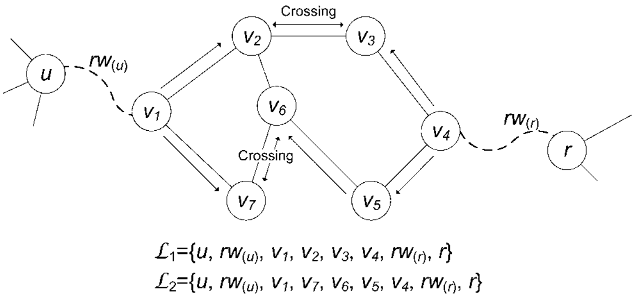

Crossing
v_2
v_3
u
$rw_{(u)}$
v_1
v_6
v_4
$rw_{(r)}$
r
Crossing
v_7
v_5
$\mathcal{L}_1=\{u, rw_{(u)}, v_1, v_2, v_3, v_4, rw_{(r)}, r\}$
$\mathcal{L}_2=\{u, rw_{(u)}, v_1, v_7, v_6, v_5, v_4, rw_{(r)}, r\}$

Fig. 4 *Illustration of the transitions over Ω*

Proof: First, following the techniques in [19], we employ a special quasi-metric $\delta(X, Y) = |X \oplus Y| + \Phi(X, Y)$, where $\oplus$ denotes the number of differences between two sets, rpc_X and rpc_Y, and $\Phi(X, Y) = \||rpc_X| - |rpc_Y|\|$. We define a coupling over the set of selected adjacent pairs based on δ as follows. Let $S = \{(X, Y)\colon X, Y \in \Omega, \delta(X, Y) = 2, N^1(\mathrm{cp}(rw_X)) = N^1(\mathrm{cp}(rw_Y))\}$. It is easy to deduce that both δ and S satisfy the conditions required by path coupling. Both Markov chains attempt to choose new participating nodes $\{r_X, r_Y\}\ \forall (X_t, Y_t) \in S$. Note that $\delta\,(X_t, Y_t) = 2$ and (X_{t+1}, Y_{t+1}) need not be in S. Next, we consider the cases where $\delta\,(X_t, Y_t) = 2$.

Case I: $|X \oplus Y| = \Phi(X, Y) = 1$, which indicates that there is only one difference between rpc_X and rpc_Y. Let $Y_t = X_t \setminus \{p\}$ for some $p \in V$ (the other case is symmetric). The coupling rule is defined as follows.

(a) Select $c_X \in \{0,\ 1\}$ and $r_X \in N^1(\mathrm{cp}(rw_X))$ u.a.r. If $r_X \in rpc_X$ and $p = \mathrm{top}(X_t)$: $c_Y = 0$, select $r_Y \in N^1(\mathrm{cp}(rw_Y))$ u.a.r. Otherwise, $c_Y = c_X$ and $r_Y = r_X$.
(b) If $c_X = 0$, set $X_{t+1} = X_t$. Otherwise: (i) $r_X \in rpc_X$: if $p = \mathrm{top}(X_t)$, set $X_{t+1} = \mathrm{pop}(X_t)$; else $X_{t+1} = X_t$. (ii) $r_X \notin rpc_X$: set $X_{t+1} = X_t \cup \{r_X\}$.
(c) If $c_Y = 0$, set $Y_{t+1} = Y_t$. Otherwise: (i) $r_Y \in rpc_Y$: if $p = \mathrm{top}(X_t)$, set $Y_{t+1} = \mathrm{pop}(Y_t)$; else $Y_{t+1} = Y_t$. (ii) $r_Y \notin rpc_Y$: set $Y_{t+1} = Y_t \cup \{r_Y\}$.

Case II: $|X \oplus Y| = 2$ and $\Phi(X, Y) = 0$, which indicate that there are two differences between rpc_X and rpc_Y, but the sizes of the sets are the same. Let $X_t \backslash \{p\} = Y_t \backslash \{q\}$, $p \neq q$, $\{p, q\} \subseteq V$. The coupling rule is defined as follows:

(a) Select $c_X \in \{0,\ 1\}$ and $r_X \in N^1(\mathrm{cp}(rw_X))$ u.a.r. Set $c_Y = c_X$. If $r_X = q$, set $r_Y = p$; else if $r_X = p$, set $r_Y = q$. Otherwise, $r_Y = r_X$.
(b) If $c_X = 0$, set $X_{t+1} = X_t$. Otherwise: (i) $r_X \in rpc_X$: if $p = \mathrm{top}(X_t)$, set $X_{t+1} = \mathrm{pop}(X_t)$; else $X_{t+1} = X_t$. (ii) $r_X \notin rpc_X$: set $X_{t+1} = X_t \cup \{r_X\}$.
(c) If $c_Y = 0$, set $Y_{t+1} = Y_t$. Otherwise: (i) $r_Y \in rpc_Y$: if $p = \mathrm{top}(X_t)$ and $q = \mathrm{top}(Y_t)$, set $Y_{t+1} = \mathrm{pop}(Y_t)$; else $Y_{t+1} = Y_t$. (ii) $r_Y \notin rpc_Y$: set $Y_{t+1} = Y_t \cup \{r_Y\}$.

For clarity, we illustrate all possible situations in Tables 1 and 2. It is not hard to validate that $\exists \beta$, $0 < \beta < 1$ such that $E[\delta(X_{t+1}, Y_{t+1})] = \beta \cdot \delta(X_t, Y_t) < \delta(X_t, Y_t)$. □

Using Proposition 1, we can derive Proposition 2 directly from the properties of rapidly mixing Markov chains [20].

Proposition 2: The mixing time of SRSO is no greater than $\ln(2n\,\varepsilon^{-1})/(1 - \beta)$.

4 Simulation results

We have evaluated the relationship between the system parameters and the mixing time of SRSO in a simple case. However, when SRSO involves multiple walks, the evaluation of L and β for random networks is too complex to analyse using mathematical models. In this section, we use the average length of walks $\bar{L}$ and the average mixing time $\bar{t}$ as metrics to evaluate the effectiveness of SRSO with different selecting functions f_{tw}. $\bar{L}$ denotes the length of walks when a crossing event occurs and $\bar{t}$ denotes the rounds of walks for a successful repair operation. We study these metrics as

Table 1: Coupling for Case I: $Y_t = X_t \backslash \{p\}$ for some $p \in V$

c_X	r_X	p	X_{t+1}	c_Y	r_Y	Y_{t+1}	$\delta(X_{t+1}, Y_{t+1})$
0	$r_X \in rpc_X$	$p = \text{top}(X_t)$	X_t	0	$r_Y \in N^1(cp(rw_Y)$	Y_t	2
	$r_X \in rpc_X$	$p \neq \text{top}(X_t)$	X_t	0	$r_Y = r_X$	Y_t	2
	$r_X \notin rpc_X$	$p = \text{top}(X_t)$	X_t	0	$r_Y = r_X$	Y_t	2
	$r_X \notin rpc_X$	$p \neq \text{top}(X_t)$	X_t	0	$r_Y = r_X$	Y_t	2
1	$r_X = p \in rpc_X$	$p = \text{top}(X_t)$	$X_t \backslash \{p\}$	0	$r_Y \in N^1(cp(rw_Y)$	Y_t	0
	$r_X = p \in rpc_X$	$p \neq \text{top}(X_t)$	X_t	1	$r_Y = p \notin rpc_Y$	$Y_t \cup \{p\}$	0
	$r_X = q \in rpc_X$				$r_Y = q \in rpc_Y$	Y_t	2
	$r_X = r \notin rpc_X$	$p = \text{top}(X_t)$	$X_t \cup \{r\}$	1	$r_Y = r \notin rpc_Y$	$Y_t \cup \{r\}$	2
	$r_X = r \notin rpc_X$	$p \neq \text{top}(X_t)$	$X_t \cup \{r\}$	1	$r_Y = r \notin rpc_Y$	$Y_t \cup \{r\}$	2

functions of the node density, the number of walks and the f_{tw} functions. In the simulations, random, weighted and balanced functions are adopted as f_{tw}.

4.1 Configuration of parameters

All simulations were performed on a static topology, with a maximum of 3000 nodes uniformly and randomly distributed in a unit-area square. We set $l = 1$ so that each node can periodically exchange information with its one-hop neighbours. Additionally, the critical transmission radius for d-connectivity in a unit-area square must satisfy $r_n \geq \sqrt{((\log n + (2d - 1) \log \log n + \xi)/\pi n)}$ [21]. When $d = 1$, the graph may contain an articulation vertex. Without loss of generality, we set $d = 2$ and $\xi = 0$. For each simulation, two nodes are randomly selected as initial probing nodes to execute the repair operation. To reduce the uncertainty of self-avoiding walks, we executed the procedure 1,000,000 times on the same graph and calculated the average value.

We evaluated the following f_{tw} functions:

(a) Random function: a simplex strategy whereby the probing node selects one of its neighbours at random.
(b) Weighted function: each node includes a counter m_i that records the number of times the node has been chosen. The weight is presented as the inverse of the counter; thus, the probing node randomly selects a neighbour based on $(1/m_i)/\sum_{i \in N^1(\text{cp}(rw))}(1/m_i)$. In other words, a node with a smaller counter value has a higher probability of being chosen.
(c) Balanced function: this strategy is very similar to the weighted function. However, instead of selecting a node in a probabilistic weighted manner, the probing node always selects the neighbour with the smallest counter value. This strategy can direct a walk in different directions.

To summarise, random and weighted functions are applied in a randomised manner, whereas the balanced function is applied in a deterministic manner. For example, consider a case where a node has three neighbours, n_1, n_2 and n_3, whose counters are $m_1 = 1$, $m_2 = 2$ and $m_3 = 4$, respectively. If the random function is adopted, n_1, n_2 or n_3 could be selected with equal probability. However, if the weighted function is used, the respective probabilities would be $P_1 = 4/7$, $P_2 = 2/7$ and $P_3 = 1/7$. As node n_1 has the lowest counter value, it is always selected when the balanced function is applied.

4.2 Analysis of experimental results

A self-avoiding walk in a random network topology is highly dependent on the degree of each node. We first evaluate the situation on a d-connectivity graph, where $d = \{1, 2, 3\}$. As shown in Fig. 5*a* and *b*, a topology with higher connectivity can execute a repair operation with a smaller $\bar{L}$, which indicates lower power consumption. Higher connectivity also leads to a shorter $\bar{t}$. Note that when the degree of the nodes in a random network is too low, walks are constrained to a specific region, resulting in a longer mixing time. This will cause an unstable curve like that of a 1-connectivity network topology. To avoid this critical problem, we applied the weighted and balanced selecting functions and compared the results with those of the random function. As shown in Fig. 6, the average mixing time of the weighted function and the balanced function is much shorter than that of the random function. $\bar{L}$ of the balanced function was almost the same as that of the random function. However, as the weighted function extends the explored region, it is more time-consuming.

Intuitively, if we launch more than one walk, the average length of the walks and the average mixing time can be reduced. In Fig. 7, we simulate cases where nodes launch

Table 2: Coupling for Case II: $X_t \backslash \{p\} = Y_t \backslash \{q\}$, $p \neq q$, $\{p, q\} \subseteq V$

$c_X = c_Y$	r_X	p	X_{t+1}	r_Y	q	Y_{t+1}	$\delta(X_{t+1}, Y_{t+1})$
0	*	*	X_t	*	*	Y_t	2
1	$r_X = r \in rpc_X$	$p = \text{top}(X_t)$	$X_t \backslash \{p\}$	$r_Y = r \in rpc_Y$	$q = \text{top}(Y_t)$	$Y_t \backslash \{q\}$	0
					$q \neq \text{top}(Y_t)$	Y_t	2
		$p \neq \text{top}(X_t)$	X_t		*	Y_t	2
	$r_X = p \in rpc_X$	$p \neq \text{top}(X_t)$	X_t	$r_Y = q \in rpc_Y$	*	Y_t	2
	$r_X = q \notin rpc_X$	*	$X_t \cup \{q\}$	$r_Y = p \notin rpc_Y$	*	$Y_t \cup \{p\}$	0
	$r_X = s \notin rpc_X$	*	$X_t \cup \{s\}$	$r_Y = s \notin rpc_Y$	*	$Y_t \cup \{s\}$	2

For brevity, we simply use the symbol ' * ' as a 'do not care' indicator

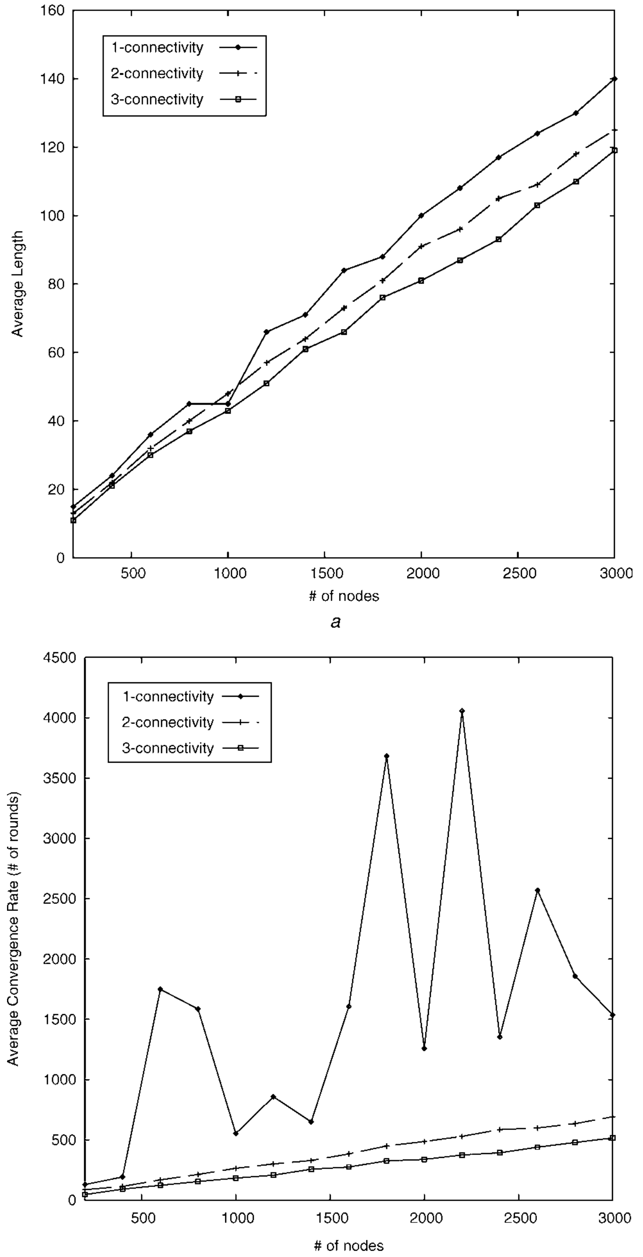


Fig. 5 *Cases where nodes launched SRSO in a d-connectivity network (d = 1, 2, 3)*

a Average length
b Average mixing time

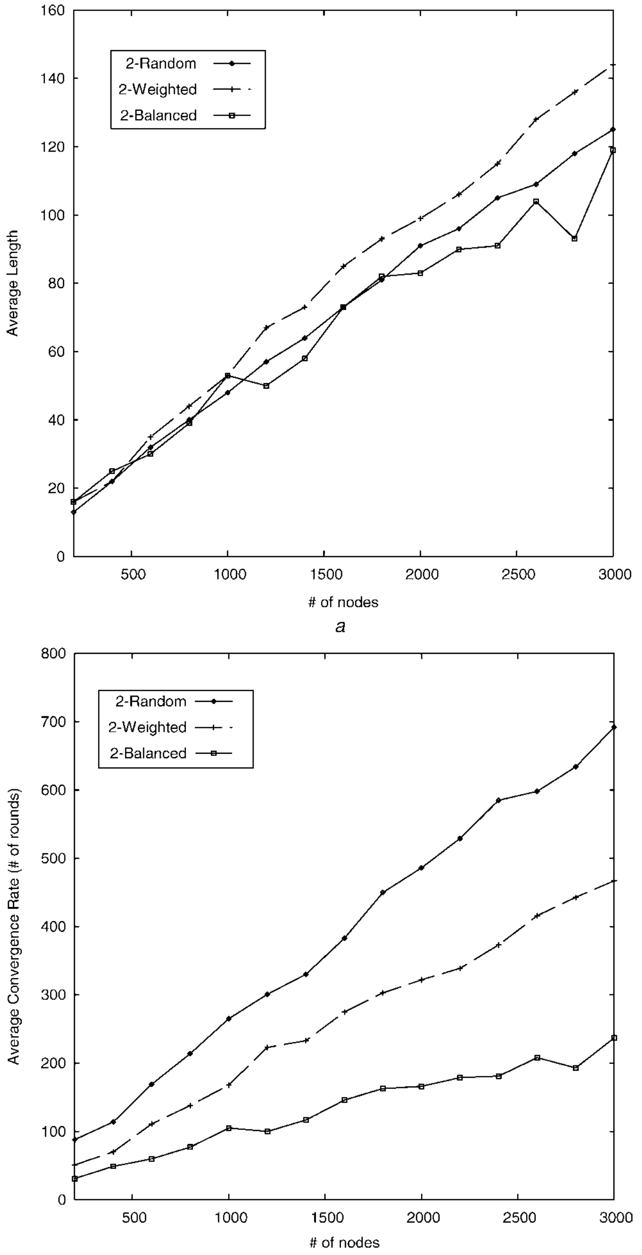


Fig. 6 *Cases where nodes launched SRSO in a 2-connectivity network under different selecting functions*

a Average length
b Average mixing time

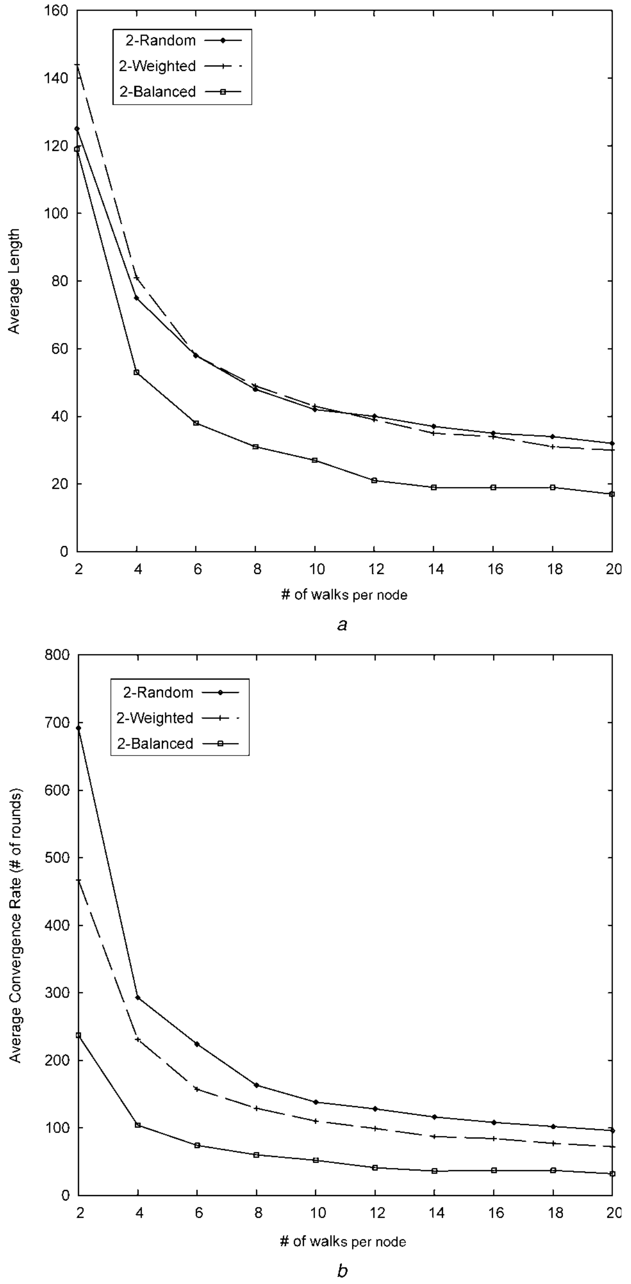
multiple walks to repair a disrupted route under different functions. Clearly, $\bar{L}$ and $\bar{t}$ decrease exponentially as the number of walks increases. Specifically, if $\bar{L} \propto e^{d_1/\kappa}$ and $\bar{t} \propto e^{d_2/\kappa}$, where d_1 and d_2 are constants, we derive a similar result to that in [9] for a random network. For example, if the length limit of only one walk is $\bar{L}$, for a case involving κ walks, the length limit $\bar{L}_\kappa$ of each walk can be presented as $\bar{L}_\kappa = e^{d_1/\kappa} \cdot \bar{L}$. However, the power consumption of κ walks also increases to $\kappa \cdot \bar{L}_\kappa$, that is, the total power consumption of κ walks is proportional to $\kappa \cdot e^{d_1/\kappa}$. Reviewing the results from the previous section, we can determine that the number of packet transmissions involved in multiple walks is proportional to $\kappa \cdot e^{d_1/\kappa}$.

5 Extension and discussion

5.1 Multiple-repair of multiple objects

Consider a case where several queries are issued almost concurrently such that there may be a number of disrupted routes. We call the corresponding repair mechanism Multiple-repairs of multiple objects (MRMO). Intuitively, MRMO can be achieved by performing SRSOs independently, one for each query. However, several refinements can be adopted to improve the overall performance. In this section, we outline some design principles and challenges of MRMO.

When multiple walks search for the same object in a network, they work in a ‘collaborative’ manner. If two probing nodes collide, two possible adjustments can be made: (i) the shorter walk takes over the tasks of the longer walk or (ii) the walks communicate, and one probing node changes course and modifies its f_{tw} in order to avoid the region in question. The former adjustment reduces the communication cost, whereas the latter increases the effective coverage of one of the walks. Unfortunately, these options offset each other. Except for situations involving direct collisions, the probing node



a

b

Fig. 7 *Cases where nodes launched multiple walks in a 2-connectivity network under different conditions ($n = 3000$, $\kappa = 2, 4, \ldots, 20$)*

a Average length
b Average mixing time

may find a node that has been traversed by other walks searching for the same object. In this case, the probing node has to make its own decision based on the potential cost it could incur. Specifically, if the length of the current walk is longer than that of the crossing walk, the probing node stops searching; otherwise, it continues walking. This rule is based on the concept that the longer walk should join the shorter one and thereby reduce power consumption.

Although these adjustments improve the performance of MRMO, target locations might vary over time due to topology changes or obsolete objects. Thus, collaborating walks sometimes become invalid and some information updates will be induced, especially when queries occur less frequently compared with network's dynamics. One possible solution is to maintain a highly adaptive P2P overlay for wireless *ad hoc* networks. However, more implementation details are needed before this can be achieved.

5.2 Discussion

For real-world distributed systems, randomised algorithms are appealing because of their simplicity and elegance. Probabilistic analysis of these algorithms, however, tends to be very complicated [22]. In random-based schemes, the convergence rate is a major concern. As noted in [20], two questions immediately arise: (i) How do we refine the sampling of legal configurations in spite of the possible intricate state distribution? (ii) How do we achieve a tight upper bound of the convergence rate? In principle, the upper bound of the mixing time can be further enhanced with smarter path coupling, such as a better design of δ, or more careful consideration of the movements over the configurations in Ω. An intuitive heuristic might require f_{tw} to exclude the additional nodes of every clique in G, because they do not contribute to the shortest path between any pair of nodes. Alternatively, if some specific geometric structures are introduced, the mixing time can be further reduced over the smaller topology. For example, a distributed algorithm that limits the degree of each node to a constant 7 is proposed in [23]. In this case, the mixing time can be more tightly bound due to a smaller β. Unfortunately, these techniques tend to complicate the subsequent analysis (e.g. see [24]). This is one reason that we have not conducted a deeper analysis of the β parameter in this paper.

Finally, some practical implications should be considered. As mentioned earlier, the concepts involved in a cross-layer design exploit interlayer dependencies and optimise overall performance. For instance, maintaining an up-to-date neighbourhood record is very important in the SRSO scheme. Such records are usually updated by periodic beacon messages. However, as the network's status is provided by continuous cross-layer interactions, the scheme can directly access more QoS information (e.g. the packet delivery/loss ratio and the channel state information) in the physical layer [25]. Accordingly, SRSO can progress more smoothly with higher quality links. Moreover, in an ideal case, SRSO operates with bit-wise data transmission, instead of traditional packets, and thereby avoids tedious packet manipulation operations. Therefore the overall performance is apt to improve.

In wireless *ad hoc* networks, the communication and computation overheads may contribute to a loss of data fidelity [26]. The efficiency of the proposed scheme also depends on the associated data replication strategies and the rate at which the nodes lose interest in searching for certain objects. Techniques to optimise the balance between different network layers have yet to be investigated. On the other hand, selecting an appropriate timeout mechanism depends on the criterion of upper-level applications and must correlate with the network asynchronously. The following is a summary of the most important open questions related to this work. We believe that the questions also apply to many other randomised distributed algorithms used in *ad hoc* networking.

- How can the proposed scheme be expanded so that it is self-stabilising, particularly in a scenario involving multiple-repairs of multiple objects?
- Is a more sophisticated design of f_{tw} and L possible, particularly one that leads to simpler performance analysis and a tighter bound on the mixing time?
- Is there an alternative to the simple timeout mechanism?

6 Conclusion and future work

Wireless *ad hoc* networks are comprised mobile devices with limited computation and communication capacity. It is well known that upgrading the performance and scalability of P2P systems require the adoption of cross-layer design concepts. The key challenge in the design process is coping with scarce resources and network dynamics. This paper addresses the problem of recovering disrupted routes when object-querying is performed in wireless *ad hoc* networks, with the emphasis on P2P data communications. We propose a repair scheme based on self-avoiding walks. As the repair procedure only needs local information and can be performed in a distributed manner, it is robust and scalable in a wireless environment. Furthermore, we present a probabilistic performance analysis to clarify the connection of certain system parameters and examine the expected memory and communication requirements for the proposed scheme. We also show that the state transition dynamics correspond to some rapidly mixing Markov chains and provide a loose upper bound. Finally, simulation results are presented and several potential challenges are discussed. We believe that the distributed, randomised nature of our method makes it robust against several types of adverse behaviour associated with network dynamics.

7 Acknowledgment

This research was supported by the National Science Council of Taiwan, Excellent Research Projects of National Taiwan University, and the Ministry of Economic Affairs under Grants NSC-94-2213-E-002-041, 95R0062-AE00-05, and 95-EC-17-A-02-S1-049, respectively.

8 References

1 Agrawal, D.P., Lu, M., Keener, T.C., Done, M., and Kimar, V.: 'Environmental Monitoring Using Wireless Sensors', *EM Mag.*, (Air & Waste Manage. Assoc., USA, 2004), pp. 27–33

2 Valera, A.C., Seah, W.K.G., and Rao, S.V.: 'Improving protocol robustness in *ad hoc* networks through cooperative packet caching and shortest multipath routing', *IEEE Trans. Mobile Comput.*, 2005, **4**, (5), pp. 443–457

3 Deshpande, A., Guestrin, C., and Madden, S.R.: 'Model-driven data acquisition in sensor networks'. Proc. Int. Conf. Very Large Databases (VLDB), Toronto, Canada, August 2004, pp. 588–599

4 Ramanathan, R.: 'Challenges: a radically new architecture for next generation mobile *ad hoc* networks'. Proc. ACM MobiCom, Cologne, Germany, August 2005, pp. 132–139

5 Sohrabi, K., Gao, J., Ailawadhi, V., and Pottie, G.J.: 'Protocols for self-organization of a wireless sensor network', *IEEE Personal Commun.*, 2000, **7**, (5), pp. 16–27

6 Barbosa e Oliveira, L., Siqueira, I.G., and Loureiro, A.A.F.: 'Evaluation of *ad hoc* routing protocols under a peer-to-peer application'. Proc. IEEE Wireless Commun. and Networking Conf. (WCNC), New Orleans, LA, USA, March 2003, pp. 1143–1148

7 Yang, S.-J.: 'Exploring complex networks by walking on them', *Phys. Rev. E*, (American Physical Doc., USA, 2005), **71**, 016107

8 Braginsky, D., and Estrin, D.: 'Rumor routing algorithm for sensor networks'. Proc. 1st Workshop on Sensor Netw. Appl. (WSNA), September 2002, pp. 22–31

9 Shakkottai, S.: 'Asymptotics of query strategies over a sensor network'. Proc. IEEE INFOCOM, March 2004, pp. 557–566

10 Bawa, M., Garcia-Molina, H., Gionis, A., and Motwani, R.: 'The price of validity in dynamic networks'. Proc. ACM SIGMOD, Paris, France, June 2004, pp. 515–526

11 Kothapalli, K., Scheideler, C., Onus, M., and Richa, A.W.: 'Constant density spanners for wireless *ad hoc* networks'. Proc. ACM SPAA, Las Vegas, Nevada, USA, 2005, pp. 116–125

12 Franciscani, F.P., Vasconcelos, M.A., Couto, R.P., and Loureiro, A.A.F.: '(Re)configuration algorithms for peer-to-peer over *ad hoc* networks', *J. Parallel Distrib. Comput.*, 2005, **65**, (2), pp. 234–245

13 Avin, C., and Brito, C.: 'Efficient and robust query processing in dynamic environments using random walk techniques'. Proc. IEEE/ACM IPSN, April 2004, pp. 396–404

14 Larralde, H., Trunfio, P., Havlin, S., Stanley, H., and Weiss, G.: 'Number of distinct sites visited by N random walkers', *Phys. Rev. A*, (American Physical Doc., USA, 1992), **45**

15 Feeney, L.M., and Nilsson, M.: 'Investigating the energy consumption of a wireless network interface in an *ad hoc* networking environment'. Proc. IEEE INFOCOM, Ankorage, Alaska, 2001, pp. 1548–1557

16 Zhang, X., and Maxemchuk, N.F.: 'A generalized energy consumption analysis in multihop wireless networks'. Proc. IEEE Wireless Commun. and Networking Conf. (WCNC), 2004, vol. 5, no. 1, pp. 1464–1469

17 Bubley, R., and Dyer, M.: 'Path-coupling: a technique for proving rapid mixing in Markov chains'. Proc. IEEE FOCS, 1997, pp. 223–231

18 Fribourg, L., Messika, S., and Picaronny, C.: 'Coupling and self-stabilization', *Distrib. Comput.*, 2006, **18**, (3), pp. 221–232

19 Guruswami, V.: 'Rapidly mixing markov chains: a comparison of techniques', May 2000, available at http:\\cs.washington.edu\homes\venkat\pubs\papers.html

20 Randall, D.: 'Mixing'. Proc. IEEE FOCS, 2003

21 Wan, P.-J., and Yi, C.-W.: 'Asymptotic critical transmission radius and critical neighbour number for k-connectivity in wireless *ad hoc* networks'. Proc. ACM MobiHoc, 2004, pp. 1–8

22 Norman, G.: 'Analysing randomized distributed algorithms', Validation of stochastic systems: a guide to current research, Lecture Notes in Computer Science, (Springer, Berlin/Heidelberg, 2004), vol. 2925, pp. 384–418

23 Li, X.-Y., Stojmenovic, I., and Wang, Y.: 'Partial delaunay triangulation and degree limited localized bluetooth multihop scatternet formation', *IEEE Trans. Parallal Distrib. Syst.*, 2004, **15**, (4), pp. 350–361

24 Hayes, T.P., and Vigoda, E.: 'Variable length path coupling'. Proc. ACM-SIAM SODA, 2004

25 Conti, M., Maselli, G., Turi, G., and Giodano, S.: 'Cross-layering in mobile *ad hoc* network design', *IEEE Comput.*, 2004, **37**, (2), pp. 48–51

26 Shah, S., Ramamritham, K., and Shenoy, P.J.: 'Resilient and coherence preserving dissemination of dynamic data using cooperating peers', *IEEE Trans. Knowl. Data Eng.*, 2004, **16**, (7), pp. 799–812