

CORRECTION EXERCICE PAIR-À-PAIR ET PETITS MONDES
(partiel MPRI 21/11/06)

Réponse 1. Si u ne possède pas la clé, i.e., si $c \notin C(u)$, alors u transmet la requête de recherche à son successeur u^+ . Si $c \notin C(u^+)$ alors u^+ retransmet la requête à son successeur. La recherche se poursuit ainsi, de successeur en successeur. Elle aboutit nécessairement car, d'une part, si u_1, \dots, u_n sont les n utilisateurs courants, alors $\cup_{i=1}^n C(u_i) = \{c_0, \dots, c_{k-1}\}$, et, d'autre part, tous les utilisateurs sont potentiellement visités par cette recherche. Au pire, tous les utilisateurs sont visités, ce qui implique au plus $n - 1$ communications. Ce pire cas est atteint par exemple lorsque la clé c recherchée par l'utilisateur u vérifie $c = c_i$ avec $i = p(u^-)$.

Réponse 2. Soit $B = \{i - d/2, \dots, i - 1, i\}$. Supposons que l'index j choisi par u est dans B . Si le contact v de u vérifie $c_i \notin C(v)$, alors $p(v) \in \{j, j + 1, \dots, j'\}$ avec $j \leq j' < i$, et en ce cas $i \leq p(v) + d/2$. On a donc $p \geq \text{Prob}(j \in B)$. Or $\text{Prob}(j \in B) \geq |B| \cdot p_d \geq \frac{d}{2} \cdot \frac{1}{d \cdot H(k)} = \frac{1}{2H(k)}$. On a donc bien $p \geq \frac{1}{2H(k)}$.

Réponse 3. Soit x l'utilisateur courant. Initialement $x = u$. On définit la distance de x à la clé cherchée $c = c_i$ par $\text{dist}(x, c) = d$ si $i = p(x) + d$. Etant donné x à distance d de la clé, alors, avec une probabilité au moins $\frac{1}{2H(k)}$, le contact y de x vérifie $c \in C(y)$ ou bien il est à distance au plus $d/2$ de la clé. Si cet événement ne se produit pas, alors de deux choses l'une : si y est à distance inférieure de la clé que le successeur x^+ de x alors la requête est transmise à y ; sinon elle est transmise à x^+ . Dans les deux cas, l'utilisateur $x' \in \{x^+, y\}$ recevant la requête de x est plus proche de la clé que x . Donc, comme le choix de l'index est inversement proportionnel à la distance, on obtient qu'avec une probabilité au moins $\frac{1}{2H(k)}$, le contact de x' possède c ou il est à distance au plus $d/2$ de la clé. L'événement \mathcal{E} défini comme

$\mathcal{E} = \text{"le contact de l'utilisateur courant possède } c \text{ ou il est à distance au plus } d/2 \text{ de la clé"}$

a donc lieu avec une probabilité au moins $\frac{1}{2H(k)}$ tout au long du routage à partir de x . L'espérance du nombre d'étapes pour que l'événement \mathcal{E} ait lieu est donc au plus $2H(k)$. Ainsi, diminuer d'un facteur 2 la distance entre l'utilisateur courant et la clé cherchée nécessite une moyenne d'au plus $2H(k)$ étapes. On peut réitérer la même raisonnement pour diviser la distance d'un facteur 2 derechef car le routage de la requête ne visite chaque utilisateur qu'une fois au plus, et par conséquent tous les événements considérés dans l'analyse sont indépendants. Comme la distance initiale entre l'utilisateur ayant émis la requête et la clé est au plus k , on obtient que le nombre moyen d'étapes pour trouver la clé cherchée est au plus $2H(k) \cdot \lceil \log_2 k \rceil$. Comme $H(k) \sim \ln k$, on obtient bien la borne $O(\log^2 k)$.

Réponse 4. Soit $r = k/n$. Chaque utilisateur sélectionne un unique index $i \in \{1, \dots, n\}$ où la probabilité de choisir l'index i est $p_i = \frac{1}{i \cdot H(n)}$. Soit i l'index choisit par l'utilisateur u et soit v l'utilisateur tel que $p(u) + i \cdot r \in C(v)$. L'utilisateur u se connecte alors à v . La même analyse qu'à la questions 2 permet de montrer que si la clé c recherchée vérifie $c = c_i$ avec $i \leq p(u) + dr$ pour $n > d \geq 1$, alors la probabilité p que l'utilisateur u ait son contact v qui vérifie $c \in C(v)$ ou $i \leq p(v) + dr/2$ est au moins $\frac{1}{2H(n)}$. La borne $O(\log^2 n)$ s'obtient alors selon le même raisonnement qu'à la question 3, et en utilisant le fait que lorsque $d = 0$ un nombre d'étapes constant en moyenne suffit à trouver la clé puisque le nombre moyen de clés gérées par chaque utilisateur est $r = k/n$.

Réponse question subsidiaire. Simplement, chaque utilisateur u tire q clés $\gamma_1, \dots, \gamma_q$ aléatoirement, uniformément dans $\{0, \dots, k-1\}$. Pour chacune des clés tirées, il contacte l'utilisateur gérant cette clé et lui demande le nombre de clés qu'il gère. Soit u_i tel que $\gamma_i \in C(u_i)$. L'utilisateur u peut donc calculer $x_q = \frac{1}{q} \sum_{i=1}^q |C(u_i)|$. La loi des grands nombre assure que x_q tend vers l'espérance de la taille des intervalles de clés associée aux utilisateurs, soit k/n , lorsque $q \rightarrow +\infty$. Bien sûr il y a un biais en pratique pour les petites valeurs de q , et le tirage du contact à un instant donné peut être remis en cause si le nombre d'utilisateurs varie beaucoup. Pour plus de précisions, vous pouvez consulter l'article de G. Manku, M. Bawa et P. Raghavan, "Symphony : Distributed hashing in a small world", in Proc. 4th Symposium on Internet Technologies and Systems, 2003.