Formal Methods: Analyzing P2P Routing

Stefanie Roos, Thorsten Strufe

P2P Lecture, 19.12.2012

1 Motivation

Knowing the exact distribution of the routing length is important for setting timeouts and estimating message overhead. When building a system, it is important to compare the expected routing performance of various solutions, before implementing the complete system architecture. This can be done by simulations, however, simulation cannot cover the whole parameter space and large-scale simulation is quite costly. Hence, when possible, formal methods, requiring few resources and being potentially fast, should be used to analyze the solution candidates beforehand.

As for structured P2P systems, the original papers provide claims on the asymptotic complexity, but this is only of limited help. To actually compare different systems, it has to be possible to compute a tight bound on both the expected number of steps as well as the routing step distribution, i.e. the probability that routing takes exactly k steps for all $k \in N$. The latter is especially important when setting timeouts. For example, queries should be repeated after a certain time, when one can be relatively sure, that the first query failed due to message loss. Such a timeout can be chosen as the 0.99 quantile of the routing length distribution, given that this can be easily calculated.

In the following, Markov Chains are introduced, being a simple yet effective tool to model DHT routing. We consider the case of Chord routing as an example.

2 Markov Chains

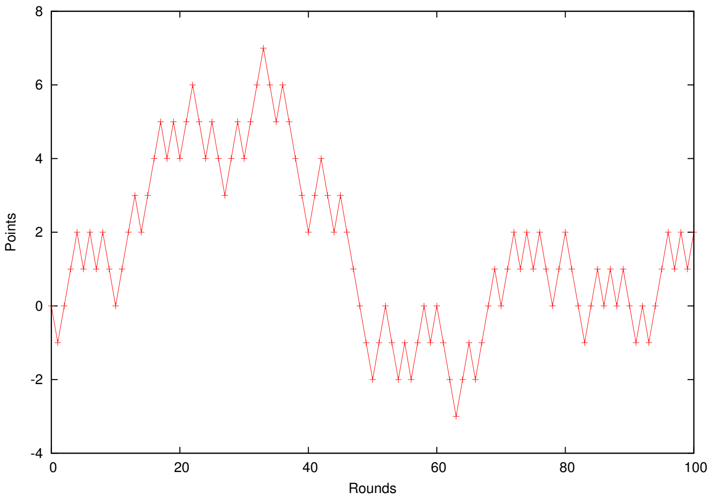

Figure 1: Random Walk with p=0.5

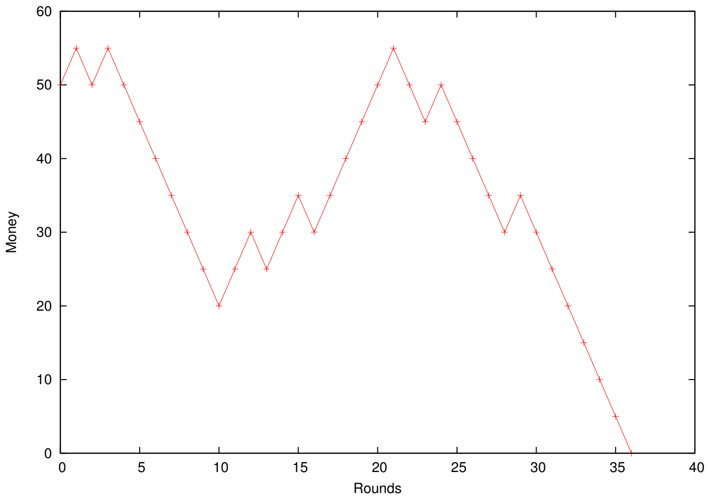

Figure 2: Playing Roulette

We model DHT routing as a process, whose next state depends purely on the current state. Before defining this concept formally, some typical examples:

Random Walk: Consider a game played in identical independent rounds. In each round, you win a point with probability p, otherwise you lose one point. Negative scores are possible. The total number of points in each state can be calculated as the points from the step before plus the points gained in this step. Knowing that you will play for n rounds, you can use Markov Chains to compute the expected gain, as well as the probability to gain at least/at most/exactly k points. One possibility for a game of 100 rounds with success probability $p = 0.5$ is displayed in Figure 1.

Ruin Probability: Imagine you are going to the casino, having 50 Euro. You play roulette betting 5 Euro in each turn. If you win you get 10 Euro back (so a gain of 5 Euro). You stop if you are either out of money or have 100 Euros. The amount of money after each round depends purely on the money you had before, and the result of this round. Markov Chain modeling is used to calculate the probability to lose all your money, as well as the expected number of rounds you can play. See Figure 2 for one possible outcome.

So, how to formalize this intuition of 'depending only on the last state and some randomness'?
Recall that for two events A and B with probability $P(B) > 0$, the *conditional probability* of A given that the event B occurred is

$$P(A|B) = \frac{P(A,B)}{P(B)}. \tag{1}$$

A *Markov Chain* is a sequence of random variables (also called a *random process*) $X_0, X_1, X_2, ...$, so that for all $k > 0$ and states $s_{i_0}, ..., s_{i_k}$

$$P(X_k = s_{i_k}|X_{k-1} = s_{i_{k-1}}, ..., X_0 = s_{i_0}) = P(X_k = s_{i_k}|X_{k-1} = s_{i_{k-1}}). \tag{2}$$

In words, this definition says that the distribution of the $k-th$ step given all earlier steps is the same as the distribution given only the step before. Hence, the earlier states do not have to be considered if the step right before is known. It can be derived that for all $k > 0$ and states s_i, s_j:

$$P(X_k = s_i|X_{k-1} = s_j) = P(X_1 = s_i|X_0 = s_j). \tag{3}$$

We call $p_{ij} = P(X_1 = s_i|X_0 = s_j)$ the *transition probabilities*.

So, now, how do we calculate the distribution after the k-th step?
Consider an example: Assume that you decide to attend the P2P exercise merely based on your experience in the last exercise. If you did not attend the last exercise, the probability of going to the next one is 0.2. If you attended the exercise, you come back with probability 0.7. Furthermore, assume you go to the first exercise with probability 0.9. This is a process $X_0, X_1, X_2, ...$, where each X_i is in $S = \{s_0, s_1\} = \{notattend, attend\}$. A graphical representation of the states and the *transition probabilities* is displayed in Figure 3.

How likely is it that you come to the second class? How likely is it that you don't come? In other words, find the distribution of X_1.

The probability that you don't attend the second class is calculated by summarizing over the two possibilities for the first exercise:

$$P(X_1 = s_0) = P(X_1 = s_0|X_0 = s_0)P(X_0 = s_0) + P(X_1 = s_0|X_0 = s_1)P(X_0 = s_1) = 0.8 * 0.1 + 0.3 * 0.9 = 0.35. \tag{4}$$

Analogously, the probability for attending the class can be computed:

$$P(X_1 = s_1) = P(X_1 = s_1|X_0 = s_0)P(X_0 = s_0) + P(X_1 = s_1|X_0 = s_1)P(X_0 = s_1) = 0.2 * 0.1 + 0.7 * 0.9 = 0.65. \tag{5}$$

Eq. 4 and 5 can be written as a matrix multiplication:

$$\begin{pmatrix} P(X_1 = s_0) \\ P(X_1 = s_1) \end{pmatrix} = \begin{pmatrix} P(X_1 = s_0|X_0 = s_0) & P(X_1 = s_0|X_0 = s_1) \\ P(X_1 = s_1|X_0 = s_0) & P(X_1 = s_1|X_0 = s_1) \end{pmatrix} \begin{pmatrix} P(X_0 = s_0) \\ P(X_0 = s_1) \end{pmatrix} \tag{6}$$

If you now want to calculate the probability for attending class the third week, one can calculate this from the distribution for the second week:

$$\begin{pmatrix} P(X_2 = s_0) \\ P(X_2 = s_1) \end{pmatrix} = \begin{pmatrix} P(X_2 = s_0|X_1 = s_0) & P(X_2 = s_0|X_1 = s_1) \\ P(X_2 = s_1|X_1 = s_0) & P(X_2 = s_1|X_1 = s_1) \end{pmatrix} \begin{pmatrix} P(X_1 = s_0) \\ P(X_1 = s_1) \end{pmatrix} \tag{7}$$

By Eq. 3, we have $P(X_2 = s_i|X_1 = s_j) = P(X_1 = s_i|X_0 = s_j)$, we can write the above as:

$$\begin{pmatrix} P(X_2 = s_0) \\ P(X_2 = s_1) \end{pmatrix} = \begin{pmatrix} P(X_1 = s_0|X_0 = s_0) & P(X_1 = s_0|X_0 = s_1) \\ P(X_1 = s_1|X_0 = s_0) & P(X_1 = s_1|X_0 = s_1) \end{pmatrix} \begin{pmatrix} P(X_1 = s_0) \\ P(X_1 = s_1) \end{pmatrix} \tag{8}$$

Substituting $(P(X_1 = s_0), P(X_1 = s_1))^t$ by Eq. 6, we get

$$\begin{pmatrix} P(X_2 = s_0) \\ P(X_2 = s_1) \end{pmatrix} = \begin{pmatrix} P(X_1 = s_0|X_0 = s_0) & P(X_1 = s_0|X_0 = s_1) \\ P(X_1 = s_1|X_0 = s_0) & P(X_1 = s_1|X_0 = s_1) \end{pmatrix}^2 \begin{pmatrix} P(X_0 = s_0) \\ P(X_0 = s_1) \end{pmatrix} \tag{9}$$

In general, assume you have a Markov Chain $X_0, X_1, X_2, ...$ with state space $S = \{s_1, \ldots, s_m\}$. You can calculate the distribution

$$P_k = (P(X_k = s_1), \ldots, P(X_k = s_m))^t$$

after the k-th step from the initial distribution

$$I = (P(X_0 = s_1), \ldots, P(X_0 = s_m))^t$$

and the transition matrix T with entries

$$t_{ij} = P(X_1 = s_i|X_0 = s_j)$$

by simple matrix multiplication:

$$P_k = T^k I \tag{10}$$

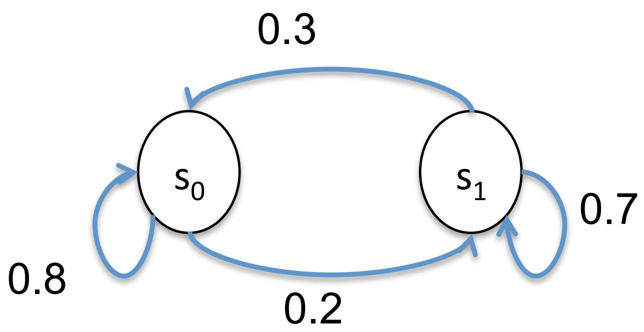

Figure 3: Exercise Attendance: States and Transitions

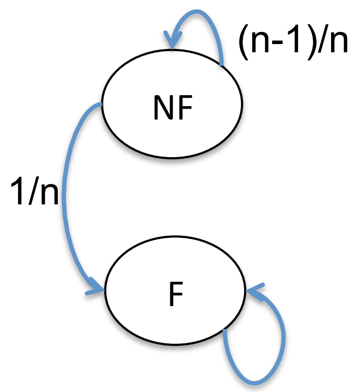

Figure 4: Neighbor Queries: States and Transitions

3 Markov Chains for Routing

Sensible routing has to terminate, either by failure, triggered by a timeout or the information that the desired content is not available, or by success. We say that a Markov Chain has a *terminal state*, i.e., a state s for which

$$P(X_1 = s'|X_0 = s) = \begin{cases} 1, s' = s \\ 0, s' \neq s \end{cases} . \tag{11}$$

One terminal state is always the state F of successfully finding a document. Let $X_0, X_1, X_2, ...$ be a random process for the routing state. To obtain a routing length distribution, we are interested in the fraction of queries that have been successful after k steps. $P(X_k = F)$ gives the probability of needing at most k steps, the probability to need exactly k steps is then computed as $P(X_k = F) - P(X_{k-1} = F)$.

To get a routing distribution for a specific system the following steps are necessary:

1. Define all possible routing states (e.g. completed and not completed)
2. Derive initial distribution I over all states (e.g. initially routing is not completed)
3. Derive transition matrix T
4. Calculate $P_k = T^k I$ for all desired k (e.g. until routing is completed with probability 1)

The simplest case for which Markov Chains can be considered is routing by asking one random node in a set of n nodes in each step. In this case, the state space $S = \{F, NF\}$, where F stands for having found the target, and NF for not having found the target. If only one of these n nodes is the target, the target is found with probability $1/n$ in the next step. See Figure 4 for the graphical representation of the transitions. Initially, the target is not found, so we have

$$I = (P(X_0 = F), P(X_0 = NF))^t = (0, 1)^t \tag{12}$$

The transition matrix T is given by

$$T = \begin{pmatrix} 1 & \frac{1}{n} \\ 0 & \frac{n-1}{n} \end{pmatrix} \tag{13}$$

4 Chord Routing

In the following, we model the routing in a Chord implementation used for instant messaging. Hence, rather than searching for file identifiers, nodes are searching for the identifiers of other nodes.

When joining the network, a node selects an identifier uniformly at random. Recall that in a b-bit Chord, each node has a link to its successor node, i.e. the next node on a ring of length 2^b. Additionally, each node has $b-1$ fingers. The target ID of finger i of node v is $ID(v) + 2^i \bmod 2^b$. The routing table entry then corresponds to the the successor of the target ID. See Figure 5 for an illustration.

For obtaining the routing length distribution, we first select a suitable state space, then determine the initial distribution and the transition probabilities. All results depend on the number of bits b as well as the number of nodes n in the system. Furthermore, we assume that the node with the destination ID actually exists, and all links are correct, so the routing cannot fail.

During routing from a source u to a destination, the choice of the next hop depends on the distance to the target and a random factor introduced by the random selection of IDs. Hence the state space consists of all possible distances $0, \dots, 2^b - 1$ to the destination, where a distance of 0 means that the routing is successful.

The sender node picks an identifier uniformly at random, but the identifier of the destination. Hence, initially all non-zero distances are equally likely. Hence the initial distribution is:

$$I = \left(P(X_0 = 0), P(X_0 = 1), \dots, P(X_0 = 2^b - 1)\right)^t = \left(0, \frac{1}{2^b - 1}, \dots, \frac{1}{2^b - 1}\right)^t \in \mathbb{R}^{2^b} \tag{14}$$

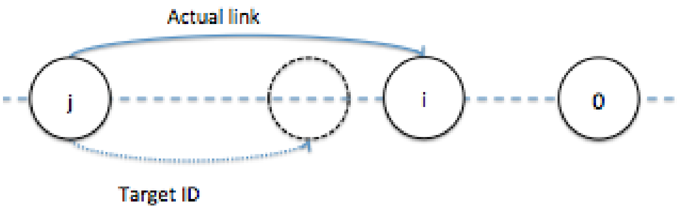

Figure 5: Routing in Chord: Node j has a finger to a certain target ID. Since no node has this identifier, the link goes to the next node i on the ring.

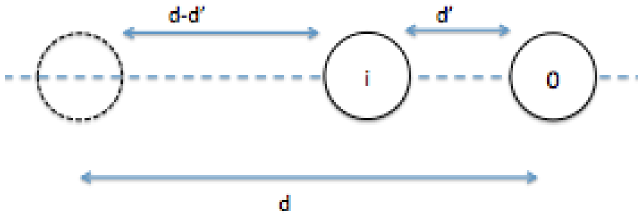

Figure 6: Distances: The target ID at distance d to destination t might not be taken by any node. The probability that the link points to node at distance d' corresponds to the probability that exactly $d - d'$ successive IDs do not correspond to any node.

It remains to derive the transition probabilities $t_{ij} = P(X_1 = i | X_0 = j)$, the probability to go from distance j to distance i. In case, the destination is already found, the routing remains in that state, so

$$t_{i0} = \begin{cases} 1, & i = 0 \\ 0 & i > 0. \end{cases} \tag{15}$$

From now on, assume the distance j of the last node is non-zero. A node at distance j has links with target IDs $j - 2^0 \mod 2^b$, $j - 2^1 \mod 2^b$, $\dots j - 2^{b-1} \mod 2^b$. The message is forwarded to the link with the target ID closest to the destination, i.e. the c-th finger with $c = \lfloor log_2 j \rfloor$ such that $j - 2^c \geq 0$ and $j - 2^{c+1} < 0$.

Hence, nodes at a higher distance than $j - 2^c$ are not contacted, so $t_{ij} = 0$ for $i > j - 2^c$. The actually contacted node has a distance between 0 and $d := j - 2^c$, and is the successor of the target ID at distance d.

We will need the following result: The probability that at least r consecutive IDs are not taken by any node is

$$\left(\frac{2^b - r}{2^b}\right)^n, \tag{16}$$

assuming that n nodes choose IDs in a b-bit space uniformly at random and independently. Eq. 16 holds because each node has to select one of the $2^b - r$ remaining IDs, and the probability of n such independent selections is multiplied.

For the routing, first consider the case that the destination is contacted. Then there are no nodes within distance d of the destination, i.e. all $n - 1$ nodes excluding the destination have chosen an identifier that is not in an interval of length $d + 1$. Knowing that the sender has distance $j > d$, we compute the probability that the remaining $n - 2$ nodes have not selected an identifier in the respective interval [1]. The probability for choosing an identifier outside an interval of length d when having 2^b identifiers is given by $\frac{2^b - d}{2^b}$, i.e. the number of identifiers not in the interval divided by all identifiers. Having $n - 2$ nodes choosing such an identifier independently results in $t_{0j} = \left(\frac{2^b - d}{2^b}\right)^{n-2}$.

If the message is forwarded to a node at distance $d \geq d' > 0$, no node has chosen an identifier in an interval of length $d - d'$, but at least one has chosen an identifier in an interval of length $d - d' + 1$ (see Figure 6). This can be calculated as the difference of the probability of having an interval of length at least $d - d'$ identifiers without any nodes and the probability of having at least $d - d' + 1$ identifiers without any nodes, i.e.

$$t_{d'j} = \left(\frac{2^b - d + d'}{2^b}\right)^{n-2} - \left(\frac{2^b - d + d' - 1}{2^b}\right)^{n-2}. \tag{17}$$

To see this, recall that for any event $A \subset B$, $P(B \setminus A) = P(B) - P(A)$ holds. We choose the event $B = \{$ at least $d - d'$ IDs not taken by a node $\}$ and the event $A = \{$ at least $d - d' + 1$ IDs not taken by a node $\}$ and apply Eq. 16 to get Eq. 17.

[1] At this point using a Markov Chain is only an approximation, because knowing the number of nodes already on the path, the number of nodes that can be in the interval is slightly reduced. However, it can be shown that the difference is negligible.

In summary, we get for $j > 0$ and $d := j - 2^{\lfloor log_2 j \rfloor}$

$$t_{ij} = \begin{cases} 0, \quad i > d \\ \left(\frac{2^b - d + i}{2^b}\right)^{n-2} - \left(\frac{2^b - d + i - 1}{2^b}\right)^{n-2}, \quad 0 < i \leq d \\ \left(\frac{2^b - d}{2^b}\right)^{n-2}, \quad i = 0 \end{cases} \tag{18}$$

Having determined the initial distribution I as well as the transition matrix T, one can now derive the cumulative routing length distribution calculation $P_k = T^k I$ and taking $P_k(0) = P(X_k = 0)$.

5 Results

As an example, we consider a 13-bit ID space and 1000 nodes. Note that taking 128 bits as is suggested for Chord is computationally infeasible, since the memory requirements are $\mathcal{O}(2^{2b})$. However, choosing b such that $2^b >> n$ leads to a very good approximation. We compared the formal analysis with a Chord implementation in GTNA [2], using 1000 nodes and 128 bits. The simulation is averaged over 30 runs. In each run a Chord network is constructed. Then the routing performance is sampled by 5 random queries for each node.

Figure 7 shows that the formal results and the simulation performance are very close. The analytical results are within the min and max of the 30 simulation runs. In general, the predicted routing length are minimally shorter due to the lower number of bits used for the formal analysis.

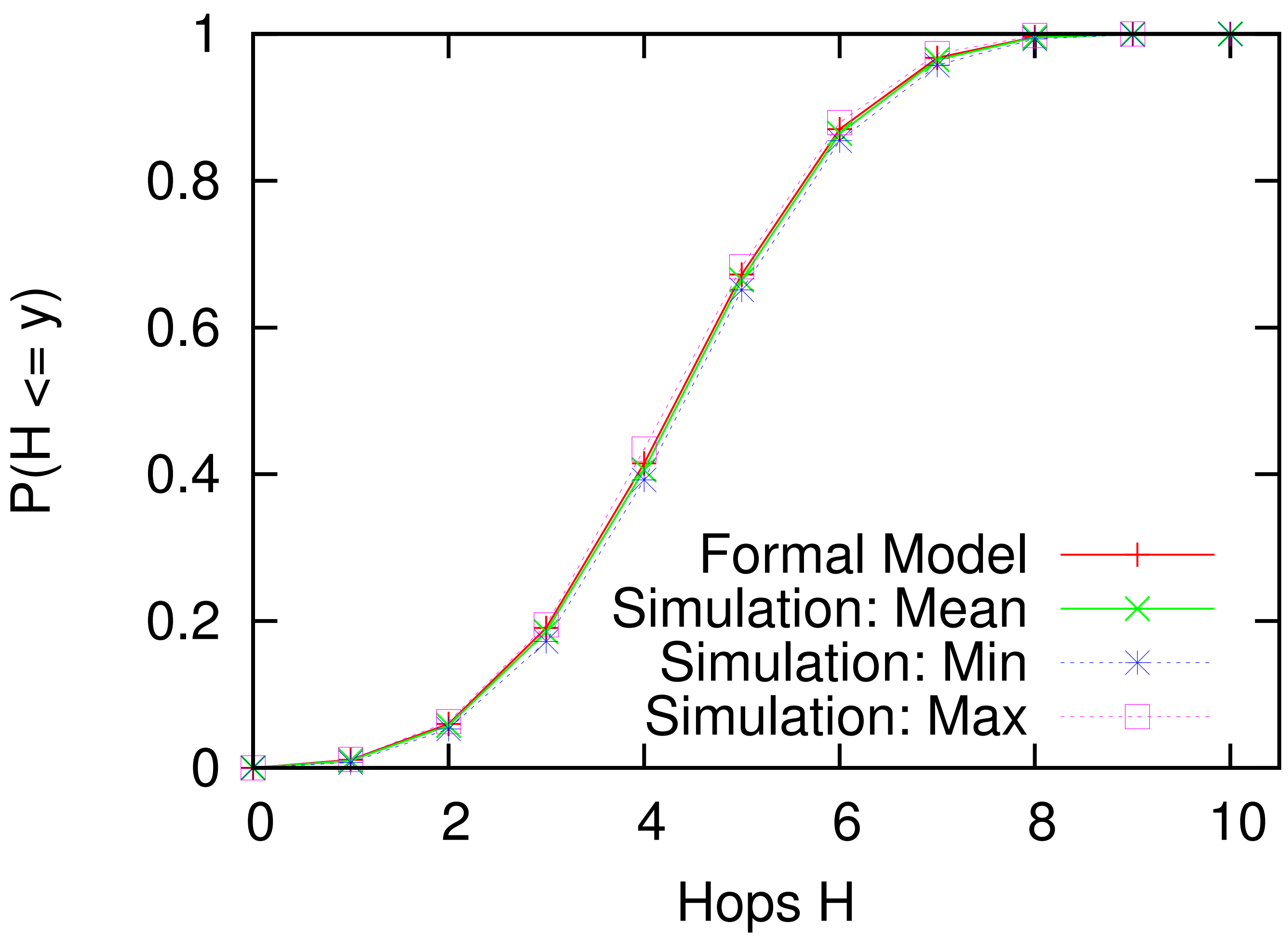

Figure 7: Formal model for 1000 nodes in comparison to simulation

[2]https://github.com/BenjaminSchiller/GTNA