Wader: Weak State Routing Using Decay Bloom Filters

Deke Guo*, Yuan He†, Yunhao Liu†
* College of Information Systems and Management, National University of Defense Technology, Changsha, China
† Department of Computer Science and Engineering, Hong Kong University of Science and Technology

***Abstract*—Weak state routing using decay Bloom filters has been widely studied in the field of data-oriented networks. The existing weak state routing schemes cannot facilitate in-network queries effectively. Given a query for any item at an arbitrary node, the *noise*[1] in an unrelated routing entries is very likely equal to the useful information in the right routing entries. Consequently, queries are routed by means of network flooding, which differs a lot from the desired way of weak state routing, irrespective of the network topology and the usage and decay models of the Bloom filters. This work addresses the root cause of the mismatch between the practically reachable performance of the existing weak state routing schemes and the desired performance. Specifically, we study the impact of decay model on the membership information in the routing entries, and evaluate the negative impact of noise on a routing decision. Based on such analytical results, we derive the necessary and sufficient condition of a feasible weak state routing using decay Bloom filters. Accordingly, we design a novel receiver-oriented approach for Bloom filters, called *Wader*, which satisfies the above condition. The simulation results match well with our theoretical analysis, which demonstrate that *Wader* guarantees the correctness and efficiency of weak state routing with high probability.**

I. Introduction

Bloom filters (bf) are often deemed as a suitable tool to aid information discovery and navigation in multi-hop networks with high query frequency. This includes navigating queries from any node. Bloom filters [1], [2] have been widely employed to realize information-guided routing in overlay networks [3], [4], [5], [6], wireless sensor networks [7], ad hoc networks [8], [9], [10], [11], and mesh networks [12]. The common idea among those proposals is that each node uses a Bloom filter to represent its data items, and broadcasts it to nodes residing with its *propagation range*, e.g. h hops. Each link, associated with all the received Bloom filters through it, is maintained as a routing entry. If a node needs to route a query to a destination residing within h hops away, it forwards the query over the link, which has at least one of its associated Bloom filers satisfy the query.

Those schemes, outperforming the blind routing schemes though, cost large amount of storage space for Bloom filters, and in turn appear inefficient when scanning the Bloom filers to seek a routing decision. The delay of each routing decision is considerably long so that the concurrency of in-network queries is reduced. These two problems become more serious when the average node degree is high and the *propagation range* of Bloom filters are large, especially in resource-constrained networks such as wireless sensor networks. Consequently, such schemes suffer poor efficiency and scalability.

Kumar et al. improve the previous schemes by employing an exponential decay Bloom filters in weak state routing[2] [13]. Bloom filters are still propagated within a specified propagation range, while the amount of information in each Bloom filter decreases exponentially with the distance from the source. In addition, in a routing entry, a link is associated with the union of all the received decay Bloom filters through it. Note that each routing entry does not contain the complete membership information of any item. Hence, a query is sent via the link whose associated routing entry has the maximum amount of information about the queried item. Such a scheme significantly saves storage space and reduces the delay of answering a query than the aforementioned schemes. X. Li et al. in [14] propose a weak state routing scheme using Scope Decay Bloom Filters. A Bloom filter is propagated without any loss within the first h_0-hops from the source, while decays exponentially or linearly outside the h_0-hops from the source.

Essentially in weak state routing, each node as a source creates information gradients in a potential filed. Hints left on nodes on the existence of data items will smoothly guide queries towards desired sources. Thus the information gradients enable efficient local routing by simply ascending the potential field. Ideally, any query will be propelled towards the destination once it enters the propagation range of the destination. In practice, however, given a query for any item at an arbitrary node, we observe that the noise in unrelated routing entries is very likely equal to even larger than the useful information of the item in the right routing entries. Being misled, almost all queries are routing to destinations using the flooding approach. Consequently, the weak state routing schemes usually run in a way that differs a lot from the ideal one and lack feasibility in practice.

In this paper, we study the cause of the mismatch between the ideal and practical performances of the weak state routing schemes, and propose approaches to guarantee the feasibility of those routing schemes with high probability. This basically involves the following design criterion. First, once a query enters the potential filed of the destination, the amount of information in correct routing entries at intermediate nodes should increase as the query are propelled towards the destination. Second, each node should finely control the strength

[1]For a node receiving a query of an item x, *noise* on a link L of the node is defined as the amount of membership information of x in the routing entry corresponding to L, if the node does not receive a decay Bloom filter from the destination of x through L.

[2]The notion of weak state routing used in this work was first proposed in [10], the authors propose the use of decay bloom filters for large scale dynamic networks.

of noise at its out-going links so that it can distinguish right out-going links from others interfering noises. Bearing these points in mind, our contributions are summarized as follows.

1) We model the weak state routing under a general decay model, and measure the strength of useful information in the right routing entries and that of noise in unrelated routing entries. We also analyze and evaluate the negative impact of noise on a weak state routing decision.
2) We derive the necessary and sufficient condition of a feasible weak state routing scheme using decay Bloom filters.
3) Based on this condition, we propose a receiver-oriented optimization approach of Bloom filters, called *Wader*, since the transmitter-oriented approach fails to satisfy the necessary and sufficient condition. We further address the redundant queries in *Wader*:

The rest of this paper is organized as follows. In Section II, we briefly introduce Bloom filters, and measure the effect of the decay model on membership information of any item in a Bloom filter for a linear and an exponential decay model. We further evaluate the impact of noise on routing decisions. In Section III, we derive the necessary and sufficient condition which ensures a feasible weak state routing scheme using decay Bloom filters, and then propose *Wader* to guarantee a feasible weak state routing scheme using Bloom filters. Section IV presents the performance evaluation results. We concludes this work in Section V.

II. Measurement of weak state routing using Bloom filters

In this section, we first analyze weak state routing under a general decay model of Bloom filters in Sections II-A and II-B. To derive the necessary and sufficient condition, as mentioned in Section III-A, for a feasible weak state routing using Bloom filters, we measure the strength of useful information in the right routing entries in Section II-C and the impact of noise on each routing decision in Section II-D.

A. Bloom filters

A set X of n items is represented by a Bloom filter using a vector of m bits which are initially set to 0. A Bloom filter uses k independent hash functions $h_1, h_2, \cdots, h_k$ with a range $\{1, ..., m\}$. When inserting an item x to X, all bits of $Bfaddress(x)$ (consisted of $h_i(x)$ for $1 \leq i \leq k$) will be set to 1. To answer a membership query for any item x, users check whether all bits $h_i(x)$ are set to 1. If not, x is not a member of X. If yes, we assume that x is a member of X, although we might be wrong in some cases. Hence, a Bloom filter may yield a *false positive* which suggests that the item x is in X even though it is not. A false positive is due to hash collisions, in which all bits of $Bfaddress(x)$ were set to 1 by other items in X [1].

Let p_0 be the probability that a random bit of a Bloom filter is 0, and let n be the number of items that have been added to the Bloom filter, then $p_0 = (1 - 1/m)^{n \times k} \approx e^{-n \times k/m}$, as $n \times k$ bits are randomly selected, with probability $1/m$ in the process of adding each item. Now we test membership of an element $x_1 \notin X$. Each of k bits of $Bfaddress(x_1)$ is 1 with a probability as above. The probability of all of k bits being 1, which would cause a false positive, is then

$$f_p = (1 - p_0)^k \approx (1 - e^{-k \times n/m})^k.$$

The minimum value of f_p is achieved when $k = \lfloor (m/n) \ln 2 \rfloor$.

B. Decay model of Bloom filters

As mentioned above, a local bf at each node is forwarded to a few nodes in order to implement a weak state routing scheme. The number of bits set to 1 in the bf decreases with the distance from the source. An accurate decay model of Bloom filters is the dominated factor which affects the correctness and efficiency of the weak state routing schemes. In this paper, we only consider the scenario that each node employs a homogeneous decay model.

Definition 1: Given a set X with n items and its Bloom filter bf, $\theta(x, bf)$ denotes the amount of information in bf for $\forall x \in X$, that is the number of bits being 1 in $Bfaddress(x)$. Let $\theta(X, bf)$ denote the amount of information in bf for X, that is the number of bits set to 1 in bf. Note that $\theta(X, bf)$ is often less than the sum of $\theta(x, bf)$ for each $x \in X$ due to the unavoidable hash collisions between Bloom filter addresses. Let $\theta(bf)$ denote the expectation of number of bits set to 1 in the bf, and equals to m multiply the probability p_1 that a random bit in the bf is set to 1. The p_1 is $1 - (1 - 1/m)^{k \times n}$, and hence

$$\begin{aligned} \theta(bf) &= m \times (1 - (1 - 1/m)^{k \times n}) \\ &\approx m \times (1 - e^{-k \times n/m}) \end{aligned} \tag{1}$$

The value of $\theta(x, bf)$ equals to k for $\forall x \in X$. There are two models to reduce $\theta(x, bf)$ by decaying the bf. In *exponential* model, if a bit in $Bfaddress(x)$ is 1, it remains 1 at a constant probability $1/d$ during each round of decay. In *linear* model, number of d random bits which are 1 in $Bfaddress(x)$ become 0 during each decay. An approximate method to implement the linear decay model is that number of $\lceil \frac{\theta(bf)d}{k} \rceil$ bits which are 1 in the bf are set to 0 during each round of decay. Note that d is a *decay factor* in both models and is a positive real number.

Definition 2: Let $Decay(bf, h, h_0, h_1, model, d)$ denote a general decay model of a bf which is propagated to nodes within h hops from the source where $1 \leq h \leq h_0 + h_1$. The bf does not decay within the first h_0-hops, while decays outside the h_0-hops by using the aforementioned decay models, where h_1 is an upper bound on the hops in the second stage [14].

Definition 3: Let bf_i denote a new Bloom filter resulted from the ith round decay of a bf where $1 \leq i \leq h$. bf_i remains $\theta(bf_i)$ bits set to 1. If the *model* is *exponential*, then

$$\theta(bf_i) = \begin{cases} \theta(bf), & i \leq h_0 \\ \lceil \frac{\theta(bf_{i-1})}{d} \rceil, & h_0 < i \leq h_0 + h_1, d \leq \theta(bf_{i-1}) \\ 0, & h_0 < i \leq h_0 + h_1, \theta(bf_{i-1}) < d \end{cases} \tag{2}$$

If the *model* is *linear*, then

$$\theta(bf_i)=\begin{cases} \theta(bf), & i \le h_0 \\ \theta(bf_{i-1})\text{-}\lceil\frac{\theta(bf)d}{k}\rceil, & h_0 < i \le h_0+h_1 \\ 0, & \theta(bf_{i-1}) < \frac{\theta(bf)d}{k} \end{cases} \quad (3)$$

Let c denote the average node degree in a given network. According to Definition 2, a Bloom filter initialized at an arbitrary node can be transmitted to at most $T_i = c(c-1)^{i-1}$ nodes during the ith round of decay. Due to the symmetry of the model, each node A can receive at most $(c-1)^{i-1}$ decay Bloom filters initialized i hops away through each link $link_j$. The received Bloom filters by node A are recorded as bf_i^l where $1 \le i \le h$ and $1 \le l \le (c-1)^{i-1}$. Thus, the number of decay Bloom filters a node can receive from the whole system through $link_j$ is denoted as $|link_j|$, and

$$|link_j| = \sum_{i=1}^{h}(c-1)^{i-1}.$$

As mentioned in [15], the union of homogeneous Bloom filters can be realized by a logical *or* operation between their bit vectors. Thus, the union of $|link_j|$ decay Bloom filters results in a joint Bloom filter $bf(link_j)$ for a link $link_j$ of node A. The $bf(link_j)$ acts as a probabilistic summary of all items which are reachable from node A along a routing path of at most h hops, and is given by

$$bf(link_j)) \quad = \quad \bigcup_{i=1}^{h}\bigcup_{l=1}^{(c-1)^{i-1}} bf_i^l. \quad (4)$$

Lemma 1: The number of bits set to 1 in any $bf(link_j)$ of each node is given by

$$\theta(bf(link_j)) = m(1-(1-1/m)^{\beta(link_j)}), \quad (5)$$

where

$$\beta(link_j) = \sum_{i=1}^{h}\sum_{l=1}^{(c-1)^{i-1}} \theta(bf_i^l). \quad (6)$$

Proof: Recall that $|link_j|$ decay Bloom filters received by a node through $link_j$ will be merged to construct $bf_i(link_j)$. During the union process, $\beta(link_j)$ balls are dropped into m bits of $bf(link_j)$ randomly, i.e., the location of each ball is independently and uniformly chosen from m possibilities. $\beta(link_j)$ denotes the total number of bits being 1 in those $|link_j|$ decay Bloom filters. Let p_0 denote the probability that a random bit in $bf(link_j)$ is 0 after dropping all $\beta(link_j)$ balls. Clearly, p_0=$(1-1/m)^{\beta(link_j)}$. Let p_1 denote the probability that a random bit in $bf(link_j)$ is set to 1. Thus, p_1=$1-p_0$. Therefore the number of bits set to 1 in $bf(link_j)$ is given by $\theta(bf(link_j))$=$m(1-(1-1/m)^{\beta(link_j)})$. Thus proved. ■

C. Effect of decay model on membership information

In general, $\theta(x, bf)$=k where an element x is represented by a bf. The general decay model might adopt the exponential or linear decay model. We first measure a metric $\theta(x, bf_i)$ which denotes the number of membership information of x in a decay Bloom filter bf_i.

For the linear decay model, we can derive the following result based on its definition.

$$\theta(x, bf_i) = \begin{cases} \theta(x, bf) = k, & i \le h_0 \\ \theta(x, bf) - d(i-h_0), & h_0 < i \le h_0+h_1 \end{cases} \quad (7)$$

For the exponential decay model, we can draw the following conclusion based on its definition.

Lemma 2: If $i \le h_0$, $\theta(x, bf_i)$ equals to k because bf_i=bf. Otherwise, $\theta(x, bf_i)$ is a discrete random variable, denoted as U_i. Its possible values are integers ranging from 0 to k. The probability mass function of U_i is defined by Formula 8.

Proof: Assume a represents the possible value of U_i, and is an integer ranging from 0 to k. Let U_i=a mean that the amount of bits being 1 in the $Bfaddress(x)$ is a. After i rounds of decay of bf, the number of $\theta(bf)$−$\theta(bf_i)$ bits being 1 in bf are reset to 0 in bf_i. The number of possibilities that outcome bf_i is $\binom{\theta(bf)}{\theta(bf)-\theta(bf_i)}$. The number of possibilities that just k−a bits in $Bfaddress(x)$ are reset to 0 during the i rounds of decay is $\binom{k}{k-a}\binom{\theta(bf)-k}{\theta(bf)-\theta(bf_i)-k+a}$. Then the probability that $\theta(x, bf_i) = a$ is given by

$$P(U_i = a) = \frac{\binom{k}{k-a}\binom{\theta(bf)-k}{\theta(bf)-\theta(bf_i)-k+a}}{\binom{\theta(bf)}{\theta(bf)-\theta(bf_i)}}. \quad (8)$$

$\theta(bf_i)$ is given by Formula 2. Therefore, Lemma 2 holds. ■

Corollary 1: If $h_0 < i \le h_0 + h_1$, the expectation of U_i can been calculated by

$$E[U_i] = \sum_{a=0}^{k} a \times P(U_i = a) \approx k/d^{i-h_0}.$$

If the decay range i exceeds h_0, $\theta(x, bf_i)$ under the linear decay model and the expectation of $\theta(x, bf_i)$ under the exponential decay model decrease with the increasing i. Fig.1 plots an illustrative example of the propagation of a bf from node A. The color of propagation filed becomes light from deep as the decay range increases. This result indicates that the number of membership information of $x \in X$ in bf reduces during the decay transmission of bf.

Practically, a node receiving bf_i through a link $link_j$ also collects other $|link_j|$−1 decay Bloom filters through the same link. As shown in Fig.1, node E receives a decay Bloom filer from nodes A, B, and C through the same link $C \rightarrow E$. Thus, the metric $\theta(x, bf_i)$ fails to support a weak state routing scheme since each node uses the union of all received Bloom filters through a link as a correlated routing entry. To address this issue, we propose a metric $\theta(x, bf_i(link_j))$ which denotes the amount of information of x in a routing entry $bf_i(link_j)$ at the node receiving bf_i through $link_j$ where $1 \le j \le c$.

Before measuring the metric, we first define two events used frequently in the rest of this paper. Given any bit in an empty Bloom filter, an event $E^z_{=i}$ means that the bit is set to i after throwing z balls into the Bloom filter. The probability of $E^z_{=0}$ can be calculated by

$$P(E^z_{=0}) = (1-1/m)^z.$$

The probability of $E^z_{=1}$ is given by

$$P(E^z_{=1}) = 1 - P(E^z_{=0}).$$

Lemma 3: In the context of exponential decay model, the metric $\theta(x, bf_i(link_j))$ is a discrete random variable, denoted as V_i. Its possible values are integers ranging from 0 to k. The probability mass function of V_i is defined by Formula 9.

Proof: In bf_i, let us consider an event $\theta(x, bf_i)$=a that a bits in $Bfaddress(x)$ are set to while other $k-a$ bits are set to 0 where $0 \le a \le k$. The probability of this event is given by Formula 8. To achieve $bf_i(link_j)$, other $|link_j|-1$ decay Bloom filters merge with bf_i based on the union operation of Bloom filters. In other words, number of $\alpha(link_j)$ balls are thrown into bf_i randomly, where $\alpha(link_j)$=$\beta(link_j) - \theta(bf_i)$. Let us consider another event that $\theta(x, bf_i)$=a and there exists b bits in $Bfaddress$ which are 0 in bf_i but are hit after throwing $\alpha(link_j)$ balls into bf_i, where $0 \le b \le k-a$. The probability of this event is denoted as $P(W_i$=$b|U_i$=$a)$, and is

$$\binom{k-a}{b} P\big(E^{\alpha(link_j)}_{=1}\big)^b \cdot P\big(E^{\alpha(link_j)}_{=0}\big)^{k-a-b}.$$

Assume v represents the possible value of V_i, and is an integer ranging from 0 to k. An event $V_i = v$ means that the amount of bits set to 1 in $Bfaddress(x)$ of $bf_i(link_j)$ is v. The probability of this event is given by

$$Pr(V_i{=}v) = \sum\nolimits_{a=0}^{v} Pr(U_i{=}a) \cdot Pr(W_i{=}v-a|U_i{=}a) \quad (9)$$

Thus proved. ■

Lemma 4: In the context of linear decay model, the metric $\theta(x, bf_i(link_j))$ is a discrete random variable, denoted as V_i. Its possible values are integers ranging from 0 to k. The probability mass function of V_i is defined by Formula 10.

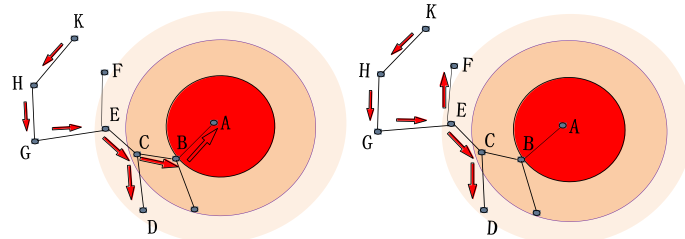
Proof: In bf_i, $\theta(x, bf_i)$ bits in $Bfaddress(x)$ remain 1 and other $k-\theta(x, bf_i)$ bits are reset to 0. The value of $\theta(x, bf_i)$ is given by Formula 7. According to the definition of $bf_i(link_j)$, number of $\alpha(link_j)$ balls will be thrown into the m bits of bf_i randomly during the construction process of $bf_i(link_j)$, where $\alpha(link_j) = \beta(link_j) - \theta(bf_i)$. The number of bits in $Bfaddress(x)$ which are set to 0 in bf_i but are hit at least once after throwing $\alpha(link_j)$ balls into bf_i is a discrete random variable, denoted as Q_i. Its possible values range from 0 to $k - \theta(x, bf_i)$. The probability that $Q_i = a$ is

$$\binom{k-\theta(x,bf_i)}{a} \cdot P\big(E^{\alpha(link_j)}_{=1}\big)^a \cdot P\big(E^{\alpha(link_j)}_{=0}\big)^{k-\theta(x,bf_i)-a}.$$

The bits being 1 in $Bfaddress(x)$ of $bf_i(link_j)$ includes those bits being 1 in bf_i and other bits set to 1 after throwing $\alpha(link_j)$ balls into bf_i. Therefore, the number of this kind of bits is a discrete random variable, denoted as V_i. Its possible values range from $\theta(x, bf_i)$ to k. The probability that V_i=v is equivalent to the probability that Q_i=$v-\theta(x, bf_i)$, and is

$$\binom{k-\theta(x,bf_i)}{v-\theta(x,bf_i)} \cdot P\big(E^{\alpha(link_j)}_{=1}\big)^{v-\theta(x,bf_i)} \cdot P\big(E^{\alpha(link_j)}_{=0}\big)^{k-v}. \quad (10)$$

Thus proved. ■



(a) Valid weak state routing. (b) Invalid weak state routing.

Fig. 1. Illustrative examples of weak state routing.

According to the first design criterion presented in Section I, we must ensure that the value of metric $\theta(x, bf_i(link_j))$ increases with the decreasing i as $\theta(x, bf_i)$ does. As shown in Fig.1(a), the value of metric $\theta(x, bf_i(link_j))$ should increase along a path $E{\to}C{\to}B{\to}A$. This issue dominates the feasibility of the weak state routing scheme using Bloom filters. The metric is a function of i and $\alpha(link_j)$, but not a monotonic decreasing function of i because $\alpha(link_j)$ is a discrete random variable with an uncertain distribution. Under a reasonable constraint on $\alpha(link_j)$, we derive Theorem 1.

Theorem 1: Given an item x represented by a bf, consider two nodes receiving bf_i and bf_{i+1} through $link_j$ and $link_j^{'}$, respectively. The expectation of $\theta(x, bf_i(link_j))$ decreases as the value of i increases if $\beta(link_j) \approx \beta(link_j^{'})$ and $h_0 \le i < h_0 + h_1$, irrespective the exponential or linear decay model.

Proof: After i rounds of decay, bf becomes a new version bf_i, the number of membership information of x becomes $\theta(x, bf_i)$ from $\theta(x, bf_i)$, and $\theta(x, bf){-}\theta(x, bf_i)$ bits being 1 in bf are reset to 0 in bf_i. For the linear decay model, Formula 7 shows that the value of metric $\theta(x, bf_i)$ decreases as the decay hop i increases. To construct $bf_i(link_j)$, number of $\alpha(link_j)$ balls are thrown into the m bits of bf_i. The probability that any bit in the m bits is hit by at least one ball is $1-(1-1/m)^{\alpha(link_j)}$. Therefore, the number of bits which belongs to those $\theta(x, bf){-}\theta(x, bf_i)$ bits and are hit is

$$f_1 = \big(\theta(x, bf) - \theta(x, bf_i)\big) \cdot \big(1{-}(1{-}1/m)^{\alpha(link_j)}\big).$$

Actually, the number of f_1 bits and remainder bits being 1 in $Bfaddress(x)$ of bf_i represent the membership information of x in $bf_i(link_j)$. Hence, we can infer that

$$E(x, bf_i(link_j)) = \theta(x, bf_i) + \\ \big(\theta(x, bf) - \theta(x, bf_i)\big) \cdot \big(1-(1-1/m)^{\alpha(link_j)}\big). (11)$$

Note that $\alpha(link_j) = \beta(link_j){-}\theta(bf_i)$ and $\alpha(link_j^{'}) = \beta(link_j^{'}){-}\theta(bf_{i+1})$. The difference between $\theta(bf_i)$ and $\theta(bf_{i+1})$ is trivial by comparing to $\beta(link_j)$ or $\beta(link_j^{'})$. Therefore the component $(1-(1-1/m)^{\alpha(link_j)}$ in Formula 11 becomes a constant factor since $\beta(link_j) \approx \beta(link_j^{'})$, and can be denoted as $0{<}g{<}1$. Formula 11 can be expressed as

$$E\big(\theta(x, bf_i(link_j))\big) = (1-g) \times \theta(x, bf_i) + g \times \theta(x, bf).$$

It is clear that $E\big(\theta(x, bf_i(link_j))\big)$ and $\theta(x, bf_i)$ are monotonic decreasing functions of i if $h_0 \le i \le h_0 + h_1$. Thus, Theorem 1 holds in the case of the linear decay model. For the

TABLE I
SUMMARY OF MAIN NOTATIONS

Term	Definition
X	a set represented by a bf
m	number of counters of a bf
n	cardinality of a set
k	number of hash functions used by a bf
$Bfaddress(x)$	bits of $h_i(x)$ for $1 \leq i \leq k$
f_p	number of incorrect item deletions
d	decay factor
h	decay range (transmission range) of a bf
c	average node degree of a given network
bf_i	a bf resulted from the ith round decay of a bf
$\theta(bf_i)$	number of bits set to 1 in bf_i
$\theta(x, bf_i)$	amount of information in bf_i for $x \in X$
$\vert link_j\vert$	number of bfs a node receives through $link_j$
$bf(link_j)$	union of $\vert link_j\vert$ decay bfs through $link_j$
$bf_i(link_j)$	$bf(link_j)$ at a node receiving bf_i through $link_j$
$\theta(x, bf_i(link_j))$	amount of information in $bf_i(link_j)$ for $x \in X$
$E^z_{=i}$	a bit is set to i after throwing z balls into a bf
p_0	fraction of bits set to zero in a bf
p_1	fraction of bits set to one in a bf
Y	noise strength about $x \in X$ at unrelated links
σ	minimum probability of a valid multi-hop routing

exponential decay model, we first achieve the expectation of $\theta(x, bf_i)$ according to Corollary 1, and replace $\theta(x, bf_i)$ with $E\big(\theta(x, bf_i)\big)$ in Formula 11. For the same reason, Theorem 1 also holds for the exponential decay model. Thus proved. ■

D. Effect of noise on routing decisions

For the weak state routing scheme, a node receiving a query for an item x selects $link_j$ so that the amount of membership information of x in $bf_i(link_j)$ is the largest one comparing to others. In other words, the node receiving bf_i through $link_j$ will send the query over $link_j$ if x belongs to a set represented by the bf and the noise at other links are trivial. It is the second design criterion mentioned in Section I. In practice, we observe that the assumption about the noise does not usually hold. To address this issue, we measure the strength of noise and evaluate its impact on weak state routing decisions.

Given an item x represented by a bf and a node A receiving bf_i through its link $link_j$, let $\theta(x, bf_i(link_j^{'}))$ denote the amount of information of x in a routing entry $bf_i(link_j^{'})$ at another link $link_j^{'}$. If node A did not receive a decayed version of bf through the link $link_j^{'}$, $\theta(x, bf_i(link_j^{'}))$ denotes the strength of noise on the information of x at that link, and is a discrete random variable, denoted as Y. Its possible value, denoted as u, is an integer ranging from 0 to k. The probability mass function of Y is defined as

$$P(Y = u) = \binom{k}{u} p_1^u p_0^{k-u}. \qquad (12)$$

Recall that $V_i{=}\theta(x, bf_i(link_j))$ denotes the amount of information of x in $bf_i(link_j)$ at node A and is a discrete random variable whose possible value, denoted as v, is an integer ranging from 0 to k. The probability mass function of V_i is given by Formulas 9 and 10 for the exponential and linear decay models, respectively. Before a query for an item x enters the decay range of a destination node, a routing decision is made randomly. As shown in Fig.1, all routing decisions along a path $K{\rightarrow}H{\rightarrow}G{\rightarrow}E$ are made randomly. Otherwise, a routing decision is made according to the following strategies.

1) The value of $\theta(x, bf_i(link_j^{'}))$ is less than v for any other $link_j^{'}$, so that node A can distinguish $link_j$ from others and forward the query for x through $link_j$. This is called an *unicast* strategy of weak state routing. For example, a query towards node A is only forwarded to node C by node E, as shown in Fig.1(a).
2) The value of $\theta(x, bf_i(link_j^{'}))$ is equal to v for some links, however, is less than v for others. In this condition, node A cannot distinguish $link_j$ from other links $link_j^{'}$ where $\theta(x, bf_i(link_j^{'}))$=$v$, and hence forwards the query through $link_j$ and such links together. This is called a *multicast* strategy of weak state routing. For example, a query towards to node A is forwarded to node B as well as node D by node C, as shown in Fig.1(a).
3) The value of $\theta(x, bf_i(link_j^{'}))$ is larger than v for a link or links except $link_j$. The strength of noise about x at such links is higher than the strength of information about x at $link_j$. Therefore, the query will be wrongly forwarded to a link or links except $link_j$. This is called an *invalid* strategy of weak state routing. For example, a query towards to node A is wrongly forwarded to node D by node C, as shown in Fig.1(b).

We will prove the probability of each aforementioned decision in theory once a query enters the propagation field of a destination. Note that each node has c links averagely and each is associated with a Bloom filter as its routing entry.

Theorem 2: A node forwards a query for an item x using the *unicast* strategy if it receives bf_i from a destination of the query. The probability of this event is given by Formula 13.

Proof: Recall that the information of x in the Bloom filter associated with $link_j$ through which current node receives bf_i is a discrete random variable, denoted as V_i. Its probability mass function has been given by Formulas 9 and 10, depending on the decay model. For any possible value v of V_i where $0 \leq v \leq k$, consider an event that the information of x in the Bloom filter associated with another link is less than v. The probability of this event is given by $\sum_{u=0}^{v-1} P(Y{=}u)$. Further, we consider an event that the information of x in the Bloom filter associated with each link except $link_j$ and the link the query came from is less than v. The probability of this event is given by $(\sum_{u=0}^{v-1} P(Y = u))^{c-2}$. Thus, the probability that the query is forwarded using the unicast strategy can be calculated by

$$f_{unicast}(V_i){=}\sum_{v=1}^{k} P(V_i{=}v)\cdot\Big(\sum_{u=0}^{v-1} P(Y{=}u)\Big)^{c-2}. \quad (13)$$

Thus proved. ■

Theorem 3: A node forwards a query for an item x using the *multicast* strategy if it receives bf_i from the destination. The probability of this event is given by Formula 15.

Proof: Recall that the information of x in the Bloom filter associated with $link_j$ through which current node receives bf_i from the destination is a discrete random variable, denoted as V_i. Its probability mass function has been given by Formulas

9 and 10, depending on the decay model. For any possible value v of V_i where $0 \leq v \leq k$, consider an event that the information of x in the Bloom filter associated with another link is less than or equal to v. The probability of this event is given by $\sum_{u=0}^{v} P(Y{=}u)$. Further, let us consider an event that the information of x in the Bloom filter associated with each link except $link_j$ and the link the query came from is less than or equal to v. The probability of this event is given by $(\sum_{u=0}^{v} P(Y{=}u))^{c-2}$. Thus, the probability that the query can be forwarded successfully using the unicast or multicast strategy can be calculated by

$$f_{valid}(V_i){=}\sum_{v=1}^{k} P(V_i{=}v) \cdot \left(\sum_{u=0}^{v} P(Y{=}u)\right)^{c-2}. \quad (14)$$

The probability that the query is forwarded using the unicast strategy has been given by Formula 13. Therefore, we can infer that the probability of the event defined in this theorem is the difference between Formulas 14 and 13, that is

$$f_{multicast}(V_i) = f_{valid}(V_i) - f_{unicast}(V_i). \quad (15)$$

Thus proved. ■

Theorem 4: A node forwards a query for an item x using the *invalid* strategy if it receives bf_i from the destination. The probability of this event is given by Formula 16.

Proof: As discussed above, the probability that queries for x are forwarded successfully according to the unicast or multicast strategy is given by Formula 14. It is easy to infer that the probability of the event defined in this theorem is

$$f_{invalid}(V_i) = 1 - f_{valid}(V_i). \quad (16)$$

Thus proved. ■

III. Feasibility of weak state routing based on decay Bloom filters

We first derive the necessary and sufficient condition for a feasible weak state routing based on decay Bloom filters. We then devise a novel receiver-oriented approach for Bloom filters to enable the weak state routing, and address redundant queries and the transmission optimization of Bloom filters.

A. Feasible weak state routing scheme with high probability

In Section II-D, we have discussed the conditions of unicast, multicast, and invalid routing strategies. Only one of such strategies will be chosen to deal with a query at each node. Theorems 2, 3, and 4 have proved the probability that each strategy is chosen. Among the three strategies, the *unicast* is a valid and desired weak state routing scheme. In this strategy, a query for an item x is only biased at an intermediate node which receives a decay Bloom filter from the destination and is closer to the destination than current node. The benefit of the *unicast* strategy is that it can ensure the correctness of routing whereas does not produce redundant queries (forwarding a query to additional intermediate nodes). The *multicast* is another valid weak state routing scheme at the cost of sending a query to some neighbors which do not receive a decay Bloom filter from the destination. A weak state routing decision is *valid* if it ensures an *unicast* or a *multicast* routing decision by preventing an *invalid* decision at nodes which reside within the decay range of the destination.

Note that the weak state routing based on decay Bloom filters is essentially a probabilistic routing. Thus, it is impossible and there is no need to achieve an absolutely valid routing decision for each query. What we need is a valid weak state routing decision for any query with a high probability. For any query, we can infer from Theorem 3 that the node which received bf_i from the destination of the query can make a valid weak state routing decision with probability $f_{valid}(V_i)$, and an unicast routing decision with probability $f_{unicast}(V_i)$.

So far, we consider the valid weak state routing decision in the scenario of one hop transmission of queries. In practice, only potential destinations of a very few queries reside one hop away from the sources of queries. Thus, we consider a general scenario in which a query traverses multiple intermediate nodes along a multi-hop path before it reaches its destination. In this scenario, a query can be sent to its destination with high probability only if each intermediate node achieves a valid routing decision for the query with high probability.

Definition 4: **(Weak state routing for multi-hop queries)** Given a multi-hop query, a *valid* routing can ensure that the query is sent to its destination by a sequence of *valid* routing decisions made at intermediate nodes once it enters the decay range of its destination. An *unicast* routing for the query requires all unicast routing decisions at intermediate nodes. An *invalid* routing for the query means that the routing decision at any intermediate node is invalid. Figures 1(a) and 1(b) plot a valid and an invalid routing for a multi-hop query, respectively.

Let σ denote a lower bound, depending on applications, on the probability that each query is sent to its destination by a valid routing scheme. According to Theorems 2 and 3, we can infer that the necessary and sufficient condition of a valid weak state routing scheme for a query is

$$\prod_{i=1}^{h} f_{valid}(V_i) \geq \sigma. \quad (17)$$

If we further seek all unicast routing decisions, the necessary and sufficient condition should be

$$\prod_{i=1}^{h} f_{unicast}(V_i) \geq \sigma. \quad (18)$$

Recall that the expectation value of the metric $\theta(x, bf_i(link_j))$ decreases as the value of i increases as shown in Theorem 1. It is easy to infer that

$$P(V_i{=}v) > P(V_{i+1}{=}v) \text{ for } \theta(x, bf_i) \leq v \leq k, 1 \leq i < h.$$

On the other hand, the noise distribution is similar in Bloom filters associated with neighbor links at each node. In summary, for any query,

$$\begin{aligned} f_{unicast}(V_i) &> f_{unicast}(V_{i+1}) \text{ and} \\ f_{valid}(V_i) &> f_{valid}(V_{i+1}). \end{aligned}$$

By now, Formulas 17 and 18 become Formulas 19 and 20, respectively, if we replace $f_{unicast}(V_i)$ and $f_{valid}(V_i)$ with

$f_{unicast}(V_h)$ and $f_{valid}(V_h)$, respectively.

$$\left(f_{valid}(V_h)\right)^h \geq \sigma. \tag{19}$$

$$\left(f_{unicast}(V_h)\right)^h \geq \sigma. \tag{20}$$

In the remainder of this paper, we will use inequation (19) or (20) to instruct the optimization of Bloom filters.

B. Receiver-oriented optimization of Bloom filters

In many distributed applications, all nodes are required to adopt the same configuration of m, k, and hash functions in order to guarantee the compatibility and inter-operability of Bloom filters. In this work, the union operation of decay Bloom filters requires the same configuration between all Bloom filters. Thus, the initial and decay Bloom filters of each node should adopt the same configuration, and so does the joint Bloom filter associated with each link. Many efforts have been made to optimize Bloom filters in stand-alone applications [16], [17] and distributed [15], [18] applications from the aspect of transmitter.

Such efforts, however, cannot address the fact that each node uses the union of all received decay Bloom filters through a link as a routing entry of that link. Although the fraction of bits set to one in each single Bloom filter might be low, that in each routing entry becomes high due to the union of many decay Bloom filters. Thus, given a query for any item at an arbitrary node, the noise about the item in unrelated routing entries is very likely equal to even stronger than the useful information in the right routing entries. Neither the design approach of Bloom filters in stand-alone applications nor the traditional transmitter-oriented optimization approach in distributed applications can be used in scenarios of the weak state routing. To address this issue, we optimize Bloom filters from the aspect of receiver, called *Wader*. The basic idea is to derive the optimal configuration of each single Bloom filter under the constraint of inequation (19) or (20).

Besides the well-known metrics of Bloom filters (the number of items n, the size of Bloom filter m, and the number of hash functions k), the decay factor d and decay range h are two additional dependent factors which impose constraints on inequations (19) and (20).

Based on a given decay model with parameters d and h, we first calculate $\theta(bf)$ and $\theta(bf_i)$ according to Formulas 1, 2 and 3 for $1 \leq i \leq h$. Note that $\theta(bf)$ is a function of variables m, n, and k, whereas $\theta(bf_i)$ is a function of variables m, n, k, and d. We then estimate the value of p_1 in each joint Bloom filter according to Formula 5, and finally obtain the distribution of noise strength at each neighbor link based on Formula 12. Note that p_1 is a function of variables m, n, k, d, and h. Similarly, According to Formulas 9 and 10, we can achieve the distribution of information of any item x in a joint Bloom filter associated with a link through which a decay Bloom filter is received from a destination. We calculate the probability of an *unicast* and a *valid* weak state routing decision by Formulas 13 and 14 which are functions of m, n, k, d, and h. Finally, inequation (19) or (20) is used to restrict the value of m, n, k, d, and h under a constraint of the lower bound σ.

The parameters n, d and h should be assigned appropriate values based on the topological properties, item distribution over nodes, query distribution over nodes, and query distribution over items. As many efforts have been done to measure the topological properties and investigate the item distribution and query distribution, depending on applications. Thus, it is reasonable to assume that we are given n, d and h. In this case, inequations (19) and (20) depend merely on parameters m and k, and hence we can optimize the number of hash functions k to maximize $f_{unicast}(V_h)$ and $f_{valid}(V_h)$ such that inequation (19) and (20) can be satisfied with m as small as possible. It is well-known that a single Bloom filter is traditionally optimal when $k{=}(m/n)\ln 2$. Such an optimal result, however, cannot ensure an optimal joint Bloom filter. After optimizing $f_{unicast}(V_h)$ or $f_{valid}(V_h)$, we can calculate the optimal value of m and k by solving inequation (19) or (20), respectively.

C. Terminating redundant queries

Given a multi-hop query, the optimal m and k for the joint Bloom filter which ensures inequation (19) are not necessary to guarantee inequation (20) since $f_{unicast}(V_i) < f_{valid}(V_i)$ under the same m and k. Therefore, the joint Bloom filter for the unicast routing always consumes more bits than that for the valid routing. The advantage of the unicast routing is that the query does not traverse any nodes which do not participate the routing path, and hence does not create any redundant query. In contrast, the *valid* routing consumes less bits than the *unicast* routing, however, usually produces redundant queries due to the potential use of *multicast* routing decision. Those replicas are sent along other paths deviating from the destination, and are redundant. Here, we first examine the number of redundant queries, and then tackle those redundant queries.

In the case of a multicast routing decision for a query at a node receiving bf_i, let S_i denote the number of neighbors to which the query is forwarded by the current node, except the neighbor receiving a bf_{i-1}. On the other hand, the query cannot be sent back to the neighbor it came from. In words, S_i measures the number of redundant queries produced by routing a query at a node which resides within the decay range of a destination. A multicast routing decision occurs if the query is sent to at least one of the remainder $c{-}2$ neighbors. Therefore, S_i is a discrete random variable. Its possible values, denoted as s, are integers ranging from 0 to $c-2$.

Theorem 5: In the case of a multicast routing decision of a query for an item x, the probability mass function of S_i is given by Formula 21.

Proof: Recall that the information of x in the Bloom filter associated with $link_j$ through which current node receives bf_i is a discrete random variable, denoted as V_i. Its probability mass function has been given by Formulas 9 and 10, depending on the decay model. For any possible value v of V_i where $0 \leq v \leq k$, consider an event that the information of x in the Bloom filter associated with one link except $link_j$ is less than v. The probability of this event is given by $\sum_{u=0}^{v-1} P(Y{=}u)$.

For any possible value v of V_i, the event $S_i{=}s$ means that the information of x in Bloom filters associated with s links among the $c{-}2$ links is equal to v, whereas that in Bloom filters associated with the other $c{-}2{-}s$ links is less than v. The probability of S_i under a given v of V_i is given by

$$\binom{c-2}{s} P(V_i{=}v) \cdot P(Y{=}v)^s \cdot \big(\sum_{u=0}^{v-1} P(Y{=}u)\big)^{c-2-s},$$

and that under all possible values of V_i is given by

$$P_m(S_i{=}s) = \binom{c-2}{s} \sum_{v=1}^{k} P(V_i{=}v) \cdot P(Y{=}v)^s \cdot \big(\sum_{u=0}^{v-1} P(Y{=}u)\big)^{c-2-s} \quad (21)$$

Thus proved. ■

So far, we only consider the event S_i caused by a multicast routing decision. Actually, such S_i might also occur under an invalid routing decision. That is, multiple links except $link_j$ have the highest information of x in their routing entries, and hence the query are wrongly forwarded to such links except $link_j$. We will discuss the probability mass function of S_i under an invalid routing decision in Theorem 6.

Theorem 6: In the case of an invalid routing decision of a query for an item x, the probability mass function of S_i is given by Formula 22.

Proof: For any possible value v of V_i where $0 \leq v \leq k$, let consider the event that s of the $c{-}2$ links (not including the link the query came from and the link $link_j$) have the highest information of x in their routing entries. The probability of this event under a given v of V_i is given by

$$\binom{c-2}{s} P(V_i{=}v) \cdot \sum_{u=v+1}^{k} \Big(P(Y{=}u)^s \cdot \big(\sum_{r=0}^{u-1} P(Y{=}r)\big)^{c-s-2}\Big),$$

and that under all possible values of V_i is given by

$$P_i(S_i{=}s) = \binom{c-2}{s} \sum_{v=0}^{k} P(V_i{=}v) \cdot \sum_{u=v+1}^{k} \Big(P(Y{=}u)^s \cdot \big(\sum_{r=0}^{u-1} P(Y{=}r)\big)^{c-s-2}\Big). \quad (22)$$

Thus proved. ■

We further consider S_i in the case of a weak state routing decision which covers the two independent cases we discussed in Theorems 5 and 6, respectively. In this case, the probability mass function and expectation value of S_i are given by

$$P(S_i = s) = P_m(S_i = s) + P_i(S_i = s) \text{ and}$$
$$E[S_i] = \sum_{s=1}^{c-2} s \cdot P(S_i = s).$$

So far, we only consider the variable S_i in the scenario of one hop transmission of a query from a node receiving bf_i to a node receiving a bf_{i-1}. Here, we consider a general query whose source node is out of the decay region of its nearest destination. In this case, the query suffers number of h one-hop transmissions in the decay region of the destination, and causes number of $E[S_i]$ redundant queries due to a routing decision for each transmission. The average number of such kind of redundant queries caused by a multi-hop query after it enters the decay range of its destination is given by

$$\sum_{i=1}^{h} \sum_{s=1}^{c-2} s \cdot P(S_i = s). \quad (23)$$

Nodes receiving such kind of redundant queries take additional computations for making decisions on routing those queries. Although each query only produces a very few redundant queries before reaching a destination as given by Formula 23, these replicas can incur non-trivial negative impact if they keep on propagating in the network. Fortunately, we find that such redundant queries can be terminated after their first transmissions with high probability as shown in Theorem 7.

Theorem 7: Given a weak state routing decision of a query for an item x at an arbitrary node, receivers of S_i resulting redundant queries stop forwarding such queries with high probability as given by Formula 24.

Proof: Recall that V_i is a discrete random variable, and denotes the information of x in the Bloom filter associated with $link_j$ through which node C receives bf_i from a destination node A, as shown in Fig.1. For any neighbor node D which does not receives a bf_{i-1} from node A, let E^i denote an event that node D receives a redundant query from node C. Let E^i_v denote an event that node C forwards a redundant query to node D since the noise strength at the link $C{\to}D$ is at least the same as a value v of V_i.

The probability of the event E^i_v and E^i is given by

$$P(E^i_v) = P(V_i{=}v) \cdot \sum_{u=v}^{k} P(Y{=}u) \text{ and}$$
$$P(E^i) = \sum_{v=0}^{k} \big(P(V_i{=}v) \cdot \sum_{u=v}^{k} P(Y{=}u)\big).$$

Thus, it is easy to infer the conditional probability of E^i_v given E^i is given by

$$P(E^i_v | E^i) = P(E^i_v)/P(E^i).$$

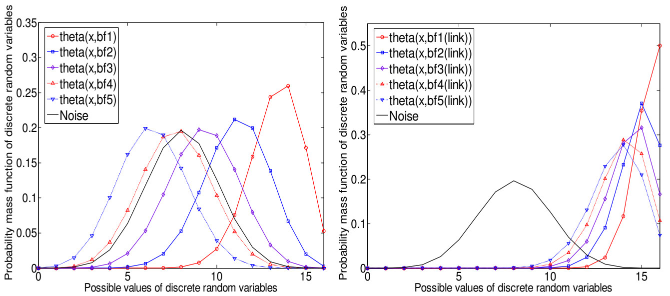
For any possible value v of V_i where $0 \leq v \leq k$, consider an event that the information of x in the Bloom filter associated with each link is less than v at node D, except the link through which a redundant query came from. This event means that the weak state routing scheme fails to find a neighbor of node D which holds higher level of information about x than v, and hence node D cannot keep on forwarding the query. The probability of this event under all possible values of V_i is

$$\sum_{v=1}^{k} P(E^i_v | E^i) \cdot \Big(\sum_{u=0}^{v-1} P(Y{=}u)\Big)^{c-1}. \quad (24)$$

Thus proved. ■

D. Optimizing transmission of Bloom filters

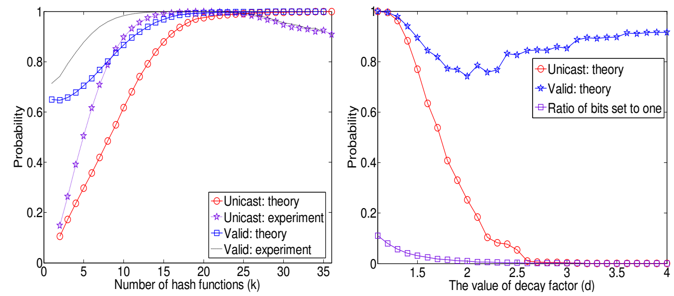
In the case of receiver-oriented design, the size of Bloom filter of each node is optimal to support a valid weak state routing with high probability. However, the size might be too large to represent items hosted by each node. On the other hand, a Bloom filter created by each node must be delivered



(a) $\theta(x, bf_i)$ and noise.

(b) $\theta(x, bf_i(link))$ and noise

Fig. 2. Probability mass functions of $\theta(x, bf_i)$, $\theta(x, bf_i(link))$ and noise, where $m = 60000$, $n = 100$, $k = 16$, $d = 1.2$, and $h = 5$.



(a)

(b)

Fig. 3. Effect of the parameters k and d on the probability of an unicast routing and a valid routing, where m=60000, n=100, and $h = 5$.

to other nodes as messages in order to establish a routing entry for each link at each node. In this case, the transmission size becomes a critical factor. To reduce message traffic, we compress each Bloom filter. Thus, besides the three well-known metrics, the transmission size corresponding to the size of Bloom filter after compression is another important metric.

The authors [19] show that the uncompressed Bloom filter, which is optimized for $k = (m/n) \ln 2$ cannot gain anything. The reason is that under good random hash functions, each bit of Bloom filter is 0 or 1 independently with probability 1/2. The Bloom filter which is optimized by receiver-oriented approach can achieve high compression gain. The reason is that under the same assumption about hash functions, each bit in joint Bloom filter is 1 with a lower probability than 1/2, that is $p_1 << 1/2$. In theory, a m bits Bloom filter can be compressed down to only $mH(p_1)$ bits, where

$$H(p_1) = -p_1 \log_2 p_1 - (1 - p_1) \log_2 (1 - p_1)$$

is the entropy function. The arithmetic coding is a near-optimal compressor which requires $m(H(p_1) + \varepsilon)$ bits for any $\varepsilon > 0$. The primary point of this theoretical analysis is to demonstrate that compression is a viable means to significantly reduce transmission size and save bandwidth. The cost is additional computation for compression and decompression, which can be implemented by using simple arithmetic coding.

IV. Performance Evaluations

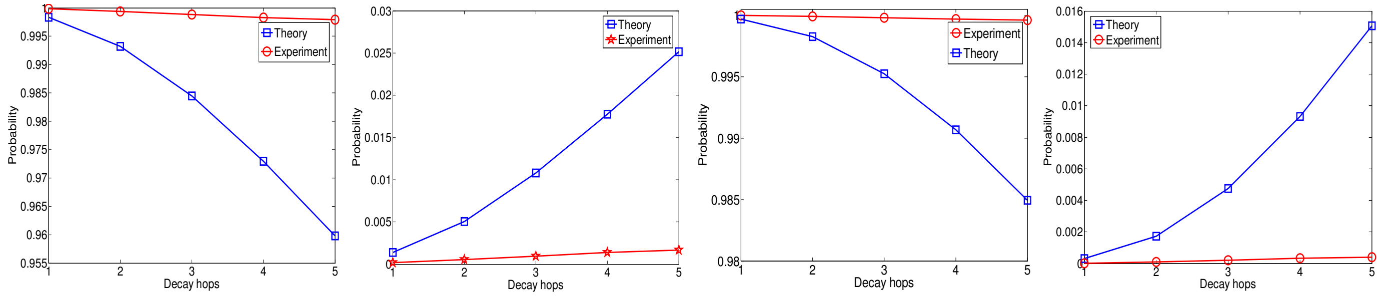
We evaluate the performance of *Wader* and demonstrate that only the receiver-oriented Bloom filters can guarantee the feasibility of the weak state routing. The settings in the simulations are as follow. We generate a random network topology in which the node degree ranges from 3 to 7 and the average node degree is c=5. The average number of items hosted by each node is n=100. All Bloom filters are decayed from the first round of propagation. That is, h_0=0. For a large h_0, we have to enlarge the value of m to satisfy the same lower bound σ, which results in unnecessarily higher space cost. We have carried out simulations under both the exponential decay model and the linear decay model, and have obtained similar results. Due to the page limit, we only report the results under the exponential decay model.

A. Effect of decaying operation on membership information

Assume the decay factor is set to be d=1.2 and the decay range is set to be h=5, depending on a given application. Then, we can derive that an optimal number of bits for each Bloom filter is m=60000 and the number of hash functions is k=16 from the aspect of receiver. Note that the optimal number of hash functions is about 416 from the aspect of transceiver if we use the known formula, k=$m/n \ln 2$. Given a bf which represents a set X, we have analyzed the amount of information of any item $x \in X$ in a decay version of bf in Lemma 2. The possible values of $\theta(x, bf_i)$ are integers ranging from 0 to k=16. Fig.2(a) shows the probability mass function of $\theta(x, bf_i)$ for $1 \leq i \leq 5$ and noise. The results match well with Formula 8. As we can see from the figure, when the possible value increases, the probabilities of $\theta(x, bf_i)$ first goes up and then goes down for $1 \leq i \leq 5$. On the other hand, the probabilities of $\theta(x, bf_i)$ for the large possible values decrease as the value of i increases, whereas that for those small possible values increase as the value of i increases. The experimental results exactly conform to the analytical results.

Recall that $\theta(x, bf_i)$ is not accurate enough to support a weak state routing scheme since each node uses the union of all received Bloom filters through the same link as a routing entry for that link. As shown in Lemmas 3 and 4, we replace $\theta(x, bf_i)$ with $\theta(x, bf_i(link_j))$ to characterize the amount of information of x in a joint Bloom filter $bf_i(link_j)$ at the node which receives bf_i through $link_j$. Fig.2(b) shows the probability mass functions of noise and $\theta(x, bf_i(link_j))$. Recall that the noise means the amount of information of x in a Bloom filter associated with unrelated links except $link_j$. The simulation results follow a similar trend as the theoretical results given by Formula 9. As we can see from the figure, when the possible value increases, the probabilities of $\theta(x, bf_i(link_j))$ first goes up and then goes down where $1 \leq i \leq 5$. On the other hand, the probabilities of the $\theta(x, bf_i(link_j))$ for the large possible values decrease as the value of i increases, whereas that for those small possible values increase as the value of i increases.

Fig.2(b) also shows that the expectation of $\theta(x, bf_i(link_j))$ decreases as the decay hop i increases, and thus the first design criterion proposed in Section I is satisfied by *Wader*. In addition, the expectation of $\theta(x, bf_i(link_j))$ is larger than that of noise for $1 \leq i \leq h$. This reveals the reason why a node



(a) Unicast decision (b) Multicast decision (c) Valid decision (d) Invalid decision

Fig. 4. Probability of four types of routing decisions, where $m = 60000$, $n = 100$, $k = 16$, $d = 1.2$, and $h = 5$.

holding bf_i can forward a query for an item x to a node holding a bf_{i-1} with high probability, and thus satisfy the second design criterion proposed in Section I. On the other hand, the simulation results conform to Theorem 1 in terms of the expectation value of $\theta(x, bf_i(link_j))$ for $1{\leq}i{\leq}h$.

B. Receiver-oriented optimization of Bloom filters

Now we examine the effect of the parameters k and d on the probability that each query is sent to its destination through an unicast routing or a valid routing. As shown in Fig.3(a), given a fixed m and d=1.2, k is the only dependent factor of all four curves which follow a similar trend. They first ascend as k increases and quickly reach the peak, and then descend as k increases. The reason is that $\theta(x, bf_i(link))$ and noise strength increase for any x and $1 \leq i \leq h$ as k increases, and $\theta(x, bf_i(link))$ is more likely higher than the noise relatively. As discussed in Section III-B, inequations (19) and (20) are the benchmarks to optimize the parameters of Bloom filters. Given a lower bound σ on the probability of an unicast routing or a valid routing for each query, we can find the optimal value of k under each scenario. Similarly, we can achieve the optimal k under varying value of m, and can finally find the global optimal k and m. The simulation results are omitted due to limited space.

As shown in Fig.3(b), given a fixed m and k=16, the probability of an unicast routing for any query decreases as the decay factor d increases in theory, and reaches almost zero after the decay factor exceeds a threshold. The reason is that $\theta(x, bf_i(link))$ and noise strength decrease as the decay factor increases for $\forall x \in X$ and $1 \leq i \leq h$, and the noise strength is more likely higher than $\theta(x, bf_i(link))$. We can also see that the probability of the valid routing first decreases, and then increases as the decay factor increases. It is worth noticing that a small decay factor should be adopted in order to ensure the unicast routing with high probability, although a large decay factor can always guarantee the valid routing with high probability. For a large decay factor, we have to enlarge the value of k in order to satisfy the same lower bound σ, which results in unnecessarily higher computation cost.

C. Impact of noise on a weak state routing decision

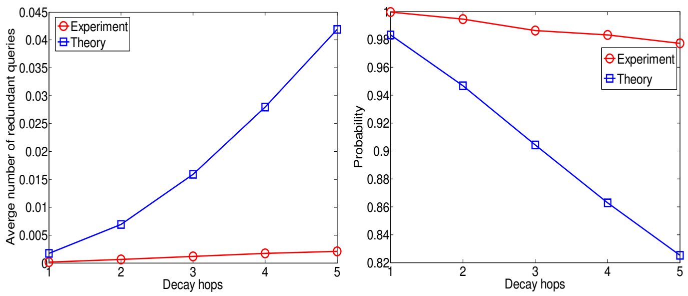
We examine the impact of noise on a weak state routing decision when each node adopts an optimal Bloom filters based on *Wader*. A weak state routing decision for a single-hop query can be valid (unicast or multicast) or invalid under the interference of noise in unrelated links once the query enters the decay range of a destination. The probabilities of the aforementioned routing decisions have been proved in Theorems 2, 3 and 4. Fig.4 shows the probabilities of those routing decisions from aspect of both theory and practice.

We can see that the probability of an unicast routing decision decreases with the increasing of the decay hop, whereas the probability of a multicast routing decision increases with the increasing decay hop. The reason is that the expectation value of metric $\theta(x, bf_i(link_j))$ decreases as the decay hop increases. Thus, the noise strength is more likely higher than $\theta(x, bf_i(link_j))$, and queries might suffer invalid or multicast routing decision. Fig.4(c) shows that the probability of a valid routing decision decreases as the decay hop increases. The reason is that the negative effect of decreasing unicast routing decision outperforms the positive effect of increasing multicast routing decision. Fig.4(d) shows that the probability of an invalid routing decision increases as the decay hop increases.

It is worth noticing that the probabilities of the unicast and valid decisions for routing a single-hop query is high for $1{\leq}i{\leq}h$. Thus, a multi-hop query can reach a destination through a sequence of valid even unicast routing decisions with high probability. As shown in Fig.4, the curve of practical probability follows the same trend as the curve of the theoretical probability for each type of routing decision. The practical probability, however, is larger than the theoretical value for the unicast and valid routing decisions. This demonstrates that Formulas 13 and 14 provide lower bounds on the probabilities of the unicast and valid routing decisions, respectively. In addition, *Wader* achieves lower probabilities of the multicast and invalid routing decisions than the corresponding upper bounds given by Formulas 15 and 16. In summary, the theoretical and practical results demonstrate that *Wader* guarantees the correctness and efficiency of weak state routing for multi-hop queries with high probability.

D. Termination of redundant queries

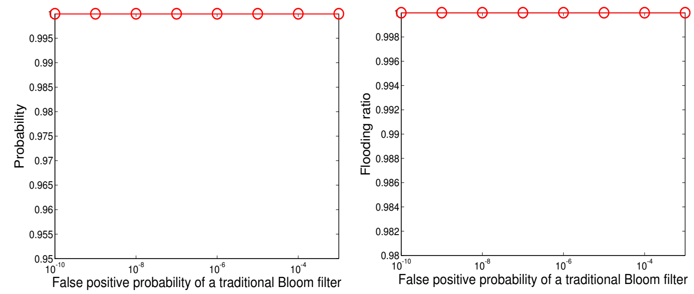
A node possibly suffers the multicast or invalid decision, and then sends very few redundant queries (according to Formula 23), to neighbors which deviate from the potential destination. We have proved the probability that such queries



(a) Redundant queries

(b) Termination probability

Fig. 5. Number of redundant queries and the probability that they will be terminated by receivers, where m=60000, n=100, k=16, d=1.2, and h=5.



(a) Valid decision

(b) Flooding ratio at any node

Fig. 6. Probability of valid decision and flooding ratio at any node, where m=60000, n=100, k=16, d=1.2, and h=5.

can be terminated by receivers with high probability in Theorem 7. As shown in Fig.5(a), the average number of redundant queries caused by routing one query increases as the decay hop increases in both theory and practice. As shown in Fig.5(b), the termination probability of those redundant queries by receivers decreases as the decay hop increases in theory as well as in practice, but the termination probability still remains at a high level in theory as well as in practice. Note that the simulation results outperform the theoretical results. Formula 23 provides an upper bound on the number of redundant queries resulted from routing any query at an arbitrary node. Formula 24 provides a lower bound on the termination probability of any redundant query by an arbitrary receiver. In summary, the practical results of *Wader* conform to the theoretical analysis, and thus the negative effect of redundant queries can be controlled at a low level. This is very helpful to ensure the feasibility and usability of the weak state routing.

E. Performance of the current weak state routing schemes

We examine the impact of noise on a weak state routing decision when each node adopts a transmitter-oriented Bloom filter. Let f_p denote an upper bound on the false positive probability of the Bloom filter. Given f_p and n, k and m could be optimized with m=$\lceil n \times \log(f_p)/\log(0.6185)\rceil$ and k=$\lceil (m/n)\ln 2\rceil$ [20]. Fig.6 shows that the probability of a valid routing and the flooding ratio of a query are very close to 1 at any node, including the source node of the query, when f_p ranges from 10^{-10} to 10^{-3}. The flooding ratio of a query at any node means the ratio of the number of query replicas caused by this node to the node degree minus one. In these cases, the ratio of bits set 1 in each routing entry is very close to 1. Thus, the noise in all unrelated routing entries and the useful information in right routing entries approximate to k. We have analyzed the root cause of this result in Section III-B.

Consequently, each node almost forwards a query replica to all neighbors expect the one the query comes from. The message complexity of the current weak state routing schemes is $O(n)$ while that of Wader is $O(logn)$, where n and $logn$ denote the network size and approximate network diameter. Actually, the current weak state routing schemes are equivalent to the well-known flooding approach due to the usage of transmitter-oriented Bloom filters. It is worth noticing that the termination rules of redundant queries mentioned in Section III-C are not invoked since it can terminate all query replicas.

V. Conclusion

In this paper, we observe that the existing weak state routing schemes cannot facilitate queries effectively because of wrong routing decisions. The queries are often routed towards nodes that don't have the required items. In such cases, weak state routing is downgraded to network flooding, which incurs excessive traffic but achieves extremely low efficiency. To address this problem, we derive the necessary and sufficient condition for a feasible weak state routing scheme, and accordingly propose a receiver-oriented approach of Bloom filters to satisfy the condition. The simulation results demonstrate that *Wader* ensures the correctness and efficiency of weak state routing using decay Bloom filters with high probability.

Following this work, we will extend it in several potential directions. First, we plan to evaluate the impact of network configurations on *Wader*, such as the topological properties, data distribution over nodes, and query distribution. Second, let each node monitor and control the fraction of 0 bits in each routing entry in order to make *Wader* adaptive to varying network configurations. Third, we will improve the efficiency of routing that is outside the decay region of a destination, by using a random replication mechanism and taking multi-path routing into account.

References

[1] B. Bloom, "Space/time tradeoffs in hash coding with allowable errors," *Commun. ACM*, vol. 13, no. 7, pp. 422–426, 1970.

[2] N. Hua, H. Zhao, B. Lin, and J. Xu, "Rank-indexed hashing: A compact construction of bloom filters and variants," in *Proc. 16th IEEE ICNP*, Orlando, Florida, USA, Oct. 2008, pp. 73–82.

[3] T. D. Hodes, S. E. Czerwinski, and B. Y. Zhao, "An architecture for secure wide-area service discovery," *Wireless Networks*, vol. 8, no. 2-3, pp. 213–230, 2002.

[4] P. Reynolds and A. Vahdat, "Efficient peer-to-peer keyword searching," in *Proc. ACM International Middleware Conference*, Rio de Janeiro, Brazil, June 2003, pp. 21–40.

[5] S. C. Rhea and J. Kubiatowicz, "Probabilistic location and routing," in *Proc. IEEE INFOCOM*, New York, USA, June 2004, pp. 1248–1257.

[6] D. Bauer, P. Hurley, R. Pletka, and M. Waldvogel, "Bringing efficient advanced queries to distributed hash tables," in *Proc. IEEE Conference on Local Computer Networks*, Tampa, FL, USA, Nov. 2004, pp. 6–14.

[7] P. Hebden and A. R. Pearce, "Data-centric routing using bloom filters in wireless sensor networks," in *Proc. the 4th International Conference on Intelligent Sensing and Information Processing (ICISIP)*, Bangalore, India, Dec. 2006.

[8] P.-H. Hsiao, "Geographical region summary service for geographical routing," in *Proc. 2nd ACM MobiHoc*, Long Beach, CA, USA, Oct. 2001, pp. 263–266.

[9] W. H. Yuen and H. Schulzrinne, "Improving search efficiency using bloom filters in partially connected ad hoc networks: A node-centric analysis," *Computer Communications*, vol. 30, no. 16, pp. 3000–3011, 2007.

[10] U. G. Acer, S. Kalyanaraman, and A. A. Abouzeid, "Weak state routing for large scale dynamic networks," in *Proc. the 13th ACM MOBICOM*, Montral, Qubec, Canada, Sept. 2007.

[11] R. Gilbert, K. Johnson, S. Wu, B. Y. Zhao, and H. Zheng, "Location independent compact routing for wireless networks," in *Proc. ACM MOBISHARE*, Los Angeles, CA, USA, Sept. 2006.

[12] C. F. Chan, "Mole: Multi-hop object location in wireless mesh networks," Ph.D. dissertation, Hong Kong University of Science and Technology, Aug. 2008.

[13] A. Kumar, J. Xu, and E. W. Zegura, "Effcient and scalable query routing for unstructured peer-to-peer networks," in *Proc. IEEE INFOCOM*, Miami, FL, United States, Mar. 2005, pp. 1162–1173.

[14] X. Li, J. Wu, and J. J. Xu, "Hint-based routing in WSNs using scope decay bloom filters," in *Proc. IWNAS*, 2006, pp. 111–118.

[15] D. Guo, J. Wu, H. Chen, and X. Luo, "Theory and network applications of dynamic bloom filters," in *Proc. 25th IEEE INFOCOM*, Barcelona, Spain, Apr. 2006.

[16] S. Lumetta and M. Mitzenmacher. Using the power of two choices to improve bloom filters. http://www.eecs.harvard.edu/michaelm/postscripts/.

[17] B. Chazelle, J. Kilian, R. Rubinfeld, and A. Tal, "The bloomier filter: an efficient data structure for static support lookup tables," in *Proc. 5th SODA*, New Orleans, Louisiana, USA, Jan. 2004, pp. 30–39.

[18] P. S. Almeida, C. Baquero, N. M. Preguiça, and D. Hutchison, "Scalable bloom filters." *Inf. Process. Lett.*, vol. 101, no. 6, pp. 255–261, 2007.

[19] M. Mitzenmacher, "Compressed bloom filters," *IEEE/ACM Transactions on Networking*, vol. 10, no. 5, pp. 604–612, 2002.

[20] A. Broder and M. Mitzenmacher, "Network applications of bloom filters: A survey," *Internet Mathematics*, vol. 1, no. 4, pp. 485–509, 2005.