Analysis of False Negative Rates for Recycling Bloom Filters (Yes, They Happen!)

KAHLIL DOZIER, Columbia University, US
LOQMAN SALAMATIAN, Columbia University, US
DAN RUBENSTEIN, Columbia University, US

Bloom Filters are a desirable data structure for distinguishing new values in sequences of data (i.e., messages), due to their space efficiency, their low false positive rates (incorrectly classifying a new value as a repeat), and never producing false negatives (classifying a repeat value as new). However, as the Bloom Filter's bits are filled, false positive rates creep upward. To keep false positive rates below a reasonable threshold, applications periodically "recycle" the Bloom Filter, clearing the memory and then resuming the tracking of data. After a recycle point, subsequent arrivals of recycled messages are likely to be misclassified as new; recycling induces false negatives. Despite numerous applications of recycling, the corresponding false negative rates have never been analyzed. In this paper, we derive approximations, upper bounds, and lower bounds of false negative rates for several variants of recycling Bloom Filters. These approximations and bounds are functions of the size of memory used to store the Bloom Filter and the distributions on new arrivals and repeat messages, and can be efficiently computed on conventional hardware. We show, via comparison to simulation, that our upper bounds and approximations are extremely tight, and can be efficiently computed for megabyte-sized Bloom Filters on conventional hardware.

CCS Concepts: • **Theory of computation** → **Data structures design and analysis**; • **Networks** → *Network algorithms.*

Additional Key Words and Phrases: Bloom Filter, Recycling Bloom Filter, False Negatives

ACM Reference Format:
Kahlil Dozier, Loqman Salamatian, and Dan Rubenstein. 2024. Analysis of False Negative Rates for Recycling Bloom Filters (Yes, They Happen!). *Proc. ACM Meas. Anal. Comput. Syst.* 8, 2, Article 21 (June 2024), 34 pages. https://doi.org/10.1145/3656005

1 INTRODUCTION

Bloom Filters [3] have long been used as a means for tracking repeat data items (i.e., messages) in a sequence where repeat occurrences are possible. The Bloom Filter is a desirable structure due to its efficient use of memory, and presumed property of only erring in the false positive direction.

However, as more new messages are inserted into the Bloom Filter, the false positive rate grows; eventually the rate is too high for the Bloom Filter to be of use. To keep the false positive rate below a reasonable threshold, many applications utilize what is termed in [9] a *recycling Bloom Filter* (RBF for short). The RBF "recycles" by clearing all the bits, continuing its regular bit-filling process afterward. While this helps to keep false positive rates low, it does introduce the possibility of false negatives– subsequent arrivals of previously recycled messages can map to unset bits in the

Authors' Contact Information: Kahlil Dozier, Columbia University, New York, NY, US, kad2219@columbia.edu; Loqman Salamatian, Columbia University, New York, NY, US, ls3748@columbia.edu; Dan Rubenstein, Columbia University, New York, NY, US, danr@cs.columbia.edu;

ACM 2476-1249/2024/6-ART21
https://doi.org/10.1145/3656005

Bloom Filter, appearing new. Prior art in several domains, including Software-Defined Networks (SDN) [19], DDoS detection [7], stateful load balancing to ensure per-connection consistency in datacenters [20] and payload attribution [26], make use of RBFs, arguing that false negative rates can be kept sufficiently low by recycling with limited frequency. Prior art has also proposed what is termed a *two-phase RBF* [17]: the memory used to implement the RBF is split in half and two RBFs are implemented in a manner that intuitively keeps false negative rates lower for a given false positive rate. To our knowledge, analytically determining false negative rates for RBFs, including the two-phase variants, has never been attempted. *In this paper, we present analytical models that can be used by practitioners to better anticipate false negative rates of their systems.* An analysis of false negative rates is challenging because it is heavily dependent on the arrival process of both new and repeat messages. Computing exact false negative rates appears intractable, but we present a lower bound. Additionally, we present several approaches to upper bound the false negative rate which trade tightness of the bound with required computation time, as well as a lower-complexity method to tightly approximate the false negative rate.

We begin in §2 by providing a brief background on Bloom Filters and review the prior art of false positive analysis. We formulate our general model for analytically describing the false negative rate in §3. We use Markov Modelling in §4 to generate a lower bound, three upper bound variants, and a tight approximation, and a Renewal model in §5 to develop an extremely tight upper bound. We verify our models using discrete-event-driven simulation in §6, and contrast performance of the RBF variants as functions of memory size and desired false positive rates. We explore the applicability of our bounds to actual workload traces in §7.

Of the various approaches to bounding RBF false negative rates, the Renewal Model is the most accurate across different arrival distributions, especially when there are many new arrivals to the distribution. However, in cases of low new arrival rates, Markov Model based bounds and Approximations may be more computationally tractable while still retaining decent accuracy.

2 BACKGROUND

2.1 Bloom Filters

A Bloom Filter (BF) [3] implemented atop an M-bit memory initially sets all memory bits to 0. An arriving message is hashed by k distinct hash functions, each of which maps to a bit in the array: the bit is then set to 1, regardless of its prior status (0 or 1). When the k hash functions are applied to a message and the corresponding bits in the BF are set, we say that the message is *recorded* in the BF. An arriving message is said to be recorded even when the corresponding bits are set prior to its arrival, such that the new arrival does not set additional bits in the BF. *Importantly, these k hash functions' mappings are independent across messages, such that each message's assignment of bits is effectively independent from those assigned to other messages.* This independence property is what makes the BF such a powerful abstraction, as well as making it more amenable to mathematical analysis.

For the case when a message is recorded but all of its corresponding bits are set prior to its arrival, it is assumed that the message is a repeat of a prior message. If the message is not a repeat, it is incorrectly classified as a repeat, i.e., a *false positive*.

2.2 RBF variants and Measuring False Positives

A *Recycling Bloom Filter* (RBF) is a BF that can be used indefinitely. The false positive rate can be kept arbitrarily low by clearing out all the bits set in the BF. However, frequent clearing of the BF can result in repeat messages being classified as new (false negatives), so the clearing must be

performed sparingly. One must therefore determine an acceptable false positive rate f_p, and recycle as infrequently as possible while bounding the false positive rate by f_p.

2.2.1 Two-phase variants. A proposed variant of the RBF is the two-phase RBF which splits the M-bit memory into two equally sized chunks of $M/2$ bits, using each half to implement a half-sized RBF. At any time, one RBF is deemed *active*, and arriving messages are hashed into the active RBF. During this time, the other RBF is deemed *frozen*, and remains unmodified. At initialization, both RBFs are cleared. When a message arrives, its hash bits are compared to the bits set in the active RBF and the frozen RBF. If the k hashes all map to set bits in either the frozen or the active RBF, the message is classified as a repeat. Upon a recycle event, the frozen filter is cleared and becomes active while the active filter becomes frozen. In this manner, the RBF frees up memory to keep false positives sufficiently low, while retaining some memory of the more recent arrivals. While the two-phase RBF must be recycled more frequently to maintain a given false positive rate, intuitively its benefit comes from its ability to retain some memory at all times of previously arrived messages, lowering its comparative false negative rate.

2.2.2 Measuring false positive rate. We note that [9] provides detailed analysis of false positive rates for both one-phase and two-phase RBFs, emphasizing that there are a variety of ways to measure the false positive rate (e.g., worst case when the BF is at its fullest, or average between recycles). There are specific intricacies that might affect the underlying false positive rate, such as whether a given messsage maps to k distinct bits versus permitting the k hash functions to "collide" and possibly map to the same bit. Our analysis is agnostic to the specific details of the underlying false positive rate and can be applied for any of these variants. Unless otherwise specified, the results presented here utilize what [9] refers to as the "colliding, retaining, σ-bounded" variant.

3 MODEL PRELIMINARIES

Our model considers an RBF of size M bits. We let k be the number of hash functions applied per arriving message. When the number of bits in the BF exceeds a threshold σ, the BF recycles, clearing all bits. As pointed out in [9], when a message's inclusion triggers a recycle, the message (still in the system's buffer) is then recorded into the now empty BF as its first arriving message since being cleared.

3.1 Distributions

False negative rates in RBFs depend heavily on the popularity (likelihood of repeat) of messages previously inserted into the RBF. This stands in contrast with false positive rates, which can be computed irrespective of prior message popularity (a repeat message never generates false positives, so one need not distinguish which message is repeating to determine false positive rates). Furthermore, false positives can only occur during a message's first arrival (subsequent arrivals are true positives). Because hash functions are used that (pseudo-)randomly select the BF bits to set, the false positive likelihood depends only on the number of bits set in the BF, which depends only on the number of (and not which) messages arrived previously in the current BF cycle.

Our analysis permits a rather general description of the message arrival process. We are interested in RBFs in a steady-state system, so we assume that message arrivals have been ongoing for some time when the analysis begins. We assume the "snapshot" at a particular time t of the distribution of active messages is stationary with respect to message popularity. Previously unseen messages may arrive and enter into the distribution, but the popularity of existing messages is adjusted such that the underlying popularity distribution remains the same. More formally:

- At any point in time, there is a fixed number D of messages that have arrived previously (and may arrive again).
- At that time, these D messages can be sorted by their popularity of being the next selected message. We say that the ith of D messages in the distribution at time t has relative popularity q_i where $\sum_{i=0}^{D-1} q_i = 1$ and the messages are ordered such that $q_i \geq q_{i+1}$. We call this the *repeat distribution*, $\mathcal{Q} = \{q_i\}$.
- When a message arrives, there is a probability p_r that the message is drawn from the set of D previously seen messages, and in this case, the likelihood of sampling the ith message is q_i.
- With probability $1 - p_r$, a new message (different from the D previously seen messages) is sampled. The Dth message $\mathcal{Q}$ is evicted, and the new message is inserted into $\mathcal{Q}$ as the ith most popular message in $\mathcal{Q}$ with probability ζ_i where $\mathcal{N} = \{\zeta_i\}$ is the distribution on the location of new message insertions: $\sum_{i=1}^{D} \zeta_i = 1$. Any message in $\mathcal{Q}$ whose popularity was q_j for $j \geq i$ has its popularity shifted to q_{j+1}; i.e., the existence of a new message with higher probability reduces its likelihood of selection.

This construction is a very general way of expressing a stationary distribution that permits new arrivals into the distribution over time. Using this construction, we will frequently need to distinguish the history of an arriving message during a particular cycle. We use the following nomenclature for a message:

- **New**: a message sampled from $\mathcal{N}$ is referred to as new, as it has never previously arrived to the RBF. Only its initial arrival can be new, since it is subsequently embedded in the repeating distribution $\mathcal{Q}$ from where it can be sampled repeatedly.
- **Non-repeat**: a message sampled from $\mathcal{Q}$ that is not currently recorded in the RBF is called a non-repeat. It is not new in the sense that since it is being drawn from $\mathcal{Q}$, it must have been previously sampled, but in a previous RBF cycle.
- **Repeat**: a message sampled from $\mathcal{Q}$ that has already been explicitly recorded during the current RBF cycle.

With this nomenclature established, we formally define false positives and false negatives: consider a sequence of N messages arrivals, possibly containing repeats. Define a random variable S_i to describe the status of the ith arrival as either a new message ($S_i = 0$), a non-repeat ($S_i = 1$) or a repeat ($S_i = 2$). We write $I[S_i = j]$ to be an indicator r.v. that equals 1 only when $S_i = j$. Also, let T_i be an indicator r.v. that equals 1 when all k hashes of the ith message are actively set within the BF.

A *false positive* is a Non-repeat message that is identified as a repeat by the RBF, due to the event of the k corresponding hash bits already being set in the RBF. The false positive rate over the N received messages is $f_p = (\sum_{i=1}^{N} I[S_i = 0]T_i)/N$.

A *false negative* is a Non-repeat message that is identified as new by the RBF, for at least one of its k corresponding hash bits were cleared upon a recycle event and remained clear until its next arrival. The false negative rate over N messages is $f_n = (\sum_{i=1}^{N} I[S_i = 1](1 - T_i))/N$.

3.2 Two-phase Variants

When an arriving message's hash bits are all set in the frozen filter, the message is considered a repeat, and we can choose whether to update the bits in the active BF. Intuitively, one might think it better not to set the bits in the active filter– after all, doing so simply replicates an identification already being made within the frozen filter, and increases the likelihood of false positives. However, simulations show that for certain popularity distributions, setting these bits provides a distinct advantage: we capture popular items that arrive with high frequency in both the active and frozen filter. Had we not done so, when recycling occurs and the frozen filter is cleared, memories of

these popular items are lost, triggering a high likelihood of false negatives. We refer to this latter approach as the *copy-over* variant.

For the Two-phase variant, the definitions of non-repeat and repeat are extended such that $S_i = 1$ only when the re-arriving message is not recorded in either the active or frozen BF, whereas $S_i = 2$ when the re-arriving message is recorded in either (or both) the active and frozen BF. Furthermore, the definition of T_i is extended to equal 1 when all k hashes are actively set in either the active or the frozen BF; the formulas remain the same to describe f_p and f_n.

3.3 Analyzing False Negative Rate

We define $f_n(M, k, \sigma, \mathcal{Q}, p_r, \mathcal{N})$ to be the steady-state false negative rate of an M-bit RBF that utilizes k hash functions, utilizes sigma threshold σ, and whose underlying popularity and new arrival distributions are $\mathcal{Q}$ and $\mathcal{N}$ respectively, and arriving messages are new with probability $1 - p_r$. Since much of our analyses work with fixed values of these parameters, we omit their inclusion when describing the function, often just using f_n.

Modeling the false negative rate is significantly more challenging than modeling the false positive rate. One challenge comes from the observation that over a single cycle, as more messages are recorded into the RBF, the false negative rate will drop (the likelihood of resampling a message that is currently recorded in the RBF generally increases as the number of messages recorded increases). Furthermore, by the pseudo-random property of the hash functions, the bits set from an arriving message can effectively be viewed as random, such that the model does not need to "track" specifics about the messages that set these bits. In fact, [9] shows that for a σ-bounded RBF, tracking the number of messages recorded is not needed: one only needs to track the number of bits set in the RBF. This is an advantage for modeling false positives; in contrast, the likelihood that an arriving message results in a false negative is dependent on the specific arriving message's popularity, as well as how many messages are currently recorded in the RBF. These computations depend on the underlying distribution $\mathcal{Q}$, making it challenging to get closed-form or even scalable recursive solutions exactly matching the false negative rate. We further elaborate in §4.1.4 after giving more context.

4 MARKOV MODEL APPROACHES

As mentioned in the previous section, the likelihood that an arriving message triggers a false negative is strongly influenced by the number of messages recorded in the RBF, which (unfortunately for modeling) is strongly affected by the number of messages in the RBF. [9] also utilized a Markov Model approach, but there, each state represents the number of bits set within a BF. We instead employ a Markov Model where each state i represents the number of *distinct* messages that have thus far been received in the cycle. The value of i counts all *new* and *non-repeat* messages. Our steady-state assumption that all messages in $\mathcal{Q}$ have been sampled at least once (but perhaps only in previous cycles), whereas messages sampled from $\mathcal{N}$ have not, implies that if a false negative event occurs, a non-repeat must have occurred.

We extend our definition of f_n to $f_n[i]$[1]: the likelihood of a false negative when a message arrives to the RBF that has recorded i distinct messages. By finding the steady-state probabilities Π_i of each state i, it follows that the overall average false negative rate is $f_n = \sum_i \Pi_i f_n[i]$.

We note that the actual false negative rate in state i, $f_n[i]$, must exclude the unlikely *get-lucky* event where a non-repeat message is recorded, but is not reported to be new because its k hashes "got lucky" and hashed to bits previously set in the BF: this is a "lucky" event for a non-repeat because its

[1]We use square brackets to distinguish between the state of the Markov Model and the input parameters to f_n, i.e., in long-form we would write $f_n(M, k, \sigma, \mathcal{Q}, p_r, \mathcal{N})[i]$.

resulting classification of "repeat" is in fact correct. Such an event is "unlucky" for a truly new (from $\mathcal{N}$) message, since this is what leads to a false positive. We can represent $f_n[i] = \widehat{f_n}[i](1-\gamma_k(i))$, where $\widehat{f_n}[i]$ is the likelihood that the sampled message from $\mathcal{Q}$ is a non-repeat, and $\gamma_k(i)$ is the likelihood that recording the message adds no bits. These two events can be treated independently due to the hash functions mapping messages to RBF bits in a pseudo-random manner: the number of messages recorded in the RBF, and not their specific identities will affect the get-lucky likelihood.[2]

4.1 Transitions

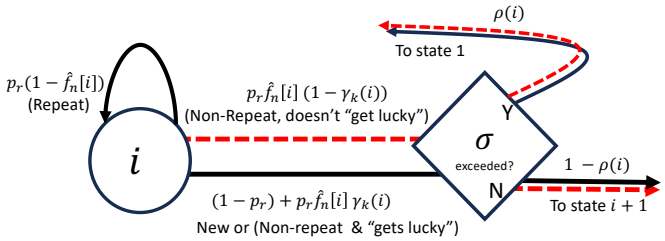


Fig. 1. Markov model showing state transitions of our model. Red transitions are dashed and indicate a false negative event.

As illustrated in Figure 1, there are three possible transitions out of each state i, and we classify two of these transitions into two types, a *red* and *black*:

- A self-loop (black), when an earlier recorded message re-arrives, i.e., a repeat message. This only occurs when the message is sampled from $\mathcal{Q}$ (with probability p_r) and the message was previously seen this cycle, and hence not a false negative (with probability $1-\widehat{f_n}[i]$), such that self-loop transitions have probability $p_r(1-\widehat{f_n}[i])$. Thus, transitions exiting state i have transition probabilities that sum to $1-p_r+p_r\widehat{f_n}[i]$.
- If the arriving message is previously unseen (new or non-repeat) in the current cycle, the Markov Model transitions to the $i+1$st state, *unless* this $i+1$st previously unseen message triggers a recycle event (i.e., the number of BF bits set exceeds σ).
- When a recycle event is triggered, there are two variants for the Markov Model. The first is the *non-retaining* variant where the arriving message is not recorded into the empty BF. The second is the *retaining* variant that does record the message. The former is represented by a transition to state 0, while the latter is represented by a transition to state 1. Our experience via simulation suggests that retaining generally yields smaller f_n and so we focus here on that variant.
- For non-self looping transitions, the transition can be marked as a false negative (red, dashed) or benign (black): its subsequent destination (state $i+1$ or state 1) is based solely on the number of bits set in the BF, and is independent of transition "color". Note that the non-self-loop probability $1-p_r+p_r\widehat{f_n}[i]$ is shared between black and red transitions: the red (false negative) transition with probability $p_r\widehat{f_n}[i](1-\gamma_k(i))$ corresponds to a sampling from $\mathcal{Q}$ (with probability p_r) being a non-repeat ($\widehat{f_n}[i]$) and not "getting lucky" ($1-\gamma_k(i)$).

Hence, for each state i, we need to determine the likelihood that the $i+1$st arriving message triggers a recycle event. Our work focuses on using the σ-bound, but we show how our approach

[2]As an aside, false positives occur at rate $(1-p_r)\gamma_k(i)$.

can be adapted if one prefers an N-bounded approach (that counts bit-setting received messages) in §4.1.2.

4.1.1 Bit-Measuring Metrics. While the Markov Model explicitly tracks the number of recorded messages, there are two reasons we need to also track the number of bits set in the RBF: The recycling event occurs when the number of set bits exceeds σ, and the likelihood of "get lucky" events is also a function of the number of set bits.

For our analysis here, we utilize the notation and computations derived in [9]. Consider an RBF with exactly j_1 bits currently set. Let $\tau_k(j_1, j_2)$ be the probability that an arriving, unrecorded message's k hashes will result in the RBF having j_2 bits set. Normally, $0 \leq j_2 - j_1 \leq k$, but also included in this formulation is the case where the hashes cause the number of set bits to exceed σ, such that the RBF recycles. τ_k can be constructed easily as a recursive function of τ_{k-1}, and there are 4 variants, depending upon the properties of the k hash functions– whether a single message's hashes can map to identical bins (colliding), or if each one maps to a distinct bin (non-colliding)– as well as whether the RBF is retaining or non-retaining. We utilize this formula, but its derivation is clearly specified for all variants in §4 of [9]. We note that for a particular RBF utilizing memory of size M and k hash functions, the needed set of $\tau_{i,j}$ can be computed in $O(Mk^2)$ time.

Let $\psi_k(\ell, i)$ be the probability a k-hash function RBF with i included messages has ℓ bits set. ψ_k can be defined recursively as

$$\psi_k(\ell, i) = \begin{cases} \sum_{j=0}^{k} \psi_k(\ell - j, i-1)\tau_k(\ell - j, \ell) & \ell \leq \sigma \\ 0 & \ell > \sigma \end{cases} \tag{1}$$

with $\psi_k(0, 0) = 1$. Equation 1 makes the observation that ℓ bits set after i message arrivals must derive from the cases where $\ell - j$ bits were set for the $i - 1$st message, $0 \leq j \leq k$.[3]

We use ψ_k to define two useful functions that map likelihoods that are more easily calculated when conditioning on the number of bits set to their value when conditioned on the number of messages recorded in the RBF:

$$\phi_k(\ell|i) = \frac{\psi_k(\ell, i)}{\sum_{j \leq \sigma} \psi_k(j, i)} \tag{2}$$

$$\rho(i) = \sum_{\sigma - k < \ell \leq \sigma} \phi_k(\ell|i) \sum_{j > \sigma} \tau_k(\ell, j) \tag{3}$$

$$\gamma_k(i) = \sum_{0 \leq \ell \leq \sigma} \phi_k(\ell|i)\tau_k(\ell, \ell) \tag{4}$$

$\phi_k(\ell|i)$ is the conditional probability that, given i messages are recorded in the RBF, ℓ bits are set in the RBF; $\rho(i)$ is the probability that a new message triggers a recycle event (set bits exceed the σ-bound); $\gamma_k(i)$ is the "get lucky" event alluded to above.

4.1.2 Extending to the N-bounded case. If false positive rates are sufficiently low, then a recycling event can be approximated for the N-bounded case by simply transitioning downward only from state N to either state 0 (for non-retaining) or state 1 (retaining). As noted in [9], a real user would not be able to distinguish between a true repeat and and a new "get lucky" message.

However, in the interest of precision, we can define $\phi_k^m(j|i)$ to be the conditional probability that j messages are observed as setting bits out of the i distinct messages received. Then $\phi_k^m(j|i)$ has base case $\phi_k^m(0|0) = 1$ and satisfies recurrence relation $\phi_k^m(j|i) = \phi_k^m(j-1|i-1)(1 - \gamma_k(i-1) + \phi_k^m(j|i-1)\gamma_k(i-1)$.

[3]Note we do not explicitly use ψ to cover the case where where crossing the σ boundary. In such cases, i resets to 0 and $\psi_k(\ell, 1)$ is computed using the base case $\psi_k(0, 0) = 1$.

4.1.3 Computing Steady State rates. Note that the set of transitions consists of self-loops, transitions from state i to $i+1$, and transitions from many of the states i back to state 0 (non-retaining) or 1 (retaining). In theory, there can be an infinite number of states, given that new messages can arrive, and some messages may fail to set bits (a false positive for new messages, a "get lucky" event for non-repeat messages). Note that the likelihood of recording i messages starts to drop rapidly as i grows: the steady state probability is certainly bounded by $1-\rho(i-1)$. To ensure an error less than some ϵ, we cut the process at state i for which $\epsilon > 1-\rho(i-1)$. We begin by choosing a default non-normalized value of 1 for the base state (state 0 for non-retaining, state 1 for retaining), and for each i, we compute its likelihood relative to that of its neighboring state. After performing this computation for all states, we sum the total weight of the states, and divide by this amount to normalize, such that $\sum_i \Pi_i = 1$.

4.1.4 Challenges Modelling Exact False Negative Rate. If we have $\widehat{f_n}[i]$, we see above that we can compute all transition probabilities, the false negative rate $f_n[i]$ of each state, and from there solve the false negative rate directly. Unfortunately, we see no easy way to directly compute $\widehat{f_n}[i]$ - we now describe why this is challenging. Define $R_{i,j}$ to be an indicator random variable that equals 1 when message j is one of the first i messages to be recorded (non-repeat or new) in the RBF, with $P_{i,j} = P(R_{i,j} = 1)$. In other words, when the Markov Model is in state i, a message sampled from $\boldsymbol{Q}$ will be a false negative only when $P(R_{i,j} = 0)$. Clearly we have both $P_{i,j}$ increasing with i and decreasing with j: more recorded messages increases the likelihood that message j is included, and more popular messages are more likely to be included than less popular ones.

However, computing $P_{i,j} = P(R_{i,j} = 1)$ is computationally expensive. To see this, consider a simple 4-message ($D = 4$) example of a distribution whose popularity probabilities are distinct. Working through the possible sample paths of a small example, we have that $P(R_{3,j} = 1) = q_j + \sum_{\ell \neq j} q_\ell [\frac{q_j}{1-q_\ell} + \sum_{k \neq j,\ell} \frac{q_k}{1-q_\ell} \frac{q_j}{1-q_\ell-q_k}]$. Note that for a given set of elements recorded, the likelihood of such a recording differs depending on the ordering of these messages. For instance, messages arriving in the order (0,1,2) occurs with probability $q_0 \frac{q_1}{1-q_0} \frac{q_2}{1-q_0-q_1}$ whereas arriving in the reverse order (2,1,0) has probability $q_2 \frac{q_1}{1-q_2} \frac{q_0}{1-q_2-q_1}$, i.e., the denominator varies. These complexities only worsens as i and j grow large, which they will for the RBFs used in practice.

4.2 Upper bounds, lower bound, and approximation

Rather than compute $\widehat{f_n}$ exactly, we explore various approaches to computing upper and lower bounds. It turns out that if $\widehat{f_n}[i]$ is upper (lower) bounded, then the corresponding false negative rate of the system is also upper (lower bounded). This is not as straightforward to show as it might seem. While it is true that $f_n = \sum_i \Pi_i f_n[i]$, such that increasing (decreasing) $\widehat{f_n}[i]$ will increase (decrease) $f_n[i]$, it also alters the steady state distribution of Π_i, since the transition probabilities are functions of $\widehat{f_n}[i]$. However, we are fortunate that these shift in a preferable direction– this is proven formally in §A of the Appendix.

We proceed by describing how, for various lower, upper, and approximation methods built atop the Markov Model, one can bound/approximate false negative rates for the system.

4.2.1 A (not-so-tight) lower bound. The Markov Model described above can be used to implement a fairly loose lower bound, $\widehat{f_n^l}$, on false negative Rate. Note that for a given i, the largest possible value for $P_{i,j} = P(R_{i,j} = 1)$ (hence the smallest possible $1 - P_{i,j}$) is $P^l_{i,j} = \sum_{\ell=0}^{i-1} q_\ell$, which is the case where the i most popular messages are recorded in the RBF. This lower-bounds the false negative

rate as:

$$\widehat{f_n^l}[i] \quad = \quad \left(\sum_{i \le j < D} q_j\right)\left(1 - \sum_{\ell=0}^{i-1} q_i\right) \tag{5}$$

In other words, by assuming for each state i that the i messages recorded are the ones with largest popularity, the likelihood of sampling a non-repeat (any but the i most popular messages) is minimized, and hence lower than what an actual sampling of i messages from $\mathcal{Q}$ might provide.

4.2.2 Sample-with-replacement upper bound. State i of the Markov Model represents the state where i distinct messages are recorded into the RBF, i.e., a sample set of messages can be derived with correct probability via sampling without replacement. Note that if instead, a sampling with replacement is used to select the set of messages recorded by state i, then the number of distinct recorded messages could also be less than i. Hence, we are simply bounding the sampling-without replacing bound of $P(R_{i,j} = 1)$ with the sampling-with-replacement bound of $P^s_{i,j} = (1 - q_j)^i$. This yields

$$\widehat{f_n^s}[i] \quad = \quad \left(\sum_{0 \le j < D} q_j (1 - q_j)^i\right) \tag{6}$$

In other words, when sampling with replacement, we select messages that are proportionately sampled correctly, but there may wind up being fewer than i distinct messages sampled when in state i. Thus, the likelihood of a false positive not being one of these $j \le i$ sampled messages will be higher than if i messages were in fact recorded.

4.2.3 Uniform distribution as an upper bound. The false negative rate f_n for an arbitrary distribution $\mathcal{Q}$ of D messages is upper-bounded by $\widehat{f_n^u} = f_n(M, k, \sigma, \mathcal{U}, p_r, \mathcal{N})$, where $\mathcal{U}_D$ is a uniform popularity distribution with D messages, each having sampling likelihood of $1/D$. This is proven formally in §C.

Determining the false negative rate when i messages are recorded for a uniform distribution is straightforward. We iterate over each message $j \in \mathcal{Q}$: $q_j = 1/D$ for all j, and $P^u_{i,j} = i/D$, i.e., each message has an equal likelihood of being one of the i recorded messages. Thus:

$$\widehat{f_n^u}[i] \quad = \quad \left(\sum_{0 \le j < D} (1 - i/D)/D\right) = (1 - i/D) \tag{7}$$

4.2.4 Bi-uniform distribution as tighter upper bound. The proximity of false negative rate of the uniform distribution to that of the underlying distribution worsens as the underlying distribution's variance increases. A tighter distribution can be obtained if one is willing to suffer greater but reasonable computational overhead by constructing a distribution with two uniform components, i.e., for index s and probability $p_h \ge 1/D$, we can construct such a distribution by assigning $q_j = p_h$ for $j < s$ and $q_j = p_l = (1 - sp_h)/(D - s) \le 1/D$ for $j \ge s$. The false negative rate is computed by summing over all cases of the jth message currently being sampled, determining the likelihood that it was not previously sampled. For state i, there are i other messages that must be drawn from this bi-uniform distribution. For the case where $q_j = p_h$, these other i messages must have been sampled from the set of $s - 1$ messages with sampling probability p_h and $D - s$ messages with sampling probability p_l. For the case were $q_j = p_l$, these other i messages must have been sampled from the set of s messages with probability p_h and $D - s - 1$ other messages with sampling probability p_l:

$$\widehat{f_n^b}[i] = \left(\sum_{0 \le j < s} q_j (1 - \sum_{\ell=0}^{i-1} p_h \binom{s-1}{\ell} {p_h}^{\ell} p_l^{i-1-\ell}) + \sum_{s \le j < D} q_j (1 - \sum_{\ell=0}^{i} p_l \binom{s}{\ell} {p_h}^{\ell} p_l^{i-1-\ell}) \right) \quad (8)$$

$$= \left(s \sum_{\ell=0}^{i-1} \binom{s-1}{\ell} {p_h}^{\ell+2} p_l^{i-1-\ell} + (D-s) \sum_{\ell=0}^{i} \binom{s}{\ell} {p_h}^{\ell} p_l^{i+1-\ell} \right) \quad (9)$$

Note that, unlike the uniform distribution, each possible pair of s and p_h maps to a distinct bi-uniform distribution, and not all of them necessarily upper bound the false negative rate of that of $\boldsymbol{Q}$. Fortunately, Corollary 5 in §D gives sufficient conditions on the bi-uniform distribution to ensure it provides an upper bound.

Lemma 1. *A false negative rate of a bi-uniform distribution $\mathcal{B}$ upper bounds $\boldsymbol{Q}$ when there is some $0 \le t < D$ for which the jth elements $q'_j \in \mathcal{B}$ and $q_j \in \boldsymbol{Q}$ satisfy $q'_j \ge q_j$ for all $j \le t$ and $q'_j < q_j$ for all $j > t$.*

Lemma 2. *Let $\mathcal{B}_1$ and $\mathcal{B}_2$ be two bi-uniform distributions that shift from high to low probabilities at the same index s, with the high probability, p_h^1 of $\mathcal{B}_1$ larger than that in $\mathcal{B}_2$ ($p_h^1 > p_h^2$). Then the false negative rate of $\mathcal{B}_1$ is lower than that of $\mathcal{B}_2$.*

Putting these two Lemmas together, we see that for each value of s, there is a specific $\mathcal{B}$ with maximal p_h that satisfies Lemma 1. Thus, we can find a tightest bi-uniform upper bound by iterating over all s for which $q_{s-1} \ge 1/D$, setting $p_h = q_{s-1}$, computing the false negative rate for each s, and choosing the minimum result as the tightest upper bound.

Note that this is the best we are able to currently do. Lemma 1 is a sufficient condition, but need not be necessary.[4]

4.2.5 A brief note on self-loops. One may note that when taking a self-loop, the Markov Model returns to the same state where the set of i items is "resampled", when in fact the underlying process would use the same set of i items as in the previous round. The likelihood of exiting the state turns out to not be the same. Fortunately, resampling actually upper-bounds the process of not resampling, such that we can use a resampling approach to upper bound a re-use approach. This is proven in the Appendix in Section B. Note this issue does not impact our lower bound, since we assume a static set of elements (the i most popular) when considering the lower bound, such that resampling in state i always remains static.

4.2.6 An approximate "drop pins" average-case analysis. Average case analyses are often good approximations of an underlying process, and false negatives are no exception. Here we distinguish the arrival a distinct message j when i messages have been recorded in the RBF. These i recorded messages were generated by sampling without replacement. Sampling i elements without replacement can be emulated by performing sampling with replacement (we refer to this sampling-with-replacement process as "dropping pins" to select elements, hence the name), and stopping after i distinct messages have been selected. The number of samples with replacement, X, is a random variable with some mean $E[X]$.

$E[X]$ can be determined by writing $X = \sum_\ell Y_\ell$ where Y_ℓ is an indicator that equals 0 when the ℓth sample is a repeat of a previous sample, and is 1 otherwise. Since the process is sampling with replacement, $E[Y_\ell] = P(Y_\ell = 1) = \sum_{0 \le j < D} q_j (1 - q_j)^{\ell - 1}$ (iterating over j for the case where the ℓth arrival is message j; the message has not been seen only when the previous $\ell - 1$ messages in the

[4]We hypothesized that a tighter could be achieved when a distribution $\mathcal{B}_1$ majorizes $\mathcal{B}_2$, but have found counterexamples.

sampling with replacement are a message other than j.) Using the rule that sum of expectations equals the expectation of sums, this can be simplified to:

$$E[X] = \sum_{0 \le j < D} q_j \frac{1 - {q_j}^{\ell}}{1 - q_j} \tag{10}$$

Note that $E[X]$ increases with ℓ (the number of new messages increases with the number of samples.) For our average-case analysis, we define $\ell^*(i)$ to be the value of ℓ for which $E[X]$ is closest to i . We now have an average "sampling with replacement" count, and can approximate the false negative rate as:

$$\widehat{f_n^a}[i] \quad = \quad \left(\sum_{0 \le j < D} (1 - q_j)^{\ell^*(i)} q_j \right) \tag{11}$$

4.2.7 Extending Markov Model to cover Two-phase RBF. Extending to cover a two-phase system is done by effectively considering the state of a frozen filter (i.e., which state i did it "freeze" in). We define F_i to be the steady state likelihood that the frozen filter remains frozen in state i:

$$F_i = \Pi_i \sum_{\ell=\sigma-k+1}^{\sigma} [\phi_k(\ell|i) \sum_{j=\sigma+1}^{\ell+k} \tau_k(\ell, j)] \tag{12}$$

i.e., we start from between $\sigma - k + 1$ and σ bits set, and the next message sends us over the edge and "freezes" the filter.

When an arriving message appears to have been previously recorded in the frozen filter, it should be classified by the BF as a repeat message. However, assuming it hashes to empty bits in the *active* filter, should these bits be filled? For the sake of our analysis, we assume yes. However, our simulation results also reveal that doing so lowers the false negative rate of the underlying two-phase system. The reasoning is simple: consider a very popular message arriving that is recorded in the frozen filter but not the active filter. If we decline recording this message in the active filter, the BF will identify it as a repeat until the active filter fills to capacity, and the frozen filter is cleared. This popular message will no longer be recorded and will, upon its almost assured imminent arrival, trigger a false negative. This assumption is important to our analysis because the set of messages recorded in the active filter is no longer dependent on the set of messages recorded in the frozen filter.

Since the configuration of the frozen filter has no effect on the message arrival and insertion process of the active filter, the 2D system that models the two-phase system can be represented as the product of two independent Markov Models. In particular, when the active filter consists of i recorded messages and the frozen filter has frozen upon j recorded messages, the false negative rate can be computed as:

$$f_n^2[i, j] \quad = \quad p_r \widehat{f_n}[i] \widehat{f_n}[j] (1 - \gamma_k(i))(1 - \gamma_k(j)) \tag{13}$$

These transitions are shown for state i of the active filter (where j is fixed for the figure) in Fig. 11.

The overall false negative rate is therefore:

$$f_n^2 = \sum_{i,j} \Pi_i F_j f_n^2[i,j] \tag{14}$$

$$= p_r \left(\sum_i \Pi_i \widehat{f_n}[i](1-\gamma_k(i)) \right) \left(\sum_j F_j \widehat{f_n}[j](1-\gamma_k(j)) \right) \tag{15}$$

.

Side note: the "red" transition out of state i of the active filter can be solved generally as: $p_r \widehat{f_n}[i](1-\gamma_k(i)) \left(\sum_j F_j \widehat{f_n}[j](1-\gamma_k(j)) \right)$.

5 RENEWAL MODEL

An alternative approach to obtaining the long term average false negative rate of the RBF is to model repeated message arrivals to the RBF as a renewal process. As messages arrive at the RBF, its bits fill and eventually it resets; we consider the resetting of the RBF to be the start of the renewal process.

5.1 One-Phase RBF, repeats only

Let us focus on a particular cycle. For that cycle, we define several random variables:

- η_j is an indicator r.v. equal to 1 when the jth arriving message has not been recorded (i.e., either new or a non-repeat) such that it may set bits.
- $\mathcal{B}_{\ell,j}$ is an indicator r.v. equal to 1 when, after j message arrivals, ℓ bits are set in the RBF. We count all messages, including repeated messages.
- $\mathcal{A}_j$ to be an indicator r.v. that is set to 1 when the jth arrival occurs within the given renewal interval (i.e., after j arrivals, we still have $\leq \sigma$ bits set in the RBF). It includes all of new, repeat and non-repeat arrivals.
- C_j is an indicator r.v. that is set to 1 when the jth arrival occurs within the given renewal interval, and it induces a false negative.

A single cycle's false negative rate is computed simply as $\sum_{j=1}^{\infty} C_j / \mathcal{A}_j$. We can compute the average false negative rate over N cycles by extending our definition such that $\mathcal{A}_j^n$ and C_j^n are 1 respectively when the jth arrival respectively occurs and counts within the nth cycle. Then the average false negative rate over these N cycles can be written as $\sum_{n=1}^{N} \sum_{j=1}^{\infty} C_j / \sum_{n=1}^{N} \sum_{j=1}^{\infty} \mathcal{A}_j$, i.e., the total number of false negative events observed over the total number of messages received during these N cycles. Dividing numerator and denominator by N, we see that this is simply the average number of false negatives per cycle divided by the average number of message arrivals per cycle, and letting $N \to \infty$, we have:

$$f_n = \frac{E[\sum_{j=1}^{\infty} C_j]}{E[\sum_{j=1}^{\infty} \mathcal{A}_j]} \tag{16}$$

$$= \frac{\sum_{j=1}^{\infty} E[C_j]}{\sum_{j=1}^{\infty} E[\mathcal{A}_j]} \tag{17}$$

We solve for $E[\mathcal{A}_j]$ and $E[C_j]$ as follows:

$$P(\eta_j = 1) = \sum_{i=1}^{D} (1 - q_i)^{j-1} q_i \tag{18}$$

$$P(\mathcal{B}_{\ell,j} = 1) = \begin{cases} (1 - P(\eta_j = 1))P(\mathcal{B}_{\ell,j-1} = 1) + \\ P(\eta_j = 1) \sum_{i=0}^{k} P(\mathcal{B}_{\ell-i,j-1} = 1)\tau_k(\ell - i, \ell) & \ell \leq \sigma \\ 0 & l > \sigma \end{cases} \tag{19}$$

Equation (19) bears resemblance to (1), the difference being that the former considers all arriving messages while the latter considers only bit-setting messages. The $(1 - P(\eta_j = 1))\mathcal{B}_{\ell,j-1}$ covers the case where the message is already recorded and hence sets no bits, and the other part matches (1).

$$P(\mathcal{A}_j = 1) = \sum_{\ell=0}^{\sigma} P(\mathcal{B}_{\ell,j-1} = 1) \tag{20}$$

$$P(C_j = 1) = \sum_{\ell=0}^{\sigma+k} P(\mathcal{B}_{\ell,j-1} = 1)P(\eta_j = 1)(1 - \tau_k(\ell, \ell)) \tag{21}$$

$$= P(\eta_j = 1) \sum_{\ell=0}^{\sigma+k} P(\mathcal{B}_{\ell,j-1} = 1)(1 - \tau_k(\ell, \ell)) \tag{22}$$

$\mathcal{A}_j$ is 1 when the number of bits set in the RBF is no larger than σ after the $j-1$st message arrival. C_j is 1 when the number of bits is no larger than σ after the $j-1$st arrival, the arriving message i did not arrive in any previous cycle, and did not "get lucky". Noting that because $\mathcal{A}_j$ and C_j are indicator r.v.s, we have that $E[\mathcal{A}_j] = P(\mathcal{A}_j = 1)$ and $E[C_j] = P(C_j = 1)$.

Note a subtle detail: in (21), ℓ iterates between 0 and $\sigma + k$. This is a subtle detail that is to $\mathcal{A}_j$ and $\mathcal{B}_{\ell,j-1}$ not being independent. As shown in Theorem 7, this is a sufficient upper bound to cover this correlation.

For the purposes of the upper bound, we note that any finite approximation of this sum satisfies:

$$\frac{\sum_{j=1}^{m} E[C_j]}{\sum_{j=1}^{m} E[\mathcal{A}_j]} \geq \frac{\sum_{j=1}^{\infty} E[C_j]}{\sum_{j=1}^{\infty} E[\mathcal{A}_j]} \tag{23}$$

and hence provides an upper bound that can be computed over a finite number of terms. The intuition behind why this is an upper bound is that we are effectively only considering the first m message arrivals in each cycle, and since the RBF records more messages during the duration of the cycle, each successive message arrival is less likely to induce a false negative than those that came before. A more formal proof is presented in §E.1.

Further note that as m increases, the upper bound tightens. To determine an appropriate point to terminate the computation (i.e., the value of m used), we choose a small ϵ, and we iterate over j until $\sum_{\ell>\sigma} \mathcal{B}(\ell, j) > 1 - \epsilon$, i.e., until the likelihood that the recycling event has yet to happen is less than ϵ.

5.2 Extension for New Arrivals

We modify equations (18)-(21) to account for new arrivals as well. First, we note that for the jth arrival, the arriving message must be a non-repeat. Non-new messages arrive with probability p_r, but whether the message i is a non-repeat does not simply depend on whether message i arrived earlier in the sequence. Previous new arrivals may have altered the position of what is currently message i; it may have previously been message i' and $i - i'$ new arrivals falling in front of the

message in the distribution could have downshifted it. Additionally, message i itself may be a new arrival. We define $h_i(j)$ to be an indicator describing the likelihood that message i is "disabled" after the jth arrival, meaning that it does not count as a repeat arrival, either because it has previously been sampled in the current cycle, or because it is a new message. $h_i(j)$ satisfies the recurrence:

$$\begin{aligned} h_i(j) &= p_r\left[h_i(j-1) + (1-h_i(j-1))q_i\right] + \\ &\quad (1-p_r)\left[d_i + h_{i-1}(j-1)\sum_{i'<i} d_{i'} + h_i(j-1)\sum_{i'>i} d_{i'}\right] \end{aligned} \tag{24}$$

The first part of this term is the case of a repeat arrival. The ith message is disabled if it was previously disabled, or if it was not previously disabled but arrives in the jth iteration. The second part of the term is the case of a new arrival. The ith message is disabled one of three cases hold:

- Case 1: the new message arrives in the ith position (with probability d_i)
- Case 2: the new message arrives in some position $i' < i$ such that all messages indexed $i'' \geq i'$ have their index incremented. Hence, if the message in position $i-1$ were disabled, then i is now disabled.
- Case 3: the new message arrives in some position $i' > i$, such that message i's position in the distribution is unaffected. Then if message i was already disabled, it remains disabled.

Using this recursion, we can formulate equations for $\mathcal{B}_{\ell,j}$. $\mathcal{A}_j$ is adjusted simply by using this reformulation of $\mathcal{B}_{\ell,j}$. C_j utilizes the reformulated $\mathcal{B}_{\ell,j}$ as well, but also needs other adjustments:

$$P(\eta_j = 1) = (1-p_r) + p_r\sum_{i=1}^{D} h_i(j) \tag{25}$$

$$P(C_j = 1) = p_r\left[\sum_{\ell=0}^{\sigma+k} P(\mathcal{B}_{\ell,j-1}=1)(1-\tau_k(\ell,\ell))\right]\left[\sum_{i=1}^{D}(1-h_{j-1}(i))q_i\right] \tag{26}$$

$\mathcal{B}_{i,j}$ is modified indirectly via the redefinition of η_j which now accounts for new arrivals in addition to non-repeats as events that may set bits. C_j can only evaluate to 1 for a non-new arrival (new arrivals cannot be false negatives). Furthermore, when element i is sampled, rather than simply assuring element i was not previously sampled, we ensure that element i was not disabled.

5.2.1 Two-Phase RBF. We can apply a similar Renewal Model to a two-phase RBF, but with the renewal cycle lasting for both the active and frozen filter to reach capacity. For arrival j to the active filter in a given renewal interval, say the frozen filter stopped after m arrivals. Then the false negative probability of the next arrival must take into account the previous $j + j'$ messages. Let $\zeta_{j'} = \sum_{\ell=0}^{\sigma+k}\left[P(\mathcal{B}_{\ell,j'-1}=1) - P(\mathcal{B}_{\ell,j'}=1)\right](1-\tau_k(\ell,\ell))$ be the probability the active filter stops (freezes) after j' arrivals. The exact expression for the two-phase RBF false negative rate is given by:

$$f_n^2 = p_r\frac{\sum_{j=1}^{\infty}\left[\sum_{\ell=0}^{\sigma+k} P(\mathcal{B}_{\ell,j-1}=1)(1-\tau_k(\ell,\ell))\left(\sum_{j'=1}^{\infty}\zeta_{j'}P(\eta_{j+j'}=1)\right)\right]}{\sum_{j=1}^{\infty}P(\mathcal{A}_j=1)} \tag{27}$$

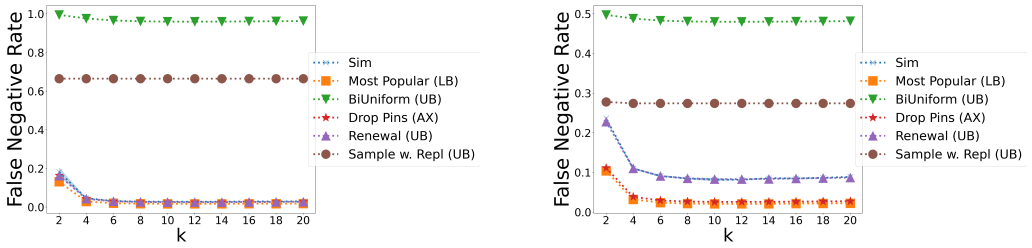
This involves an infinite some over both j and j', but the same property of the sum approaching the exact value from above as $j + j'$ increases holds, and once more the bound quickly converges in tightness in practice. With a bound m over the set of $j + j'$, this can be efficiently computed by

iterating over a variable $s = j + j'$, the numerator of (27):

$$p_r \sum_{s=0}^{m} \sum_{j=0}^{s} \sum_{\ell=0}^{\sigma+k} P(\mathcal{B}_{\ell,j-1} = 1)(1 - \tau_k(\ell, \ell))\zeta_{s-j} P(\eta_s = 1). \tag{28}$$

Computationally, we iterate over j and compute $\sum_{\ell=0}^{\sigma+k} P(\mathcal{B}_{\ell,j-1} = 1)(1 - \tau_k(\ell, \ell))$, ζ_j, and η_j, storing them in a table until we decide which m is sufficiently large (using the same stopping criteria as in the one-phase version). Then we iterate over s and and j and combine the corresponding entries stored in the table.

6 ANALYTICAL MODEL VALIDATION AND RESULTS



(a) $p_r = 1$ (b) $p_r = 0.5$

Fig. 2. Bounds, Approximations and Simulations

6.1 Model Accuracy, Comparisons and Trends

We verify the accuracy of our bounds and approximation through discrete event-driven Python simulations. We simulated sequences of random message arrivals to a RBF, computing False Positive and False Negative rates. All randomness was handled by the standard python `random` library [5]. In Fig. 2, we show results of a simulation for a One-Phase RBF, with $M = 1000$ bits; the distribution Q is a Zipf of 1000 elements and parameter $\alpha = 2$. As additional evidence for the accuracy of our results, we carry out multiple simulation runs and compute 99% confidence intervals on the collection of results, indicated by the shaded areas around the plotted sim curves.

In all cases, the results of the Renewal Model are closer to simulation than the Markov Model; this is not surprising as the Renewal Model can be tuned to be arbitrarily close to the exact false negative rate, and furthermore takes explicit account of the distribution $\mathcal{N}$.

Figure 2(a) shows the case of low new arrival rates ($p_r = 1$) and (b) the case of high new arrival rates ($p_r = 0.5$). The Lower Bound and Drop Pins Approximation closely track simulation much like the Renewal Bound for low new arrival rates, but are less accurate for higher ones. The relative effect of (not) including Q in modelling is illustrated by this result.

6.2 Computation Time Comparison

In Fig. 3 we plot the computation time for our bounds and approximations as well as the time to simulate 10 runs of 100,000 message arrivals each on a standard MacBook Pro (2 GHz Intel i5 processor, 16 GB RAM). The Renewal and bi-uniform bounds become dramatically more expensive than the rest as σ grows larger; thus less accurate bounds like Sampling With Replacement and the Drop Pins Approximation still have some utility if we need to estimate bounds for large-sized

[5]All the code to generate the Figures and Tables in §6 and §7 will be shared on a GitHub repository https://github.com/kadzier/Recycling-Bloom-Filters/tree/kahlil-summer2022

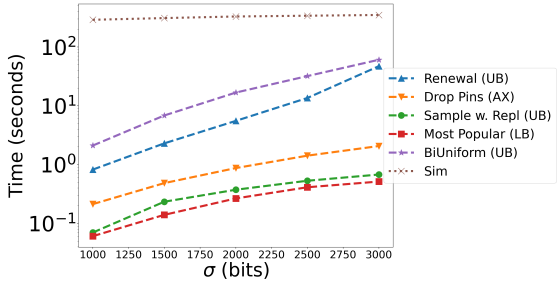


Fig. 3. Computation Time Comparison for a RBF with $M = 10000, k = 5, p_r = 0.9$.

σ	Renewal Time (seconds)	Approx. Time (seconds)	Approx. % error
1000	0.823	0.213	0.4%
2000	5.541	0.874	0.1%
3000	47.148	2.07	0.4%

Table 1. Accuracy of Approximation and Relative Computation Time

filters. This point is further illustrated in Table 1; in the most extreme case, computing the Drop Pins approximation is over is over 20x faster than computing the Renewal Bound, with a sacrifice in accuracy of less than 1%.

6.3 Trade-off Between Filter Parameters, False Positives and False Negatives

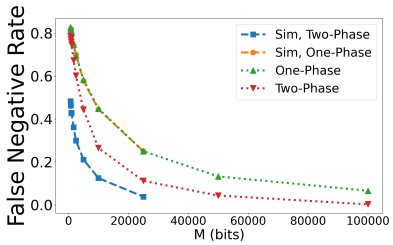
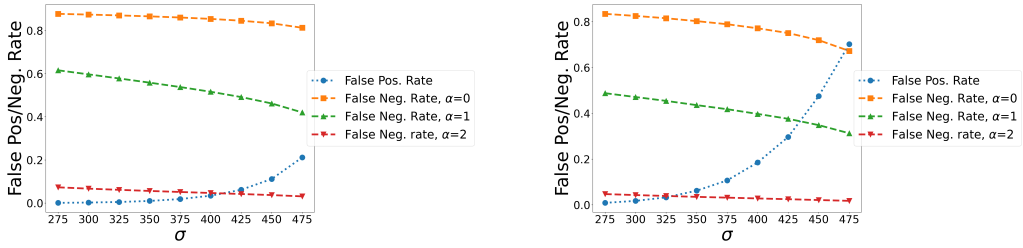


Fig. 4. Best achievable False Negative rate vs. total filter size M

In Fig. 4, we show how varying the maximum filter size M impacts the best achievable false negative rate for a fixed false positive rate of .0001 (which constrains σ). Q is a Zipf distribution with $N = 1000$ elements and $\alpha = 1$, with a repeat message arrival rate of $p_r = 0.9$. For $M < 25000$, we also plot corresponding simulations. Increasing the filter size is seen to have a dramatic effect on False Negative rates for smaller filters, and effect that gradually diminishes as the filter grows larger.

Fig. 5 has a fixed filter size $M = 500$, and we vary σ instead. We plot two different arrival processes, both Zipf distributions, with alpha parameters of 0, 1 and 2 with $p_r = 0.9$. One can observe two trends. First, the message arrival distribution does not impact the false positive rate at all. The false negative rate, however, is significantly impacted by the arrival distribution, with more "uniform" arrivals leading to higher false negative rates. Second, the average False Positive rate exponentially increases with RBF capacity, while the corresponding false negative rate linearly decreases. When designing RBFs to balance false positive and false negative rates one can keep this trend in mind; for RBFs operating near their σ capacity, dramatic improvements to average false positive rates can be achieved by slightly reducing the capacity, without unduly impacting the false negative rate.



(a) One-Phase RBF (b) Two-Phase RBF

Fig. 5. False Positive and False Negative trends, varying RBF capacity σ.

Summarizing, if a RBF user wants to minimize both their false positive and false negative rates, to some extent, the largest lever (the arrival distribution) is out of the user's individual control. But given a distribution, one can then vary σ to find an acceptable trade-off between False Positive and False Negative rates.

6.4 One Phase vs. Two Phase Filter

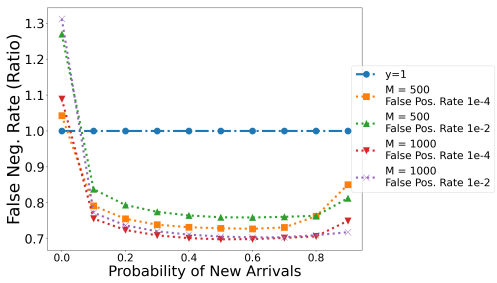


Fig. 6. False Negative rate, One-Phase RBF vs. Two-Phase RBF for a new arrivals probability of 0.5. Two-Phase clearly performs better.

In Fig. 6, we investigate the performance of the One-Phase vs. the Two-Phase RBF. Total filter memory $M = 1000$ and false positive rates were fixed at .01 and .0001. For high arrival rates, the one-phase filter beats out the two-phase; this switches for low arrival rates. The exact "crossover point" depends on parameters such as false positive rate tolerance and M.

7 TWO USE-CASES ON REAL WORKLOADS

In this section, we examine the tightness of our upper bound through the study of two distinct real-world workloads. The first workload focuses on traffic from an end-user perspective of interest in scenarios where an ISP seeks insight into the Internet access patterns of its users at a fraction of the cost necessary to run Netflow[6]. The second workload views traffic from a CDN perspective, with the objective of serving users as efficiently as possible. In particular, CDNs have in the past used Bloom Filters to inform their content caching strategies, ultimately enhancing content delivery efficiency and reducing latency. This technique involves caching content only if it hits both phases of a recycling Bloom Filter to avoid caching unlikely one-hit wonders ([17] provides an in-depth discussion of this setup).

The key characteristics of the user-centered dataset include significant temporal correlations, where specific requests predictably lead to a series of subsequent requests. This pattern is indicative of (i) specific user behaviors and (ii) widespread interactions between internet services, which can be highly sequential. The second dataset also demonstrates temporal correlations, highlighting patterns in how content is requested and cached over time. These temporal dynamics break our initial assumption but make the upper-bound more conservative as we discuss in Section 7.3.

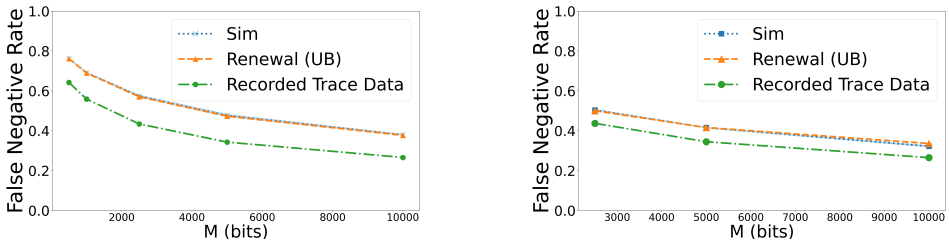


Fig. 7. Renewal Bound, Simulation, and Recorded Trace Data, (a) One-Phase RBF, (b) Two-Phase RBF.

7.1 User-centered Application

Dataset description: To evaluate the tightness of our upper-bound in practical scenarios, we compiled data on all IP flows to and from 13 residential buildings on Columbia's network for two days in May 2023 [6]. This resulted in a total of 35857439 flows, with 227668 unique flows to a specific /24 identified. The dataset has been fully anonymized to guarantee the privacy of the users. We designed a RBF to efficiently identify duplicate flows and avoid storing redundant ones.

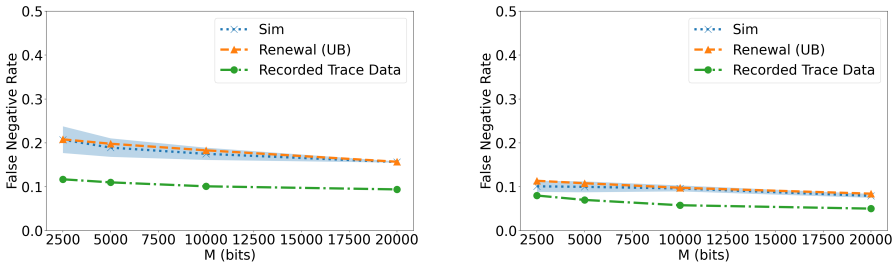
The effect of correlation: A challenge in analyzing this dataset lies in the presence of temporal dependencies among the samples. Typically, user interactions with web applications initiate a sequence of related network flows. These flow patterns can change as CDNs adjust traffic (e.g., via DNS [21]) to manage datacenter load. Such sequences have been previously been used to fingerprint applications and deduce user behaviors [23]. While these dependencies complicate the mathematical analysis, they actually benefit caching by decreasing both false negatives and false positives. Thus, our upper-bound remains valid, albeit potentially less tight than in our simulations; we talk about this phenomenon in greater detail in §7.3.

Results: From the flow statistics, we can construct the equivalent of Q and $\mathcal{N}$ in the Renewal Model. Because we have the entire flow record, we can compute the real false positive and negative rates of our RBF once the parameters M, σ and k have been specified. In an effort to show the performance for uncorrelated samples, we also study the result of simulations where messages are sampled according to Q. In particular, we are interested in answering two main questions: (1) Given a specific memory size and false positive rate, what is the best negative rate that we can provably accomplish, and (2) Given a requirement of false positive and negative, what are the parameters of the resulting RBF k, σ, and M?

In Figure 7, we answer (1) by showing the best achievable false negative rate under our Renewal upper-bound, alongside results from our simulated uncorrelated scenario, and actual recorded data using an RBF with varying memory capacities and a fixed 0.0001 false positive rate. As we had hypothesized, the upper bound is less tight for the recorded trace data but still holds.

[6]This dataset has been used in a recent publication but cannot be shared in its current form because of its sensitive nature [14]

The answer to (2) is somewhat more open-ended; our analysis finds that, depending on the message arrival distribution, there may be a range of k, σ and M values that can meet the False Positive and False Negative constraints. Generally, we find that increasing M independently of the other parameters increases the range of False Negative rates that can be achieved, while keeping the range of achievable False Positive rates consistent.



(a) Mixed, Two-Phase (b) Web, Two-Phase

Fig. 8. Renewal Bound, Simulation, and JEDI for web traffic and mixed-traffic.

7.2 CDN-centered Application

Dataset description: As obtaining real traces from CDN is challenging for privacy and security reasons, we rely on a state-of-the-art synthetic trace generation tool named Jedi [25]. Using Jedi, we generate traces representative of diverse object types directly based on data observed in Akamai's CDN traffic. We concentrate on two types of traces: a mixed trace containing a variety of traffic from multimedia sources, and a trace exclusively focused on web content [7]. For both types, we simulate 1 million requests at a rate of 10 requests per second, using the first 5M requests to populate the cache and the remaining 5M to conduct our analysis. A significant observation in both traces was the prevalence of "one-hit wonders," with over 60% of entries appearing only once. False positives and false negatives both harm the user experience, with false negatives potentially having a more detrimental effect by failing to cache popular content that meets the caching threshold.

Results: In Figure 8, we show that our upper-bound is tight for web traffic and less so for mixed-traffic. To measure this relationship, we examine the frequency at which requests cluster together (as opposed to occurring independently). We create a Markov Chain model based on request sequences, and calculate the average entropy of its transition probabilities. Although this approach does not directly measure the sequence's "predictability", it allows us to study the number of instances where an element is followed by many elements with equiprobability. In particular, a completely sequential trace, where every element is followed by the same element, would have a full sequentiality and average entropy equal to 0. Mixed traffic shows lower entropy (≈ 0.51) than web traces (≈ 0.59) due to the sequential nature of requests for certain applications, but more than the user-focused dataset (≈ 0.39).

7.3 Discussion

Correlations in time-series: The real workload examples highlight a promising method to refine our bounds: dynamically identifying correlations. Our current approach is based on the premise that the arrivals have stabilized into a stationary distribution. However, in practical scenarios, no two cycles are the same; our data continuously evolves, with elements both fading and emerging

[7]We use the 'EU' traffic class to generate a synthetic trace that is representative of the original trace obtained from an Akamai server that is serving a mix of traffic classes and the 'W' traffic class for the web content traffic

with different existing correlations. We have assumed homogeneity within each cycle for analytical simplicity. Yet this condition is not essential; we believe a more precise upper bound could be achieved by relaxing this assumption. This improvement might require more sophisticated analytical tools, which we plan to explore in future work.

8 RELATED WORK

Recycling Bloom Filters: Our work focuses on RBFs. These have appeared as a cost-effective and simpler solution than Bloom Filter variants with deleting capabilities such as the Counting BF [11], Quotient BF [2], Cuckoo Filter [10], the Ternary Bloom Filter [15] and the Deletable Bloom Filter [24]. A related variant type uses the concept of a "sliding" window of fixed width [22], [1], [16], and can guarantee an absence of false positives and false negatives within the window as well as a maximum false positive rate for never-before seen messages. However, these variants impose no constraints on previously seen messages outside of the window, and the false negative rates are not analyzed. RBFs have demonstrated their utility in various applications, including managing authorized traffic in Software-Defined Networks (SDN) [19], detecting DDoS attacks [7], ensuring consistent load balancing in data centers [20], attributing payload [26], and facilitating routing in Information-Centric Networks (ICN) [18]. Akamai made a specific mention of a two-phase RBF in [17] which was a particular motivation for this work.

False Negative Rates: There has been a considerable amount of literature studying false-positive for Bloom Filters [3–5, 13]. However, false negatives have been significantly less studied as they only occur in certain variants of Bloom Filters. Guo *et al.* demonstrated that Counting Bloom Filters could suffer from false negatives and provided analytical bounds on the number of false negatives in that instance [12]. In a different context, Donnet *et al.* introduces a Bloom Filter variant that selectively deletes entries that create a high false positive rate at the expense of false negatives [8]. Similarly, Kleyko *et al.* introduces the "autoscaling" Bloom Filter that automatically adjusts its capacity with probabilistic bounds on false positives, true positives and false negatives. Our study is unique in that it is, to the best of our knowledge, the first to investigate the false negative rates associated with recycling in Bloom Filters. A recent paper has shown that these recycling Bloom Filters result in slightly different false positive bounds than in the literature as well [9].

9 CONCLUSION

To date, a formal analysis of False Negative rates in Recycling Bloom Filters has not been realized; we provide this analysis by deriving approximations, upper bounds and lower bounds of False Negative rates for several variants of RBFs.

We validate our analysis through both simulation and comparison to real-world trace data, and find that Renewal Models work quite well as a general-purpose upper bound for False Negative rates in a wide variety of message arrival distribution scenarios. However, in the case of low rates of new message arrivals, we find Markov Models can be a computationally tractable alternative. Our analysis enables users of RBFs to more effectively select filter design parameters and balance false positive and false negative considerations to achieve optimal performance in applications.

10 ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant Nos. CNS-1910138, CNS-2106197, and CNS-2148275. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. We also wish to thank Shuyue Yue, Ethan Katz-Bassett, and Gil Zussman, who are funded in part by NSF grant OAC-2029295, for providing the user-centered dataset referenced in Section 7.

REFERENCES

[1] Eran Assaf, Ran Ben Basat, Gil Einziger, and Roy Friedman. 2018. Pay for a sliding bloom filter and get counting, distinct elements, and entropy for free. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, IEEE, Honolulu, HI, 2204–2212.

[2] Michael A. Bender, Martin Farach-Colton, Rob Johnson, Russell Kraner, Bradley C. Kuszmaul, Dzejla Medjedovic, Pablo Montes, Pradeep Shetty, Richard P. Spillane, and Erez Zadok. 2012. Don't Thrash: How to Cache Your Hash on Flash. *Proc. VLDB Endow.* 5, 11 (jul 2012), 1627–1637. https://doi.org/10.14778/2350229.2350275

[3] Burton H Bloom. 1970. Space/time trade-offs in hash coding with allowable errors. *Commun. ACM* 13, 7 (1970), 422–426.

[4] Prosenjit Bose, Hua Guo, Evangelos Kranakis, Anil Maheshwari, Pat Morin, Jason Morrison, Michiel Smid, and Yihui Tang. 2008. On the false-positive rate of Bloom filters. *Inform. Process. Lett.* 108, 4 (2008), 210–213.

[5] Ken Christensen, Allen Roginsky, and Miguel Jimeno. 2010. A new analysis of the false positive rate of a bloom filter. *Inform. Process. Lett.* 110, 21 (2010), 944–949.

[6] Benoit Claise. 2004. *Cisco Systems NetFlow Services Export Version 9.* RFC 3954. Internet Engineering Task Force. https://www.ietf.org/rfc/rfc3954.txt Accessed: [Insert access date here].

[7] Biplob Debnath, Sudipta Sengupta, Jin Li, David J Lilja, and David HC Du. 2011. BloomFlash: Bloom filter on flash-based storage. In *2011 31st International Conference on Distributed Computing Systems*. IEEE, IEEE, Minneapolis, MN, USA, 635–644.

[8] Benoit Donnet, Bruno Baynat, and Timur Friedman. 2006. Retouched bloom filters: allowing networked applications to trade off selected false positives against false negatives. In *Proceedings of the 2006 ACM CoNEXT conference*. ACM, Lisboa Portugal, 1–12.

[9] Kahlil Dozier, Loqman Salamatian, and Dan Rubenstein. 2024. Modeling Average False Positive Rates of Recycling Bloom Filters. In *IEEE Infocom*. IEEE, Vancouver, CA.

[10] Bin Fan, Dave G Andersen, Michael Kaminsky, and Michael D Mitzenmacher. 2014. Cuckoo filter: Practically better than bloom. In *Proceedings of the 10th ACM International on Conference on emerging Networking Experiments and Technologies*. ACM, Sydney Australia, 75–88.

[11] Li Fan, Pei Cao, J. Almeida, and A.Z. Broder. 2000. Summary cache: a scalable wide-area Web cache sharing protocol. *IEEE/ACM Transactions on Networking* 8, 3 (2000), 281–293. https://doi.org/10.1109/90.851975

[12] Deke Guo, Yunhao Liu, Xiangyang Li, and Panlong Yang. 2010. False Negative Problem of Counting Bloom Filter. *Knowledge and Data Engineering, IEEE Transactions on* 22 (06 2010), 651 – 664. https://doi.org/10.1109/TKDE.2009.209

[13] Florian Klingler, Reuven Cohen, Christoph Sommer, and Falko Dressler. 2018. Bloom hopping: Bloom filter based 2-hop neighbor management in VANETs. *IEEE Transactions on Mobile Computing* 18, 3 (2018), 534–545.

[14] Thomas Koch, Shuyue Yu, Sharad Agarwal, Ethan Katz-Bassett, and Ryan Beckett. 2023. PAINTER: Ingress Traffic Engineering and Routing for Enterprise Cloud Networks. In *Proceedings of the ACM SIGCOMM 2023 Conference* (<conf-loc>, <city>New York</city>, <state>NY</state>, <country>USA</country>, </conf-loc>) *(ACM SIGCOMM '23)*. Association for Computing Machinery, New York, NY, USA, 360–377. https://doi.org/10.1145/3603269.3604868

[15] Hyesook Lim, Jungwon Lee, Hayoung Byun, and Changhoon Yim. 2016. Ternary bloom filter replacing counting bloom filter. *IEEE Communications Letters* 21, 2 (2016), 278–281.

[16] Yang Liu, Wenji Chen, and Yong Guan. 2013. Near-optimal approximate membership query over time-decaying windows. In *2013 Proceedings IEEE INFOCOM*. IEEE, Turin, Italy, 1447–1455. https://doi.org/10.1109/INFCOM.2013.6566939

[17] Bruce M Maggs and Ramesh K Sitaraman. 2015. Algorithmic nuggets in content delivery. *ACM SIGCOMM Computer Communication Review* 45, 3 (2015), 52–66.

[18] Ali Marandi, Torsten Braun, Kavé Salamatian, and Nikolaos Thomos. 2017. BFR: A bloom filter-based routing approach for information-centric networks. In *2017 IFIP Networking Conference (IFIP Networking) and Workshops*. IFIP, Stockholm, Sweeden, 1–9. https://doi.org/10.23919/IFIPNetworking.2017.8264842

[19] Luke McHale, Jasson Casey, Paul V Gratz, and Alex Sprintson. 2014. Stochastic pre-classification for SDN data plane matching. In *2014 IEEE 22nd International Conference on Network Protocols*. IEEE, IEEE, Raleigh, NC, USA, 596–602.

[20] Rui Miao, Hongyi Zeng, Changhoon Kim, Jeongkeun Lee, and Minlan Yu. 2017. SilkRoad: Making Stateful Layer-4 Load Balancing Fast and Cheap Using Switching ASICs. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication* (Los Angeles, CA, USA) *(SIGCOMM '17)*. Association for Computing Machinery, New York, NY, USA, 15–28. https://doi.org/10.1145/3098822.3098824

[21] Giovane C. M. Moura, John Heidemann, Ricardo de O. Schmidt, and Wes Hardaker. 2019. Cache Me If You Can: Effects of DNS Time-to-Live. In *Proceedings of the Internet Measurement Conference* (Amsterdam, Netherlands) *(IMC '19)*. Association for Computing Machinery, New York, NY, USA, 101–115. https://doi.org/10.1145/3355369.3355568

[22] Moni Naor and Eylon Yogev. 2013. Sliding bloom filters. In *International Symposium on Algorithms and Computation*. Springer, Springer, Hong Kong, China, 513–523.

[23] Fatemeh Rezaei and Amir Houmansadr. 2021. FINN: fingerprinting network flows using neural networks. In *Annual Computer Security Applications Conference*. ACM, New York, NY USA, 1011–1024.
[24] Christian Esteve Rothenberg, Carlos AB Macapuna, Fábio L Verdi, and Mauricio F Magalhaes. 2010. The deletable Bloom filter: a new member of the Bloom family. *IEEE Communications Letters* 14, 6 (2010), 557–559.
[25] Anirudh Sabnis and Ramesh K. Sitaraman. 2022. JEDI: model-driven trace generation for cache simulations. In *Proceedings of the 22nd ACM Internet Measurement Conference* (Nice, France) *(IMC '22)*. Association for Computing Machinery, New York, NY, USA, 679–693. https://doi.org/10.1145/3517745.3561466
[26] Kulesh Shanmugasundaram, Hervé Brönnimann, and Nasir Memon. 2004. Payload attribution via hierarchical bloom filters. In *Proceedings of the 11th ACM conference on Computer and communications security*. ACM, New York, NY, USA, 31–41.
[27] David Starobinski, Ari Trachtenberg, and Sachin Agarwal. 2003. Efficient PDA synchronization. *IEEE Transactions on Mobile Computing* 2, 1 (2003), 40–51.

A UPPER LOWER BOUND RESULT

We formulate a Markov Model that we refer to as a *red-black Markov Model* because it contains both black and red transitions, and we will be interested in measuring the respective rates at which red transitions are taken in two particular variants. The description of an N-state red-back Markov Model $\mathcal{BM}$ is as follows:

- Each state i's outgoing transitions are colored red or black, and their probability weights are described by the following variables: $x_i(\mathcal{BM})$, $y_i(\mathcal{BM})$, and for each other state $j \neq i$ in $\mathcal{BM}$, $p_{i,j}(\mathcal{BM})$ such that $\sum_{j\neq i} p_{i,j}(\mathcal{BM}) = 1$. We drop the $\mathcal{BM}$ when its reference is unique.
- Each state i takes a self-loop transition with probability $(1 - x_i)y_i$. This transition is black.
- When not taking a self-loop, it exits the state, and conditioned on exiting it goes to state $j \neq i$ with probability $p_{i,j}$. It is important to emphasize that the likelihood that, when transiting out of state i, the choice of next state is only a function of the $p_{i,j}$, and specifically not of the values of x_i and y_i.
- The transition from i to j can be a red transition or a black transition. The transition is black with (not conditioned) probability $p_{i,j}(1 - y_i)$, leaving the probability of $p_{i,j}x_iy_i$ for the red transition. Note $(1 - x_i)y_i + \sum_{j\neq i} p_{i,j}(1 - y_i) + \sum_{j\neq i} p_{i,j}x_iy_i = 1$, i.e., the sum of all probability transitions is 1.

Define $\mathcal{R}(\mathcal{BM})$ to be the fraction of time red transitions are taken in the steady state system described by $\mathcal{BM}$. This is clearly $\mathcal{R}(\mathcal{BM}) = \sum_i \Pi_i \sum_j p_{i,j}x_iy_i$ in the steady state.

An alternate but equivalent sample path definition is:

$$\lim_{n\to\infty} \sum_{P\in\{P\}} meas(P)C_n(P)/n$$

where $\{P\}$ is the set of all possible sample paths of some length $m \geq n$, $meas(P)$ is the likelihood that path P is taken, and $C_n(P)$ is the number of red transitions seen during the first n transitions of path P.

Theorem 3. *Let $\mathcal{BM}_1$ and $\mathcal{BM}_2$ be a pair of red-black Markov Models that are identical except that for each state i, $x_i(\mathcal{BM}_1) \geq x_i(\mathcal{BM}_2)$. Then $\mathcal{R}(\mathcal{BM}_1) \geq \mathcal{R}(\mathcal{BM}_2)$.*

Proof. Consider the following method for generating sample paths for $\mathcal{BM}_2$ that relies on i.i.d. sampling of a random variable that is uniformly distributed within $[0, 1]$:

(1) Start $\mathcal{BM}_2$ in its initial state 0.
(2) When in state i, sample a value s uniformly in the interval $[0, 1]$. If the value falls under $p_{loop2} = (1 - x_i(\mathcal{BM}_2))y_i$, take the self loop transition back to i, and repeat step 2. Otherwise proceed.

(3) Sample another s again uniformly in $[0, 1]$ and transition to state j whenever $\sum_{\ell<j} p_{i,\ell} < s \leq \sum_{\ell \leq j} p_{i,\ell}$, i.e., choosing the next state to transition with probability $p_{i,j}$.
(4) Sample s again uniformly in $[0, 1]$ to choose the color of the aforementioned i-to-j transition , with the edge being red when $s < p_{red2} = \frac{x_i(\mathcal{BM}_2)y_i}{(1-y_i)+x_i(\mathcal{BM}_2))y_i}$ (this the probability conditioned on a non-self loop having been taken).
(5) return to step 2

The method for generating sample paths for $\mathcal{BM}_1$ is similar with a few changes:

- For step 4 substitute $x_i(\mathcal{BM}_1)$ in place of $x_i(\mathcal{BM}_2)$. Since $p_{red1} = x_i(\mathcal{BM}_1) \geq x_i(\mathcal{BM}_2)$, this increases the likelihood of a red edge being chosen since it follows that:

$$p_{red1} = \frac{x_i(\mathcal{BM}_1)y_i}{(1-y_i)+x_i(\mathcal{BM}_1)y_i} \geq \frac{x_i(\mathcal{BM}_2)y_i}{(1-y_i)+x_i(\mathcal{BM}_2)y_i}$$

- Step 2 remains the same, using the $x_i(\mathcal{BM}_2)$ bound to determine when to transition to the next step and select the exiting transition. However self-loops are only for samples s for which it and all previous samples s' since entering the state satisfy $s, s' < (1-x_i(\mathcal{BM}_1))y_i = p_{loop1}$. In other words, self-loops are added until some $s \geq (1-x_i(\mathcal{BM}_1))y_i$, and the sample path for $\mathcal{BM}_1$ stays in a "holding pattern", leaving the sample path unmodified, until an s is sampled where $s > (1-x_i(\mathcal{BM}_2))y_i$, at which point the construction can move to step 3.

Note that since the samples s are i.i.d., the "tossing" of samples when building $\mathcal{BM}_1$ does not alter its sample path distribution. Also note that, while we initially pose s as a random value uniformly distributed within $[0, 1]$, since our resulting action only depends upon the interval within which s lies, we can discretize the random variable being sampled such that s selects from a finite set of outcomes, where the size of the set equals the number of sub-intervals, and the likelihood of choosing a particular option is equal to the size of its subinterval in the original formulation.

Finally, we can now use a single sample sequence to simultaneously generate sample paths P_1 and P_2 respectively for both $\mathcal{BM}_1$ and $\mathcal{BM}_2$ that satisfy their respective probability distributions. For instance, to make the determiniation of the color of outgoing edges in $\mathcal{BM}_1$ and $\mathcal{BM}_2$ simultaneously (i.e., step 4), we choose both red with probability p_{red2}, $\mathcal{BM}_1$'s transition red and $\mathcal{BM}_2$'s transition black with probability $p_{red1} - p_{red2}$, and both black with probability $1 - p_{red1}$. The following properties also hold for sample paths P_1 and P_2:

- The sequence of states that P_1 and P_2 progress through, ignoring self-loops, is identical.
- If P_2 marks its j non-self-loop transitioning edge as red, then P_1 does as well.
- After taking their respective kth non-self-loop transitioning edges to enter the some identical state i_k, the number of times that P_2 self-loops before exiting the state is never less than that of P_1. It follows that for all k, $\mathcal{BM}_2$ will never reach its kth non-self loop before $\mathcal{BM}_1$ does.
- This implies that the number of red transitions crossed within an n-transition sample path is never larger in $\mathcal{BM}_2$ than in $\mathcal{BM}_1$, both because fewer non-self loop transitions will be red in $\mathcal{BM}_2$ and because there are more self-loops between non-self-loop transitions.

Define $C_n(P)$ to be the number of non-self-loop transitions occuring within the first n transitions of sample path P. Note that when P_1 and P_2 are generated with the same uniform sampling process, it is always the case that $C_n(P_1) > C_n(P_2)$. We can define $\phi_j(S)$ as the process that generates sample path P_j for $\mathcal{BM}_j$ using sampling sequence S, and we have that $\forall S, C_n(\phi_1(S)) \geq C_n(\phi_2(S))$. Also, $meas(P_j) = \sum_{\{S:\phi_j(s)=P_j\}} meas(S)$, i.e., the likelihood of sample path P_j is equal to the likelihood that the sampling process using samples s generates P_j. unique sample path P_2 for $\mathcal{BM}_2$ such that $meas(P_2) = meas(S)$. There are many S that generate each P_1 in $\mathcal{BM}_1$ (i.e., different-length

"freeze" portions), but $meas(P_1) = \sum_{\phi_1(S)=P_1} meas(S)$. Clearly, because $\forall S, n, C_n(\phi_1(S) \geq C_n(\phi_2(S))$, the result holds. □

Our initial version of the red-black model omits one subtlety of our final implementation of our Markov Model for false negative: the "get-lucky" event. It was omitted to simplify presentation. We address it here. Per state, we introduce a *bias* parameter z_i that is used to bias away from marking a transition red. In effect, the probability of marking a non-self-loop transition red changes from $p_{i,j}x_iy_i$ to $p_{i,j}x_iy_i(1-z_i)$, and hence the probability of marking a non-self-loop transition black from $p_{i,j}(1-y_i)$ to $p_{i,j}((1-y_i)+x_iy_iz_i$. Intuitively, since this bias is identical per state across Markov Models $\mathcal{BM}_1$ and $\mathcal{BM}_2$, it should be no surprise that the main Theorem's results extend to the case where we have some $0 \leq z_i \leq 1$.

Corollary 1. *With an additional bias per state i of $1-z_i$ against red transitions in both $\mathcal{BM}_1$ and $\mathcal{BM}_2$, we still have $\mathcal{R}(\mathcal{BM}_1) \geq \mathcal{R}(\mathcal{BM}_2)$.*

Proof. The conditional probability of choosing red on a transition from i to j, given a non-self-loop transition is being taken changes from $\frac{x_i(\mathcal{BM}_j)y_i}{(1-y_i)+x_i(\mathcal{BM}_j)y_i}$ to $\frac{x_i(\mathcal{BM}_j)y_i}{(1-y_i)(1-z_i)+x_i(\mathcal{BM}_j)y_i}$. The fact that the former is larger for $\mathcal{BM}_1$ implies that so is the latter. Hence, the application of the proof of Theorem 3 still applies. □

Corollary 2. *Our Markov Model that measures false negative rate can be presented as a red-black Markov Model*

Proof. Set $y_i = p_r$. Note that $p_{i,j}$ is non-zero only for $j = i+1$ and for $j = 0$ (for non-retaining) and $j = 1$ for retaining in states i, and $p_{i,i+1} = 1-\rho(i)$. Finally, x_i equals the corresponding first-time sampling rate $\widehat{f_n}$, while z_i is the get-lucky likelihood $\gamma_k(i)$.

To use the theorem in the context of an upper bound, apply Corollary 1 with $\mathcal{BM}_2$ modeling the actual false negative rate (using $x_i = \widehat{f_n}$), while $\mathcal{BM}_1$ implements an upper bound using x_i set to one of the appropriate upper bounding formulae. For the lower bound, use $\mathcal{BM}_1$ to model the actual false negative rate, while $\mathcal{BM}_2$ models the appropriate lower bounding formula.

□

Lemma 1. *An item currently sampled with popularity q_j had popularity $q_{j'}$ in previous iterations with $j' \leq j$, such that q_j can be no higher than in earlier iterations.*

Corollary 3. *Equations* (6)*, 7, 9 further upper-bound as p_r decreases.*

Proof. Each of these equations is derived for state i by iterating over the set of D messages in the popularity distribution, and for each $0 \leq j < D$, considering the case where that message is currently sampled and is not one of the previously i sampled messages. This formula is making two assumptions:

- that the j message was not replaced by a "new" message sampled earlier in the current cycle (such that its sampling is for certain not a false negative).
- that in previous iterations, the likelihood of sampling this message was also q_j.

However, due to the new arrival process, the message sampled from the distribution might be in fact be "new" such that the false negative likelihood associated with this message is 0. And second, in earlier iterations, new arrivals might have shifted this message from some higher popularity to its current popularity.

For all upper bound variants, this first assumption causes us to add positive value when nothing should have been added (further support for an upper bound). The second assumption means that $(1-q_j)$ used in (6) is an overestimate, again furthering to support the upper bound. This second

assumption has no affect on the uniform distribution, but on the bi-uniform distribution, when sampled from the smaller value, we may be underestimating its likelhiood in earlier samples, such that such terms in the computation is off by a factor of p_h/p_l (again, furthering support to the upper bound).

□

B RESELECTING IS UPPER BOUND

To understand the challenge we are addressing here, consider a Markov Model which arrives in state i where the likelihood of self-transitioning depends upon a particular of the configuration: in our problem, it is the specific set of i messages that are recorded. An accurate model must consider all possible such configurations, and then determine the likelihood of entering into state i in a particular configuration and then modeling the self-transitioning process for each such configuration. Instead, we assume that each time the self-transition is taken, the i elements are simply resampled. We show that our presumed process is in fact an upper bound on the more accurate process. When computing our lower bound, note that we always assumed the same set of elements being selected (the most popular), such that this issue does not arise.

We formulate the problem as a series of coin tossing sequences, where the different coins represent different possible configurations (sets of i elements sampled): entering the state in a particular configuration and self-transitioning back to this same configuration is akin to selecting a coin and reflipping this coin repeatedly. In contrast, resampling and choosing a new configuration after each self-transition is akin to re-selecting a coin on each toss.

Consider a set of biased coins $\Omega = \{i\}$ where the ith coin has probability P_i of being selected and flips to heads with probability p_i. WLOG, sort the coins according to the p_i such that $i < j \rightarrow p_i \leq p_j$, where we make no implicit assumption about the relative sizes of the P_i. Consider a series of rounds in which a process ψ is used to select a coin each round and flip it. Let $\rho(\psi)$ represent the rate of heads for process ψ. We focus on two specific processes ψ.

- ψ_r (random): each round, a new coin is sampled and flipped.
- ψ_h (held): after a flip of heads, a new coin is selected, but after a flip of tails, the same coin is used in the subsequent round.

Theorem 4. $\rho(\psi_r) \geq \rho(\psi_h)$

Proof. Let ϕ_i^x represent the steady-state likelihood of coin i being the coin flipped in process x. Then $\rho(\psi_x) = \sum_i \phi_i^x p_i$.

In process ψ_r, we have that $\phi_i^r = P_i$, whereas the likelihoods may be different in ψ_h, such that there are constants M_i where we write $\phi_i^h = M_i P_i$ for constants $M_i > 0$ such that $\sum_i M_i P_i = 1$. We let $\Delta_i = \phi_i^h - \phi_i^r = (M_i - 1)P_i$ represent the difference per i of these steady-state likelihoods, and note that since $\sum_i \phi_i^x = 1$, it must be that $\sum_i \Delta_i = 0$. We can therefore split Ω into two sets, $\Omega^+ = \{i : \Delta_i \geq 0\}$ and $\Omega^- = \{i : \Delta_i < 0\}$ such that $\sum_{i\in\Omega^+} \Delta_i \geq 0, \sum_{i\in\Omega^-} \Delta_i < 0$ and $\sum_{i\in\Omega^+} \Delta_i = -\sum_{i\in\Omega^-} \Delta_i$.

Note that in process ψ_h, once a coin i is selected, the number of rounds until obtaining a heads is purely a function of p_i, and is independent of P_i: in fact, the probability of exceeding j rounds is simply $(1 - p_i)^j$. If we define L_i to be a random variable equal to the number of flips in a row of coin i when it is selected in ψ_h, we have that $E[L_i] = 1/p_i$, and by renewal theory, $\phi_i^h = \frac{P_i E[L_i]}{\sum_j P_j E[L_j]} = \frac{P_i}{p_i \sum_j P_j/p_j}$. So $M_i = \frac{1}{p_i \sum_j P_j/p_j}$ is decreasing with increasing p_i. It follows that there is some index n and constants $0 < p^+ < p^- < 1$ for which $i \leq n \rightarrow p_i \leq p^+$ and $i \in \Omega^+$, and otherwise $i > n \rightarrow p_i > p-$ and $i \in \Omega^-$.

We thus have:

$$\rho(\psi_h) - \rho(\psi_r) = \sum_i p_i(\phi_i^h - \phi_i^r) = \sum_i p_i\Delta_i = \sum_{i\in\Omega^+} p_i\Delta_i + \sum_{i\in\Omega^-} p_i\Delta_i < p^+ \sum_{i\in\Omega^+} \Delta_i + p^- \sum_{i\in\Omega^i} \Delta_i.$$

Since we have that $p^+ < p^-$, $\sum_{i\in\Omega^+} \Delta_i \geq 0, \sum_{i\in\Omega^-} \Delta_i < 0$ and $\sum_{i\in\Omega^+} \Delta_i = -\sum_{i\in\Omega^-} \Delta_i$, we conclude:

$$\rho(\psi_h) - \rho(\psi_r) = p^+ \sum_{i\in\Omega^+} \Delta_i + p^- \sum_{i\in\Omega^i} \Delta_i \leq \sum_{i\in\Omega^+} \Delta_i + \sum_{i\in\Omega^i} \Delta_i = 0.$$

□

C UNIFORM UPPER BOUNDING LEMMA

Define $P_{\mathcal{X}}^K(j)$ to be the probability that when sampling K elements without replacement from distribution $\mathcal{X}$, that element j is one of the K sampled elements.

Lemma 2. *For any distribution $\mathcal{X}$, when sampling K elements without replacement, $\sum_j P_{\mathcal{X}}^K(j) = K$.*

Proof. Let E_j be an indicator random variable that equals 1 when element j is one of the K elements included in the sample (without replacement). Clearly $\sum_j E_j = K$ (exactly K distinct elements are sampled), such that $\sum_j E[E_j] = E[\sum_j E_j] = K$. Because E_j is an indicator, we have $E[E_j] = P_{\mathcal{X}}^K(j)$. □

Some of our proofs will make use of the fact that sampling without replacement is easily emulated by using a sampling with replacement process. Given a distribution $\mathcal{Q}$ (where we sort elements from highest sampling likelihood to smallest), to sample K items without replacement, we sample with replacement, and discard any repeated selections until we select K distinct elements. In the case of $\mathcal{Q}$, we can do each such sampling by directly sampling the elements proportional to their probability. Note that a particular sequence of sampling ℓ elements (with replacement : $(i_1, i_2, \ldots, i_\ell)$ occurs with probability $\prod_{j=1}^{\ell} q_{i_j}$.

An slight variant on this method of sampling that we will use later is to repeatedly choose a value x uniformly within $[0, 1)$, and selecting item j when x falls within $[\rho_{j-1}, \rho_j)$, where $\rho_j = \sum_{n=1}^{j} q_n$, i.e., a region of size q_j.

Lemma 3. *For any K, $P_{\mathcal{Q}}^K(j)$ is a non-increasing sequence of j.*

Proof. We consider two elements in particular, j and $j' < j$ and show that $P_{\mathcal{Q}}^K(j') \geq P_{\mathcal{Q}}^K(j)$ using sample path analysis upon sampling-with-replacement sequences that are used to emulate sampling without replacement.

Consider the set of all distinct sample paths that select K elements without replacement from $\mathcal{Q}$ by using the sampling-with-replacement process above. Any sample path P that contains neither j nor j' contributes to neither probability measure, and any sample path that contains both j and j' contributes equally to both elements' probability measures.

Now consider a sample path P that contains samples of j but none of j'. We map this sample path to an alternative sample path P' that replaces each sample of j with j'. While the former contributes only to the probability measure of j, the latter only contributes to the probability measure of j'. Since we replace an independent sampling of likelihood q_j with one of likelihood $q_{j'\geq q_j}$, the likelihood of path P' is no less than that of P. Since this mapping is one-to-one, clearly the sum of the probabilities over all sample paths that include element j is no larger than the sum over those including element j'.

□

The likelihood of a false negative event in the ith state of the Markov Model implies that the $i+1$st arriving message differs from the preceding i messages that are recorded in the RBF. For a D-element distribution $\boldsymbol{Q}$, the probability of such an event is thus $\sum_{j=1}^{D} q_j(1-P_{\boldsymbol{Q}}^i(j))$, i.e., summing over all possible elements j, that the $i+1$st sample is j and that it was not one of the previous i samples (wihtout replacement). It is slightly easier to work with the negation of this event, the non-false negative: the probability that the $i+1$st arriving element is not a false negative, which is $1-\sum_{j=1}^{D} q_j(1-P_{\boldsymbol{Q}}^i(j)) = \sum_{j=1}^{D} q_j P_{\boldsymbol{Q}}^i(j)$, i.e., the $i+1$st sample was previously sampled.

Theorem 5. *The uniform distribution $\mathcal{U}$ over D elements, each of whose elements has sampling likelihood of $1/D$, has the largest false negative rate in state i (where i elements have already been recorded in the RBF).*

Proof. From Lemma 2, we have that $\sum_j P_{\mathcal{U}}^i(j) = i$, and since all elements are identical in $\mathcal{U}$, it must be that $P_{\mathcal{U}}^i(j) = i/D$. We prove that $\mathcal{U}$ has the lowest non-false negative rate by showing that $\sum_{j=1}^{D} q_j P_{\boldsymbol{Q}}^i(j) - \sum_{j=1}^{D} 1/D(i/D) \geq 0$, which we can rewrite as $\sum_{j=1}^{D}(q_j P_{\boldsymbol{Q}}^i(j) - 1/D(i/D)) > 0$.

We can rewrite the above sum as the sum of S_1 and S_2 where $S_1 = \sum_{j=1}^{D} q_j(P_{\boldsymbol{Q}}^i(j) - i/D)$ and $S_2 = \sum_{j=1}^{D} i/D(q_j - 1/D)$. For S_1, note that by Lemma 2, $\sum_{j=1}^{D} P_{\boldsymbol{Q}}^i(j) = i = \sum_{j=1}^{D} i/D$ and so $\sum_{j=1}^{D}(P_{\boldsymbol{Q}}^i(j) - i/D) = 0$, and since by Lemma 3, we have that the $P_{\boldsymbol{Q}}^i(j)$ are non-increasing, there must be some k for which $j \leq k \leftrightarrow P_{\boldsymbol{Q}}^i(j) \geq i/D$, such that we can decompose S_1 into two positive parts $T_1 - T_2$ where $T_1 = \sum_{j=1}^{k} q_j(P_{\boldsymbol{Q}}^i(j) - i/D)$ and $T_2 = \sum_{j=k+1}^{D} q_j(i/D - q_j P_{\boldsymbol{Q}}^i(j))$. Finally noting that each term in T_1 satisfies $q_j(P_{\boldsymbol{Q}}^i(j) - i/D) \geq q_k(P_{\boldsymbol{Q}}^i(j) - i/D)$ and each term in T_2 satisfies $q_j(i/D - q_j P_{\boldsymbol{Q}}^i(j)) \leq q_{k+1}(i/D - P_{\boldsymbol{Q}}^i(j)$ we have that

$$
\begin{aligned}
S_1 &= \sum_{j=1}^{k} q_j(P_{\boldsymbol{Q}}^i(j) - i/D) - \sum_{j=k+1}^{D} q_j(i/D - q_j P_{\boldsymbol{Q}}^i(j)) && (29)\\
&\geq \sum_{j=1}^{k} q_k(P_{\boldsymbol{Q}}^i(j) - i/D) - \sum_{j=k+1}^{D} q_{k+1}(i/D - q_j P_{\boldsymbol{Q}}^i(j)) && (30)\\
&= q_k \sum_{j=1}^{k}(P_{\boldsymbol{Q}}^i(j) - i/D) - q_{k+1} \sum_{j=k+1}^{D}(i/D - q_j P_{\boldsymbol{Q}}^i(j)) && (31)\\
&\geq q_{k+1} \sum_{j=1}^{D}(P_{\boldsymbol{Q}}^i(j) - i/D) && (32)\\
&= 0 && (33)
\end{aligned}
$$

We proceed with showing $S_2 \geq 0$ similarly by using the fact that distributions always sum to 1 (such that $\sum_{j=1}^{D}(q_j - 1/D) = 0$ whose terms are non-increasing) to obtain our result.

□

D BIUNIFORM UPPER BOUNDING LEMMA

Definition 1. *Consider two d-length sequences $A = a_1, a_2, \ldots, a_d$ and $B = b_1, b_2 \ldots, b_d$ where $\sum_{i=1}^{d} a_d = \sum_{i=1}^{d} b_d$. We say sequence A* **dominates** *sequence B with crossover point j if $a_i \geq b_i \leftrightarrow i \leq j$, i.e., The sequence A starts off larger than its corresponding element in B until crossing index j, and from then on is smaller. We say that A* **semi-dominates** *B if a crossover point j exists such that $i \leq j \rightarrow a_i \geq b_i$, i.e., some $a_i > b_i$ are possible even for $i > j$.*

Note this definition of a dominating sequence extends easily to a pair of d-element distributions, where for each distribution, the sum over all d elements must be one.

Given two distributions, Q and $\mathcal{X}$ where Q dominates $\mathcal{X}$, we will apply the above sampling-with-replacement approach to Q, but will alter the sampling space of $\mathcal{X}$ somewhat to align it with Q as follows:

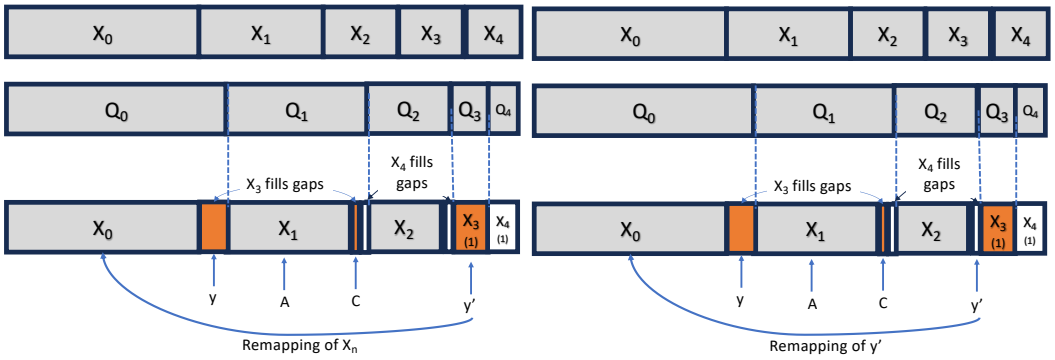


Fig. 9. Sample $d = 5$ element distributions with Q dominating $\mathcal{X}$ with crossover point 2 ($q_i \geq \chi_i$ for $i \leq 2$). $\mathcal{X}$ is reconfigured on the bottom for purposes of computing sample path, such that the χ_i start at the same point as q_i for $i \leq 2$, and χ_3 and χ_4 must be "chopped up" and distributed into the gaps. On the left, we remap the non-gap sample y' to the bump, and on the right we remap one bump sample to another bump.

For each item j where $q_j \geq \chi_j$, we select item j from $\mathcal{X}$ when x falls in the region $[\rho_{j-1}, \rho_{j-1}+\chi_j)$. These intervals in $\mathcal{X}$ are "spaced apart" from one another, leaving gaps, but always fitting inside the region allocated to q_j. For the remaining j where $q_j < \chi_j$, we assign the same region $[\rho_{j-1}, \rho_j)$ to cover part of the χ_j that are too large to fit entirely within the interval. The remaining parts of the $\chi_j > q_j$ are covered by the unused "gaps" that exist where intervals are spaced apart. This is depicted in Fig. 9. Note that even with this "chopped up" layout, the likelihood of sampling any element χ_j remains equal to its probability mass, such that it can be used in the sampling-with-replacement process to identify K items sampled without replacement with accurate likelihood.

Define $Q(x)$ and $\mathcal{X}(x)$ to equal the elements that a uniformly chosen x maps to under the respective distributions. Then a sequence $X = (x_1, x_2, x_3, \ldots, x_\ell)$ whose samples are chosen uniformly at random from the unit interval maps to respective sequences $S_1 = (Q(x_1), Q(x_2), \ldots, Q(x_\ell))$ and $S_2 = (\mathcal{X}(x_1), \mathcal{X}(x_2), \ldots, \mathcal{X}(x_\ell))$, which are respectively sequences of elements (with possible repetition) in Q and $\mathcal{X}$. With large enough ℓ, both sequences will contain at least K distinct samples under these distributions such that the first K distinct samples under each distribution are the samples chosen without replacement.

Let us state an obvious result that we will utilize:

Lemma 4. *Consider a set of K distinct elements selected from distribution $\mathcal{D}$ by the sampling-with replacement above described above with sequence $(x_1, x_2, \ldots, x_\ell)$. Then a reordering of the sequence will still sample the same set of K distinct elements.*

Proof. The set of elements sampled does not change by reordering the sampling points, such that every of the K elements is still sampled and no element that was previously not sampled can now be sampled. □

Let us now distinguish between two types of sampling points x:

- gap-fillers: these are points that sample within the "gaps", i.e., they will select an element in the dominating $\mathcal{Q}$ whose position in the distribution lies at or before the dominating point j, and an element in the dominated $\mathcal{X}$ that lies after j.
- non-gap: these are points that sample outside the gaps, such that the sample for each of $\mathcal{Q}$ and $\mathcal{X}$ whose position within the respective distributions is identical.

In Fig. 9, sample points x are shown at the bottom, with arrows pointing to their sample location. A is a non-gap, y and C are gap-fillers, and y' is non-gap in the left figure, and is a gap-filler in the right figure.

Note that there is a bijective mapping for non-gap samples, i.e., a non-gap sample within $\mathcal{Q}$ uniquely determines the corresponding non-gap sample within $\mathcal{X}$, and vice versa.

We define $P^K_{\mathcal{D}}(i)$ to be the probability that a sampling of K elements from distribution $\mathcal{D}$ without replacement includes element i.

Lemma 5. *Let $\mathcal{Q} = \{q_1, \ldots q_d\}$ and $\mathcal{X} = \{\chi_1, \ldots, \chi_d\}$ be two distributions where $\mathcal{Q}$ dominates $\mathcal{X}$ with crossover point j'. If $K < d$ items are sampled without replacement from each distribution, and let $P_{\mathcal{D}}(i)$ be the probability that item i is included in the K elements sampled from distribution $\mathcal{D} \in \{\mathcal{Q}, \mathcal{X}\}$. Then $i \le j' \rightarrow P^K_{\mathcal{Q}}(i) \ge P^K_{\mathcal{X}}(i)$.*

This is an intuitive result that the probability of sampling a dominating element is larger than the probability of sampling its dominated counterpart. While intuitive, it is hard to prove when sampling without replacement. In the context of Fig. 9, the order in which y, A, C, y' are sampled does not affect the set of distinct items chosen.

Proof. Consider the two sequences
$S_1 = (\mathcal{Q}(x_1), \mathcal{Q}(x_2), \ldots, \mathcal{Q}(x_\ell))$ and $S_2 = (\mathcal{X}(x_1), \mathcal{X}(x_2), \ldots, \mathcal{X}(x_\ell))$ of elements generated from an i.i.d. uniform sampling sequence $(x_1, x_2, \ldots, x_\ell)$. The measure of the set of all sequences $(x_1, x_2, \cdots, x_\ell) \in X$ that map specifically to S_1 and S_2 respectively under distributions $\mathcal{Q}$ and $\mathcal{X}$ respectively equal $\prod_{j=1}^{\ell} A_j$ and $\prod_{j=1}^{\ell} B_j$ where A_j is the measure of all x that satisfy $\mathcal{Q}(x) = \mathcal{Q}(x_j)$ and B_j are all the x that satisfy $\mathcal{X}(x) = \mathcal{X}(x_j)$.

We prove the result by contradiction and assume that $P_{\mathcal{Q}}(i) < P_{\mathcal{X}}(i)$ for some $i \le j'$. This means that there must have been a sequence of sampling-with-replacement x's $(x_1, x_2, \ldots x_\ell)$ that selects K distinct elements in $\mathcal{X}$, where i is one of the first K distinct elements selected under $\mathcal{X}$, but is not one of the first K distinct elements selected under distribution $\mathcal{Q}$. This means there is some $m \le \ell$ such that x_m is a sample where $\mathcal{X}(x_m) = i$. However, since $i \le j'$, x_m must also satisfy $\mathcal{Q}(x_m) = i$. In the example Fig. 9, x_m falls within the region mapping to the $i = 1$st element in $\mathcal{X}$, such that it must also fall within the region covering the ith element in $\mathcal{Q}$.

Since the ith element is clearly sampled within $\mathcal{Q}$ within the first ℓ samples, the only way for i to not appear as one of the K sampled elements under $\mathcal{Q}$ is if all of its K distinct elements were chosen prior to this mth sampling, i.e., some x_j mapped to the Kth distinct element under $\mathcal{Q}$ with $j < m$. In order for $\mathcal{Q}$ to complete its sampling of K elements prior to the mth selection of x, $\mathcal{Q}$ must have "gotten ahead" in the count of distinct elements.

We now consider a reordering $(y_1, y_2, \cdots, y_{m-1})$ of the sample points $(x_1, x_2, \cdots x_{m-1})$, where the reordering places all non-gap samples ahead of gap-fillers. By Lemma 4, the sets of elements sampled will remain the same. During the sampling using this reordered sequence, the bijective nature of the mapping of non-gap samples ensures that during this initial part of the sampling sequence where we are only sampling non-gap samples, the number of distinct elements sampled in $\mathcal{X}$ matches that of $\mathcal{Q}$. Only when we get to the gap-filler sample points can there be a difference.

How can gap-fillers cause $\mathcal{Q}$ to "get ahead"? The uniform sample must sample an element $f \in \mathcal{X}$ has has been previously sampled, while sampling an element $e \in \mathcal{Q}$ that has not been previously

sampled. Hence, it must be a gap-filler that falls into a gap associated with $f > j'$ where the corresponding $e \leq j'$ has not yet been sampled. We call any such uniform sample a *bump*. In Fig. 9, y is a bump in both versions, C is not a bump (because of A), and y' is a bump in the righthand figure. Note bump samples are the only samples that can increase the count of distinct elements in Q while leaving the number of distinct elements in $\mathcal{X}$ unchanged.

To see the impact of bump samples on a particular element $f > j'$, we again reorder the elements such that bump samples on f come before non-bump samples on f. For the case where the non-gap f was previously sampled (adding distinct element f for both Q and $\mathcal{X}$, then each bump sample adds a distinct element to Q while making no change to $\mathcal{X}$ (since f has already been added), and the subsequent non-bump samples do not add elements to either Q or $\mathcal{X}$. Hence, the number of bumps equals the number of bump samples. For the case where the non-gap f was not previously sampled, the first bump adds an element to both Q and since it is the first sample to add f to $\mathcal{X}$, such that the number of bumps is one less than the number of bump samples. Hence Q "gets ahead" by bumps to f only when there is more than one bump to f, or when there is the non-gap f sample and at least one bump.

For the case where f does cause Q to get ahead, we modify a single sample $y \in (y_1, \ldots, y_{m-1})$ as follows: we select y to be a bump. Let e^* be the distinct element in Q that it samples. We then select another $y' \in (y_1, \ldots, y_{m-1})$ that maps either to another bump or else maps the non-gap sample that selected f (one of which must exist for Q to have gotten ahead). We then replace y' with a sample that maps instead to the non-gap e^*. If there are several samples of y that map to the same gap region (or non-gap f) as y', we modify them all similarly.

Note the change in y' adds e^* to $\mathcal{X}$: it was not added previously because no non-gap sample had added e^*. Additionally, Q either no longer samples f (in the case that y' was the non-gap sample to f), or it no longer samples the e that was covered by the bump that was sampled by y'. Hence, we have increased the number of distinct samples in $\mathcal{X}$ by 1, and decreased the number of distinct samples in Q by 1. This is depicted in both version displayed in Fig 9. On the left, the move of y' removes Q_3 from Q and adds X_0 to $\mathcal{X}$. On the right, the move of y' removes Q_2 from Q and again adds X_0 to $\mathcal{X}$. Note that in both cases, sample y had already added Q_0 to Q

Furthermore, the measure of the set of possible y' that could map to f is bounded by the size of f in $\mathcal{X}$, whereas the measure of the set of possible replaced y' that could map to non-gap e^* is the size of e^* in $\mathcal{X}$. Since $e^* \leq j' < f$, we have that the measure of e^* is larger, and we our mapping maps a sample path with a smaller probability to one with larger probability.

We can repeat the above process for all f that might cause Q to get ahead until we reach a sequence where Q falls behind in the number of distinct elements it samples by the $m-1$st sample. By again applying Lemma 4, to return to the original ordering (with the appropriate elements replaced), we have demonstrated that for each sample path that might permit an element $i \leq j'$ to be added to $\mathcal{X}$ but not Q, there is a one-to-one mapping to a sample path with larger measure that permits element $i \leq j'$ to be added to Q, but not $\mathcal{X}$.

□

Corollary 4. *If Q dominates $\mathcal{X}$ with crossover point j', then for all K, $P_Q^K(i)$ semi-dominates $P_{\mathcal{X}}^K(i)$ with the same crossover point $j'' \geq j'$.*

Theorem 6. *Let $Q = \{q_1, \ldots q_d\}$ and $\mathcal{X} = \{\chi_1, \ldots, \chi_d\}$ be two distributions where Q dominates $\mathcal{X}$ to crossover point j', and $\chi_j = \chi_{j+1}$ for all $j > j'$ (i.e., $\mathcal{X}$ "flattens"). Then the false negative rate of Q is lower than that of $\mathcal{X}$*

Proof. When distribution Q dominates $\mathcal{X}$, Corollary 4 gives us that $P_Q^K(i)$ semi-dominates $P_{\mathcal{X}}^K$ with a crossover point at least as large $j'' \geq j'$ for all K. Note that for $j > j''$, if $Q_j - \mathcal{X}_j \leq 0$, then

because Q is non-increasing in j, and $\chi_j = \chi_{j+1}$ for all $j \geq j'$, it must be that $Q_{j+1} - \mathcal{X}_{j+1} \leq 0$, i.e., once this difference is smaller than 0, it stays smaller than 0. Hence, following the line of proof used for Theorem 5, splitting into a positive weighted T_1 and negative weighted T_2 at crossover point j'', we show that the false negative rate over distribution $\mathcal{X}$ is larger than that of Q for all K. □

Corollary 5. *Let $\mathcal{X}$ be a bi-uniform distribution where Q dominates $\mathcal{X}$, and the index s where $\mathcal{X}$ switches from higher popularity p_h to lower popularity p_l occurs at a point where $p_h \leq q_s$. Then the false negative rate of $\mathcal{X}$ is larger than that of Q.*

Proof. Since $p_h \leq q_s$, it must be that $p_h \leq q_j$ for all $j \leq s$. Then at some larger $j'' > s$, there is a first occurrence where $\chi_{j''} = p_l > q_{j''}$ perhaps even $j'' = s + 1$. However, wherever this j'' occurs, distributions Q and $\mathcal{X}$ satisfy the requirements of Theorem 6 (i.e., Q dominates $\mathcal{X}$ at least to point s after which $\mathcal{X}$ is assured to be "flat". □

E RENEWAL RESULTS

The previous formula assumed independence between η_j and $\mathcal{B}_{\ell,j}$ when they appear in the numerator of (21) when in fact they are clearly not (i.e., $\mathcal{B}_{\ell,j}$ is defined in terms of η_j.

One approach would be use a revised $\mathcal{B}_{\ell,j}$ that conditions on element i not having been one of the messages received during the current cycle prior to the j iteration. Define $S_{-i,j}$ to be an indicator that is 1 when element i is not sampled in the first j arrivals. Then what we really need in place of (21) is :

$$P(C_j = 1) = \sum_{i=1}^{D} (1 - q_i)^j q_i \sum_{\ell=0}^{\sigma+k} P(\mathcal{B}_{\ell,j-1} = 1 | S_{-i,j-1} = 1)(1 - \tau_k(\ell, \ell)) \tag{34}$$

i.e., we could define $P(\eta_j^{(-i)} = 1)$ as η_j conditioned on element i not having been received, and $P(\mathcal{B}^{(-i)}) = P(\mathcal{B}_{\ell,j-1} = 1 | S_{-i,j-1} = 1)$ distinctly for each i as:

$$P(\eta_j^{(-i)} = 1) = \sum_{m=1}^{D-1} (1 - q'_m)^{j-1} q'_m \tag{35}$$

$$P(\mathcal{B}_{\ell,j}^{(-i)} = 1) = \begin{cases} (1 - P(\eta_j^{(-i)} = 1) P(\mathcal{B}_{\ell,j-1}^{(-i)} = 1) + \\ P(\eta_j^{(-i)} = 1) \sum_{m=0}^{k} P(\mathcal{B}_{\ell-m,j-1}^{(-i)} = 1) \tau_k(\ell - m, \ell) & \ell \leq \sigma \\ 0 & l > \sigma \end{cases} \tag{36}$$

where $\{q'_m\}$ are the (re-normalized) elements of Q with element i removed from the distribution (i.e., allowing sampling conditioned not choosing i). And then

$$P(C_j = 1) = \sum_{i=1}^{D} (1 - q_i)^j q_i \sum_{\ell=0}^{\sigma+k} P(\mathcal{B}_{\ell,j-1}^{(-i)} = 1)(1 - \tau_k(\ell, \ell)) \tag{37}$$

However, this approach substantially increases the complexity of computing $P(C_j = 1)$, which we would rather avoid.

Theorem 7. $P(\mathcal{B}_{\ell+k,j}) \geq P(\mathcal{B}_{\ell,j}^{(-i)})$.

Proof. Consider the following approach to sampling j items from Q, conditioned on not sampling element i: sample from Q, and whenever sample i is chosen, simply discard it. Clearly, the resulting sample path has no elements i and each element was sampled proportionally to its value in Q. Consider a specific sample path P that generated the j-element sample without i: P contains

$j' \geq j$ samples where the number of times element i was sampled is $j' - j$. Let $N(P)$ be the unique number of elements in the j'-length sample excluding element i, i.e., the number of distinct elements in the length j sampling without i. Alternatively, we can measure $N'(P)$ as the number of distinct elements within the first j samples *including* i, i.e., $N'(P)$ is the number of distinct samples from a sampling where we include i. Since both $N(P)$ and $N'(P)$.

We claim that $N'(P) \leq N(P) + 1$. This is because $N'(P)$ contains $j' - j$ instances of element i (counting as a single distinct element), whereas $N(P)$ has a final set of $j' - j$ samples of elements other than i that are not included in the count of $N'(P)$. $N(P) - N'(P)$ is smallest when these final $j' - j$ elements are re-samples of earlier elements. In this case, $N'(P)$ would have 1 additional element (i) not counted by $N(P)$. $N(P)$ could of course be much larger if some of these final $j' - j$ elements were not previously sampled.

Finally, due to the pseudo-random nature of the hash functions, the distribution on number of bits set in the RBF is a function of the number of distinct elements recorded, i.e., it does not depend upon which are recorded. For our sample path P, if the $N(P)$ distinct messages do not exceed the σ bound, then adding one additional message cannot exceed the $\sigma + k$ bound, since the additional message can set at most k additional bits. It follows that for each sample path P, whenever the $N(P)$ distinct messages do not exceed the σ bound, then the $N'(P) \leq N(P) + 1$ distinct messages cannot exceed the $\sigma + k$ bound.

By observing that the probability of not exceeding the σ bound involves summing over all the measure of all sample paths P times the likelihood that the $N(P)$ recorded elements don't cross the σ theshhold. Since this likelihood is no more than the likelihood of $N'(P)$ elements crossing the $\sigma + k$ threshold, and since the probability of exceeding the $\sigma + k$ bound with i included equals the measure of these paths P times the likelihood that the $N'(P)$ elements exceed the $sigma + k$ bound, the result holds. □

E.1 Truncated Renewal Process is an upper bound

Define $\alpha(i, m) = \sum_{j=i}^{m} E[C_j]$ and $\beta(i, m) = \sum_{j=i}^{m} E[\mathcal{A}_j]$, and $\mathcal{R}(i, j) = \alpha(i, j)/\beta(i, j)$. Then the false negative rate can be expressed as $\mathcal{R}(1, \infty)$.

Note that since $P(\eta_j = 1)(1 - \gamma_k(j))$ is non-increasing in j and converges to 0, we have that

$$\begin{aligned}
\alpha(1, m) &= \sum_{j=1}^{m} E[C_j] = \sum_{\ell=0}^{\sigma+k} P(\mathcal{B}_{\ell,j-1} = 1)P(\eta_j = 1)(1 - \gamma_k(j) && (38)\\
&\geq P(\eta_m = 1)(1 - \gamma_k(m))\beta(1, m) && (39)\\
\alpha(m + 1, \infty) &= \sum_{j=m+1}^{\infty} E[C_j] = \sum_{\ell=0}^{\sigma+k} P(\mathcal{B}_{\ell,j-1} = 1)P(\eta_j = 1)(1 - \gamma_k(j) && (40)\\
&\leq P(\eta_m = 1)(1 - \gamma_k(m))\beta(m + 1, \infty) && (41)
\end{aligned}$$

such that $\mathcal{R}(1, m) \geq P(\eta_m = 1)(1 - \gamma_k(m) \geq \mathcal{R}(m + 1, \infty)$.

For any m, $\mathcal{R}(1, \infty) \leq \mathcal{R}(1, m)$. The proof is to rewrite as:

$$\frac{\alpha(1, m) + \alpha(m + 1, \infty)}{\beta(1, m) + \beta(m + 1, \infty)} \leq \frac{\alpha(1, m)}{\beta(1, m)}$$

and look at the conditions for this to hold true. Multiplying through and reducing yields:

$$\alpha(m + 1, \infty)\beta(1, m) \leq \alpha(1, m)\beta(m + 1, \infty)$$

Recalling that $\alpha(1, m) \geq P(\eta_m = 1)(1 - \gamma_k(m))\beta(1, m)$ and that $\alpha(m + 1, \infty) \leq P(\eta_m = 1)(1 - \gamma_k(m))\beta(m + 1, \infty)$, we get

$$\alpha(m+1,\infty)\beta(1,m) \leq P(\eta_m = 1)(1-\gamma_k(m))\beta(m+1,\infty)\beta(1,m) \leq \alpha(1,m)\beta(m+1,\infty),$$

so the result always holds.

F OPTIMAL NUMBER OF HASH FUNCTIONS

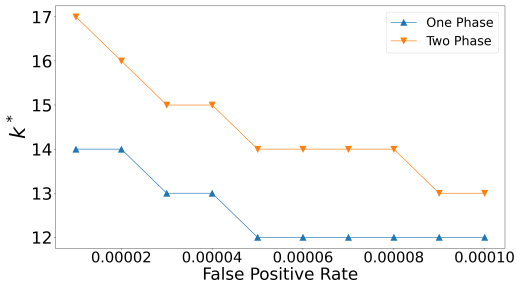


Fig. 10. k parameter to minimize False Negative Rate for a given False Positive tolerance.

We have already seen that the choice of the number of hash functions k clearly has an impact on RBF performance, with respect to both False Positive and False Negative rates, and for each fixed set of parameters there is an "optimal" k. Historically, this parameter has been chosen through either approximations based on the "worst case" False Positive rate, or by a kind of "trial and error" approach [27]. We investigate this relationship by plotting best values of k for one and two-phase RBFs in Fig. 10. The total RBF memory is $M = 10,000$ (5,000 each for the two-phase RBF).

We see that the optimal value of k follows a monotonically-decreasing trend with the False Positive rate; the relationship is also somewhat nonlinear, emphasizing the need for an analytical method to find it exactly.

G ADDITIONAL MARKOV MODEL DETAILS

G.1 Markov Model 2-phase

The active filter and frozen filter can be evaluated independently. This assumes that bits in the active filter are filled even when a message is marked as repeat by the frozen filter, and that hash functions "rotate" each time we change a filter (such that the same message sets different bits in active and frozen). Note this latter property only affects the independence of the "get lucky" process, so its effect is likely negligible.

The steady state of the system is simply $\sum_{i,j} \Pi_i F_j$, and outgoing transitions are as pictured below:

G.2 Disabling

$$h_i(j+1) = \frac{p_r}{(1-p_r)+p_r\widehat{f_n}[j]}[h_i(j)+(1-h_i(j))q_i] + \tag{42}$$

$$\frac{1-p_r}{(1-p_r)+p_r\widehat{f_n}[j]}\left[d_i + h_{i-1}(j)\sum_{i'<i} d_{i'} + h_i(j)\sum_{i'>i} d_{i'}\right] \tag{43}$$

Unlike the renewal case, we update h_i only when transitioning from state j to state $j+1$. Note that this update can occur during a repeat transmission or during a new arrival transition. Conditioned on not taking the self-loop, these relative probabilities are respectively $\frac{p_r}{(1-p_r)+p_r\widehat{f_n}[j]}$

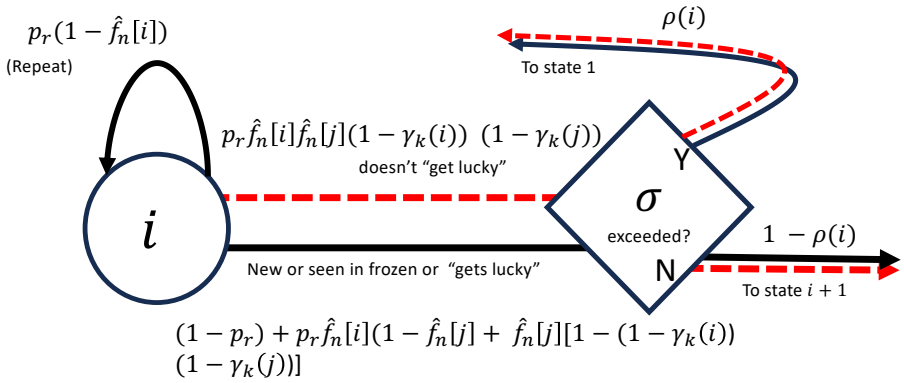


Fig. 11. Markov model showing state transitions of our model. Red transitions are dashed and indicate a false negative event.

and $\frac{1-p_r}{(1-p_r)+p_r\hat{f}_n[j]}$, i.e., we must normalize by the likelihood that the self-loop transition was not finally not taken, i.e., thin by 1 minus the probability of the self-loop being taken.

Received January 2023; revised April 2024; accepted April 2024