

Retry Budgets and Collision Bounds

What exponential backoff and jitter can prove under explicit assumptions

Kylian de Groot

5 September 2026

ABSTRACT

A retry policy should state which resource it bounds. This note separates a deterministic bound on attempts and elapsed time from a probabilistic model of success. It derives finite geometric-sum formulas, shows how retries multiply across service layers, and computes the expected number of pairwise collisions when clients choose independent delay slots. These bounds explain useful properties of backoff and jitter without claiming that either mechanism guarantees recovery, prevents duplicate effects, or measures the behaviour of a deployed service.

Expository note: elementary results with explicit proofs. No claim of mathematical novelty or measured product performance.

1. State the budget before the probability

Consider a logical operation that performs at most K sequential attempts, where $K \geq 1$ includes the first attempt. Let $\tau \geq 0$; each attempt consumes at most τ units of elapsed time. After the j -th failed attempt, before attempt $j + 1$, it waits at most $b \cdot 2^{j-1}$, where $b \geq 0$ and j ranges from 1 to $K - 1$. There is no wait after the last attempt.

The attempt bound is enforced, not inferred from a likely failure rate. The duration assumption includes every part of an attempt counted by the budget. If queueing or a blocked call can exceed τ , the following elapsed-time bound does not cover it.

Proposition 1. Completion or abandonment of the retry sequence occurs within $K\tau + b(2^{K-1} - 1)$, under the stated sequential-duration assumptions.

Proof. There are at most K attempts, contributing at most $K\tau$. The $K - 1$ intervening waits contribute at most b times $1 + 2 + \dots + 2^{K-2}$. For $K = 1$ the sum is empty and equals zero. Otherwise, subtracting the sum from twice itself gives $2^{K-1} - 1$. Adding attempt and wait bounds proves the result.

$$T \leq K\tau + b \sum_{j=0}^{K-2} 2^j = K\tau + b(2^{K-1} - 1).$$

With four attempts, $\tau = 500$ milliseconds and $b = 100$ milliseconds, the waits are bounded by 100, 200 and 400 milliseconds. The complete bound is $4 \times 500 + 100 + 200 + 400 = 2,700$ milliseconds. A policy with four retries would instead have five attempts; the two descriptions must not be confused.

If each wait is capped at $C \geq 0$, the exact sum of wait bounds is sum over $j = 0, \dots, K - 2$ of $\min(C, b2^j)$, which is at most $(K - 1)C$. Randomly choosing a wait inside each allowed interval cannot exceed the same deterministic bound. Randomness does not remove the need to enforce K or τ .

2. A finite probability model for successful completion

Now add a separate idealised assumption: each attempt independently fails with the same probability p , where $0 \leq p \leq 1$. Stop at the first success or after K failures. Let A be the number of attempts actually made. This is a mathematical model, not an estimate of a real API's failure probability.

Proposition 2. Success probability is $1 - p^K$. For $p \neq 1$, the expected number of attempts is $(1 - p^K)/(1 - p)$; for $p = 1$, it is K .

Proof. All K attempts fail with probability p^K by independence. This is exactly the event of no success. For the expectation, write A as the sum of the K indicators of the events $A \geq j$. Attempt j occurs exactly when the preceding $j - 1$ attempts failed, so its probability is p^{j-1} . Linearity of expectation gives $E[A] = \sum_{j=0}^{K-1} p^j$. Multiplication by $1 - p$ telescopes to $1 - p^K$ when $p \neq 1$. At $p = 1$ every term equals one.

$$P(\text{success}) = 1 - p^K, E[A] = \sum_{j=0}^{K-1} p^j.$$

For $p = 1/5$ and $K = 4$, the success probability is $624/625$ and the expected attempt count is $1 + 1/5 + 1/25 + 1/125 = 156/125 = 1.248$. These exact fractions describe the hypothetical independent-attempt model. A shared outage can make failures strongly dependent and invalidate the product p^K . The deterministic attempt cap remains valid.

3. Retry budgets multiply across layers

Proposition 3. Suppose a call traverses d layers. An invocation at layer i makes at most K_i attempts at the next layer. Then one initial invocation can generate at most the product $K_1 K_2 \dots K_d$ leaf attempts. This bound is attained by an execution that exhausts every budget.

Proof. One layer produces at most K_1 next-layer invocations. Each can produce at most K_2 further invocations, giving at most $K_1 K_2$. Repeating this multiplication proves the bound by induction on depth. If every invocation fails in the manner that triggers all permitted retries, each branching factor equals its cap and the product is reached.

Five retrying layers with three attempts each admit $3^5 = 243$ leaf attempts. This is a bound for the stated call-tree structure, not a measurement of any particular service. Sharing a logical-operation budget across layers is one way to change the structure being bounded; setting a local limit independently at every layer is not equivalent.

4. What uniformly random delay slots accomplish

Let n and m be positive integers. Suppose n clients each choose one of m discrete retry slots, independently and uniformly. A collision pair is an unordered pair of clients that choose the same slot. Let C_{pair} count those pairs. A slot containing three clients contributes three pairs, not one.

Proposition 4. The expected number of collision pairs is $n(n - 1)/(2m)$. The probability of at least one collision is at most $\min(1, n(n - 1)/(2m))$.

Proof. For each pair $i < j$ define I_{ij} to be one when their slots coincide and zero otherwise. Independence and uniformity give $P(I_{ij} = 1) = m \times (1/m)^2 = 1/m$. There are $n(n - 1)/2$ pairs. Summing their expectations gives the formula for C_{pair} ; the pair indicators need not themselves be independent. Since the indicator of $C_{\text{pair}} \geq 1$ is no larger than the nonnegative integer C_{pair} , taking expectations gives the probability bound. Every probability is also at most one.

$$E[C_{\text{pair}}] = n(n-1)/(2m).$$

For ten clients and one hundred slots, the expected pair count is $45/100 = 0.45$. The probability bound is also 0.45, but it is not an equality. When $n \leq m$, the exact probability of no collision is the product $(1 - j/m)$ for $j = 0, \dots, n-1$: the first choice is free, and each later choice must avoid the slots already used. If $n > m$, a collision is certain by the pigeonhole principle.

5. The boundaries of the model

The collision result is for independent uniform choices in one slot-selection round. Shared random seeds, nonuniform choices and correlated schedules can change it. It counts coincident slots, not server overload: request durations, service capacity and the width of a slot require another model.

Backoff bounds waiting only when its parameters and stopping rule are enforced. Jitter can spread attempts under a specified distribution. Neither proves eventual recovery, nor prevents a committed financial effect from being repeated after a lost response. That latter question requires the state invariant developed in the companion note on idempotency.

The equations are elementary finite sums and indicator arguments. Their role is to make assumptions inspectable before a reliability claim is attached to a product. No synthetic example above is presented as observed uptime, latency or production success rate.

References

Marc Brooker. [Timeouts, retries, and backoff with jitter](#). Amazon Builders' Library.

Kylian de Groot. [Idempotency Is an Invariant](#). Technical note, 2026.

Companion verification. [Download the finite checks \(Python 3\)](#). Exact integer and rational checks supplement the proofs; they do not establish universal claims beyond the checked domains.