Exponential backoff

From Wikipedia, the free encyclopedia

Exponential backoff is an algorithm that uses feedback to multiplicatively decrease the rate of some process, in order to gradually find an acceptable rate.

Binary exponential backoff / truncated exponential backoff   [edit]

In a variety of computer networks, **binary exponential backoff** or **truncated binary exponential backoff** refers to an algorithm used to space out repeated retransmissions of the same block of data, often as part of network congestion avoidance.

Examples are the retransmission of frames in carrier sense multiple access with collision avoidance (CSMA/CA) and carrier sense multiple access with collision detection (CSMA/CD) networks, where this algorithm is part of the channel access method used to send data on these networks. In Ethernet networks, the algorithm is commonly used to schedule retransmissions after collisions. The retransmission is delayed by an amount of time derived from the slot time and the number of attempts to retransmit.

After c collisions, a random number of slot times between 0 and 2^c - 1 is chosen. For the first collision, each sender will wait 0 or 1 slot times. After the second collision, the senders will wait anywhere from 0 to 3 slot times (inclusive). After the third collision, the senders will wait anywhere from 0 to 7 slot times (inclusive), and so forth. As the number of retransmission attempts increases, the number of possibilities for delay increases exponentially.

The 'truncated' simply means that after a certain number of increases, the exponentiation stops; i.e. the retransmission timeout reaches a ceiling, and thereafter does not increase any further. For example, if the ceiling is set at i = 10 (as it is in the IEEE 802.3 CSMA/CD standard[1]), then the maximum delay is 1023 slot times.

Because these delays cause other stations that are sending to collide as well, there is a possibility that, on a busy network, hundreds of people may be caught in a single collision set. Because of this possibility, the process is aborted after 16 attempts at transmission.[2]

An example of an exponential back-off algorithm   [edit]

This example is from the Ethernet protocol,[3] where a sending host is able to know when a collision has occurred (that is, another host has tried to transmit), when it is sending a frame. If both hosts attempted to re-transmit as soon as a collision occurred, there would be yet another collision — and the pattern would continue forever. The hosts must choose a random value within an acceptable range to ensure that this situation doesn't happen. An exponential back-off algorithm is therefore used. The figure 51.2μs is used as an example here because it is the slot time for a 10 Mbit/s Ethernet line (see Slot time). However, 51.2μs could be replaced by any positive value, in practice.

1. When a collision first occurs, send a "Jamming signal" to prevent further data being sent.
2. Resend a frame after either 0 seconds or 51.2μs, chosen at random.
3. If that fails, resend the frame after either 0s, 51.2μs, 102.4μs, or 153.6μs.
4. If that still doesn't work, resend the frame after k · 51.2μs, where k is a random integer between 0 and $2^3 - 1$.
5. In general, after the cth failed attempt, resend the frame after k · 51.2μs, where k is a random integer between 0 and $2^c - 1$.

Expected backoff [edit]

Given a uniform distribution of backoff times, the expected backoff time is the mean of the possibilities. That is, after c collisions, the number of backoff slots is in $[0, 1, ..., N]$ where $N = 2^c - 1$ and the expected backoff time (in slots) is

$$\frac{1}{N+1}\sum_{i=0}^{N} i$$

For example, the expected backoff time for the third ($c = 3$) collision, one could first calculate the maximum backoff time, N:

$$N = 2^c - 1$$
$$N = 2^3 - 1 = 8 - 1$$
$$N = 7$$

... and then calculate the mean of the backoff time possibilities:

$$\mathrm{E}(c) = \frac{1}{N+1}\sum_{i=0}^{N} i$$
$$\mathrm{E}(3) = \frac{1}{7+1}\sum_{i=0}^{7} i = \frac{1}{8}(0 + 1 + 2 + 3 + 4 + 5 + 6 + 7) = \frac{28}{8}$$
$$\mathrm{E}(3) = 3.5$$

... obtaining 3.5 as the expected number of back off slots after 3 collisions.

The above derivation is largely unnecessary when you realize that the mean of consecutive integers is equal to the mean of the largest and smallest numbers in the set. That is, the mean of a set **A** of consecutive integers $a_0, a_1, a_2, ... a_m$ is simply the mean of a_0 and a_m or $(a_0 + a_m) / 2$.

When applied to the above problem of finding the expected backoff time, the formula becomes simply:

$$\mathrm{E}(c) = \frac{2^c - 1}{2}$$

... or otherwise interpreted as half of the maximum backoff time.

Also note that the summation is a triangular number, such that it is equal to...

$$\frac{(2^c - 1)2^c}{2} = \frac{N(N+1)}{2}$$

...which cancels with the denominator outside the summation to leave only...

$$\frac{N}{2} = \frac{2^c - 1}{2}$$

See also [edit]

- Control theory

References [edit]

1. ^ "IEEE Standard 802.3-2008" (PDF). IEEE. Retrieved 22 September 2010.
2. ^ *Computer Networks, 5th Edition, p. 303*, Tanenbaum
3. ^ *Computer Networks*, Peterson and Davie

- This article incorporates public domain material from the General Services Administration document "Federal Standard 1037C".

Categories: Networking algorithms | Ethernet | Scheduling algorithms