KIP-144: Exponential backoff for broker reconnect attempts

This KIP mostly describes the implementation of PR #1523 by Dana Powers.

Status

Current state: *Adopted*

Discussion thread: *here*

JIRA: Unable to render Jira issues macro, execution error.

Please keep the discussion on the mailing list rather than commenting on the wiki (wiki discussions get unwieldy fast).

Motivation

The client currently uses a constant backoff policy, configured via 'reconnect.backoff.ms' (default is 100ms). To reduce network load during longer broker outages, it would be useful to support an optional exponentially increasing backoff policy.

Public Interfaces

Introduce a new client config for producer consumer and admin client `reconnect.backoff.max.ms` defined as:

The maximum amount of time in milliseconds to wait when reconnecting to a broker that has repeatedly failed to connect. If provided, the backoff per host will increase exponentially for each consecutive connection failure, up to this maximum. To avoid connection storms, a randomization factor of 0.2 will be applied to the backoff resulting in a random range between 20% below and 20% above the computed value.

This config will default to 1000ms if `reconnect.backoff` is not set explicitly. Otherwise, it will default to the same value as `reconnect.backoff`.

No broker configs will be introduced so this feature will not be supported for inter-broker connections where the benefit is less clear (there are typically less brokers than clients and exponential backoff for inter-broker connections could delay recovery of cluster health).

Proposed Changes

We will modify `ClusterConnectionStates`, which is used by `NetworkClient`, so that `reconnectBackoffMs` is updated after every successful or failed connection attempt. The implementation will behave as specified in the Public Interfaces section.

Compatibility, Deprecation, and Migration Plan

For users who have not set `reconnect.backoff` explicitly, the default behaviour will change so that the backoff will grow up to 1000ms. Because we expect most users not to change these settings, we think it's important to choose good defaults. For users who have set `reconnect.backoff` explicitly, the behaviour will remain the same as they may have specific requirements.

Rejected Alternatives

1. Provide a pluggable interface for reconnect attempts like in KIP-53: it seems like we can provide the desired functionality via a single config so pluggability is not worth the additional complexity.
2. Default `reconnect.backoff.max.ms` to the same value as `reconnect.backoff.ms` so that existing behaviour is always maintained: for the reasons explained in the compatibility section
3. Default `reconnect.backoff.max.ms` to `1000ms` unconditionally: for the reasons explained in the compatibility section.