KIP-861: Add BytesIn/BytesOut per listener metrics

This page is meant as a template for writing a KIP. To create a KIP choose Tools->Copy on this page and modify with your content and replace the heading with the next KIP number and a description of your issue. Replace anything in italics with your own description.

Status

Current state: *"Under Discussion"*

Discussion thread: *here*

JIRA: *here*

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

Motivation

In managed Kafka service world, the service provider needs a way for billing the customer by the network usage inbound/outbound. To do that we have the following options:

1. Counting the network usage via the metrics in other component, ex: proxy software like nginx, HAProxy. That works only for the connection from /to external endpoints. Besides, using the metrics in proxy software will have some over-counted number because not every network package entering Kafka in the end, ex: when an IP is rejected by the proxy software, or authentication failure... etc. But all the package flow will be counted into the metrics in proxy software.
2. Using the `BrokerTopicMetrics` in Kafka.

Description	Mbean name	Normal value
Byte in rate from clients	kafka.server:type=BrokerTopicMetrics, name=BytesInPerSec,topic=([-.\w]+)	Byte in (from the clients) rate per topic. Omitting 'topic=(...)' will yield the all-topic rate.
Byte in rate from other brokers	kafka.server:type=BrokerTopicMetrics, name=ReplicationBytesInPerSec,topic=([-.\w]+)	Byte in (from the other brokers) rate per topic. Omitting 'topic=(...)' will yield the all-topic rate.
Byte out rate to clients	kafka.server:type=BrokerTopicMetrics, name=BytesOutPerSec,topic=([-.\w]+)	Byte out (to the clients) rate per topic. Omitting 'topic=(...)' will yield the all-topic rate.
Byte out rate to other brokers	kafka.server:type=BrokerTopicMetrics, name=ReplicationBytesOutPerSec,topic=([-.\w]+)	Byte out (to the other brokers) rate per topic. Omitting 'topic=(...)' will yield the all-topic rate.

   While this metrics can count bytes in (ex: client produce records) and bytes out (ex: consumer fetch records) to clients and to brokers in all topics, there are still some network packages are not included, ex: consumer heartbeat, producer send retry, authentication, authorization...etc.
3. Kafka also exposed request size in bytes per request type:

Description	Mbean name	Normal value
Request size in bytes	kafka.network:type=RequestMetrics,name=RequestBytes,request=([-.\w]+)	Size of requests for each request type.

However, the metrics is in "histogram" type, so it needs further process to sum all the data. Besides, it also expose request size, not response size. And grouping by request type is not helpful in our use case.

Therefore, we're going to expose metrics for bytes in/bytes out per listener. This can help precisely calculate the network usage externally or internally.

Public Interfaces

Two new metrics will be added:

Description	Mbean name	Normal value
Byte in rate per listener.	kafka.network:type=RequestMetrics,name=BytesInPerSec,listener=([-.\w]+)	Byte in rate per listener.
Byte out rate per listener.	kafka.network:type=ResponseMetrics,name=BytesOutPerSec,listener=([-.\w]+)	Byte out rate per listener.

Proposed Changes

Two new metrics will be added.

Compatibility, Deprecation, and Migration Plan

- *What impact (if any) will there be on existing users?*
- *If we are changing behavior how will we phase out the older behavior?*
- *If we need special migration tools, describe them here.*
- *When will we remove the existing behavior?*

Test Plan

Describe in few sentences how the KIP will be tested. We are mostly interested in system tests (since unit-tests are specific to implementation details). How will we know that the implementation works as expected? How will we know nothing broke?

Rejected Alternatives

If there are alternative ways of accomplishing the same thing, what were they? The purpose of this section is to motivate why the design is the way it is and not some other way.