package com.twitter.product\_mixer.core.quality\_factor

import com.twitter.util.Duration

import com.twitter.util.Stopwatch

case class LinearLatencyQualityFactor(

override val config: LinearLatencyQualityFactorConfig)

extends QualityFactor[Duration] {

private val delayedUntilInMillis = Stopwatch.timeMillis() + config.initialDelay.inMillis

private var state: Double = config.qualityFactorBounds.default

override def currentValue: Double = state

override def update(latency: Duration): Unit = {

if (Stopwatch.timeMillis() >= delayedUntilInMillis) {

if (latency > config.targetLatency) {

adjustState(getNegativeDelta)

} else {

adjustState(config.delta)

}

}

}

override def buildObserver(): QualityFactorObserver = LinearLatencyQualityFactorObserver(this)

private def getNegativeDelta: Double =

-config.delta \* config.targetLatencyPercentile / (100.0 - config.targetLatencyPercentile)

private def adjustState(delta: Double): Unit = {

state = config.qualityFactorBounds.bounds(state + delta)

}

}