Reliability

Web Push; IETF ninety-something; somewhere

Problem

Mobile devices are **always** offline
Push services (often) provide a reliable delivery function
Push messages are stored and then:
> sent when the user agent shows signs of life
> sent periodically and (hopefully) wakes the user agent

Acknowledgments

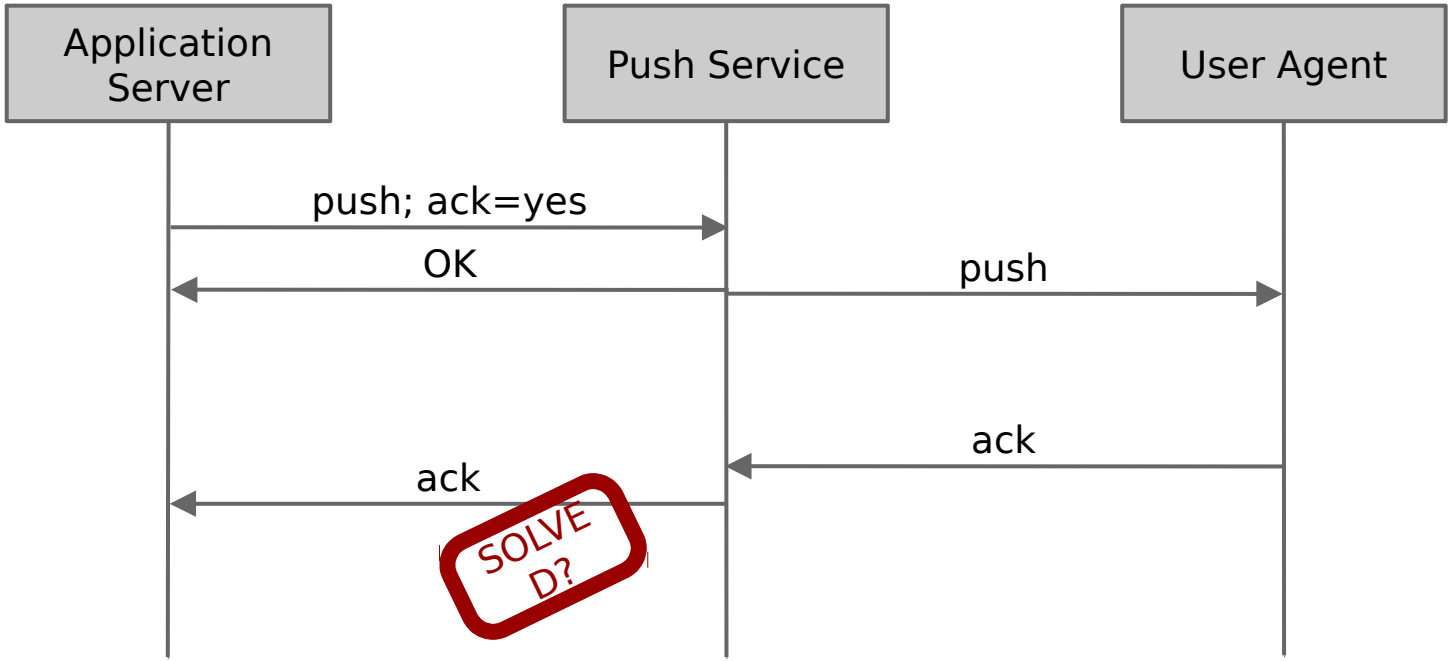
Sometimes the push service delivers a message
... and sometimes it gives up

Applications want to know about both
... so that they can build reliable systems

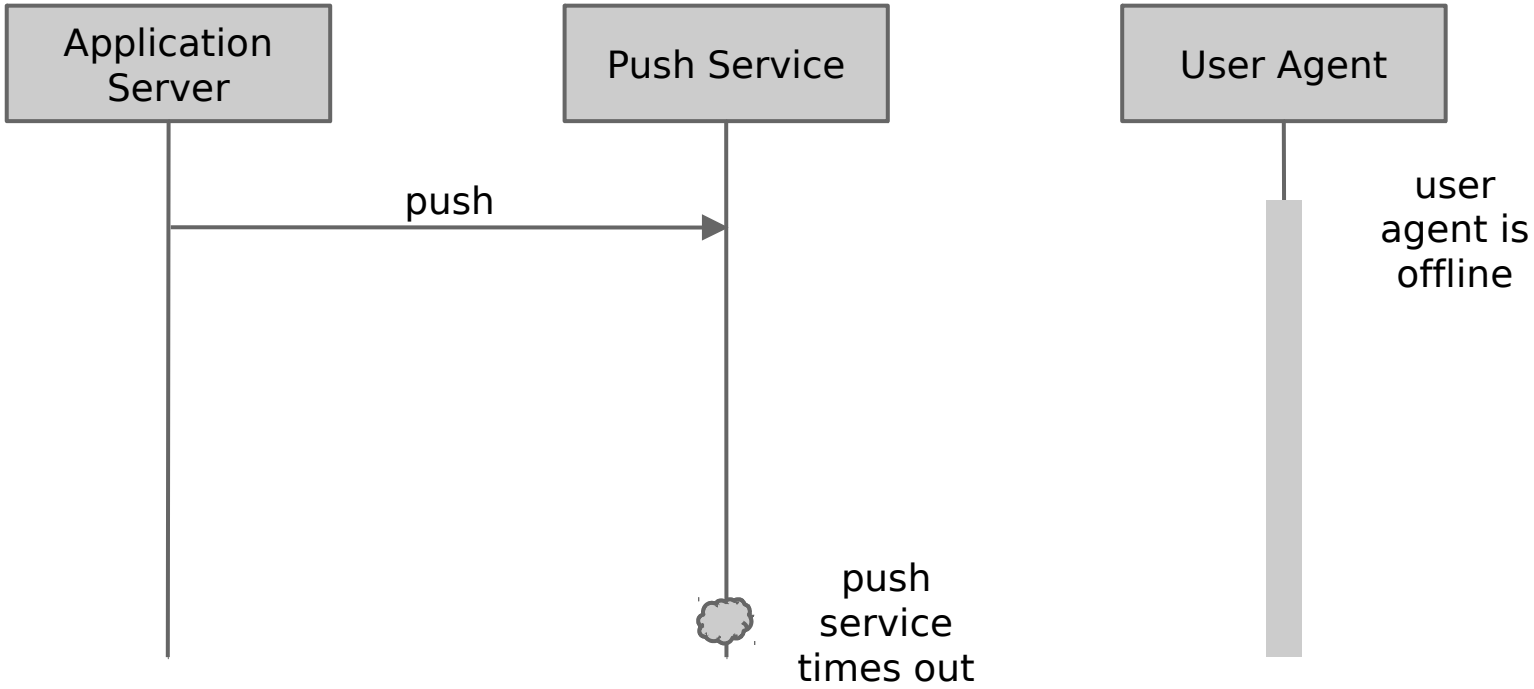
User agent acknowledges every message

This enables other reliability options
Making it optional is possible, but saves little
... but optional parts increase complexity

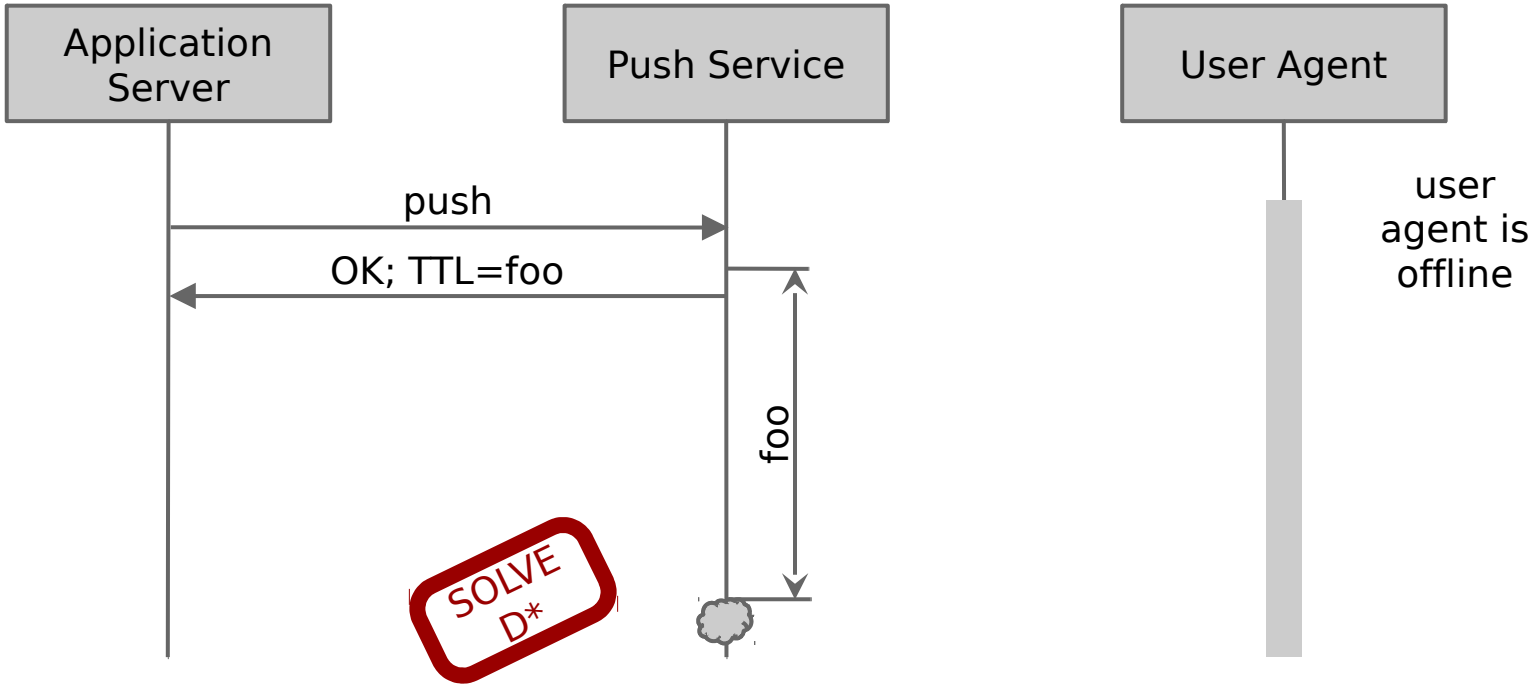
Scenario 1: Positive Acknowledgment



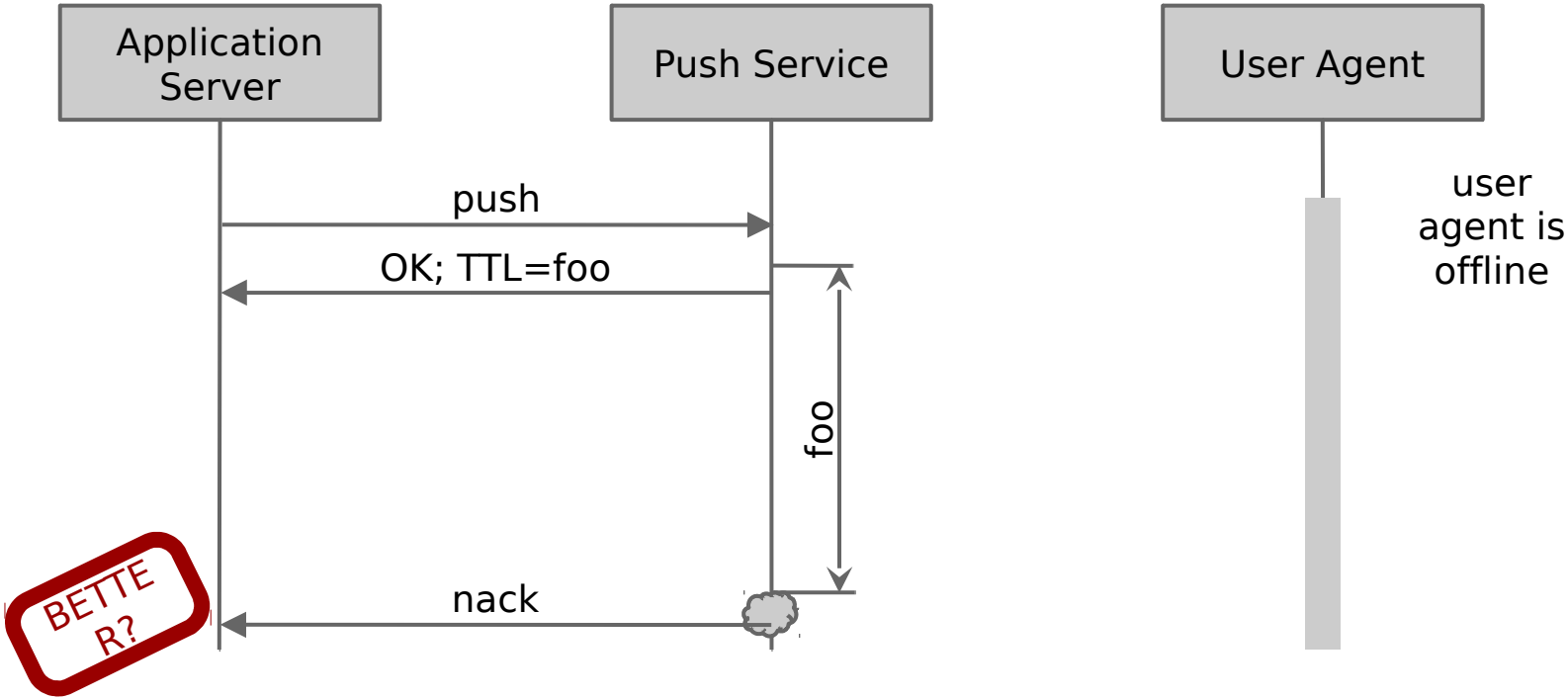
Scenario 2: Push Message Timeout



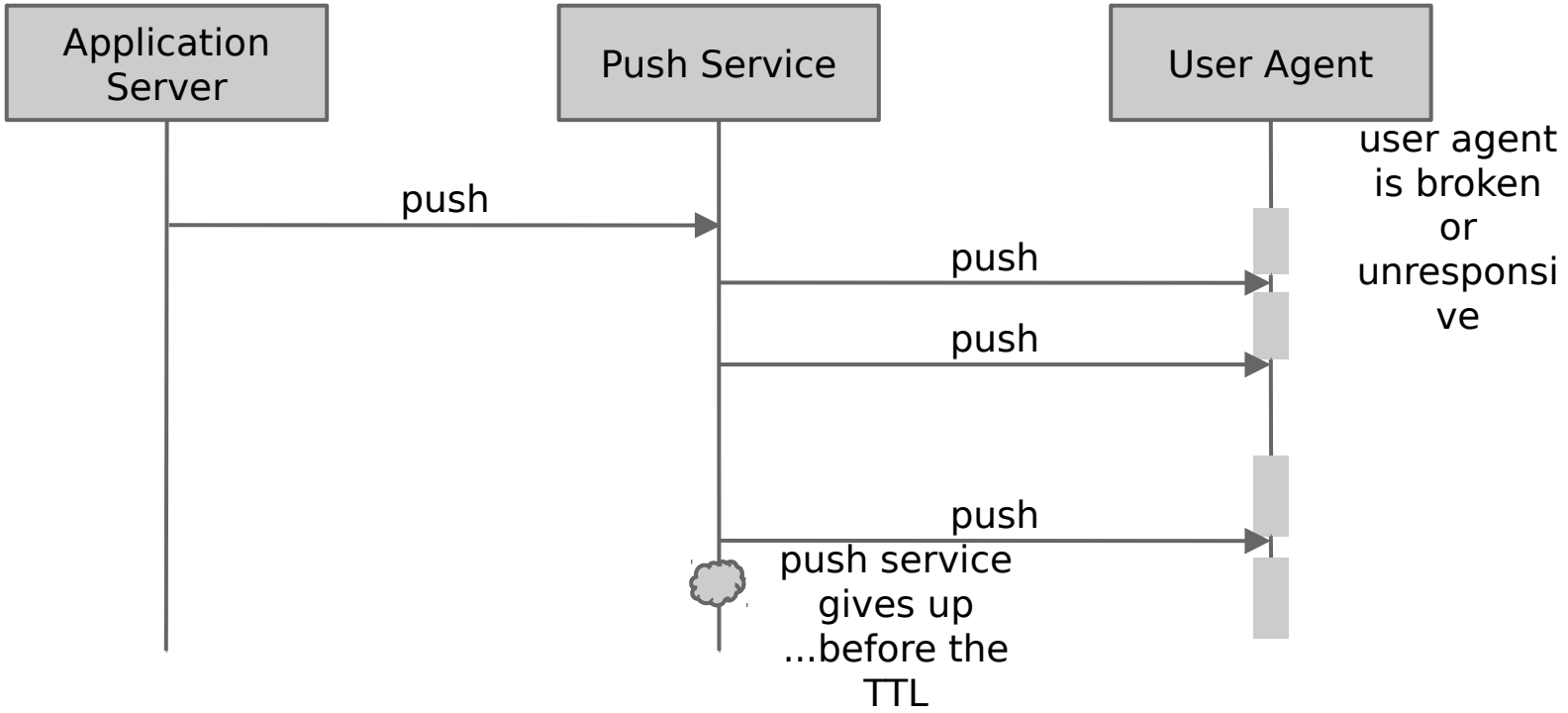
Scenario 2: Push Message Timeout



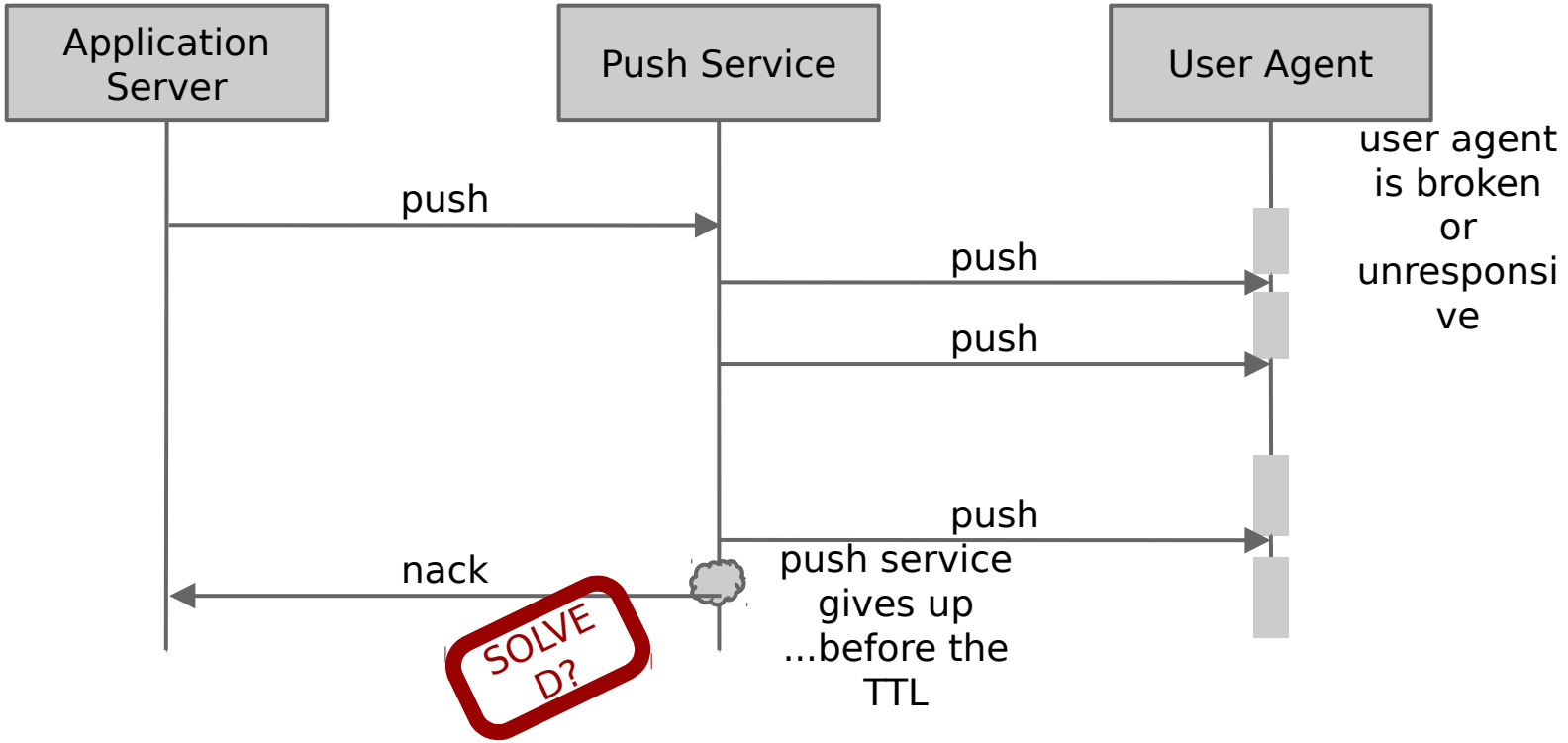
Scenario 2: Push Message Timeout



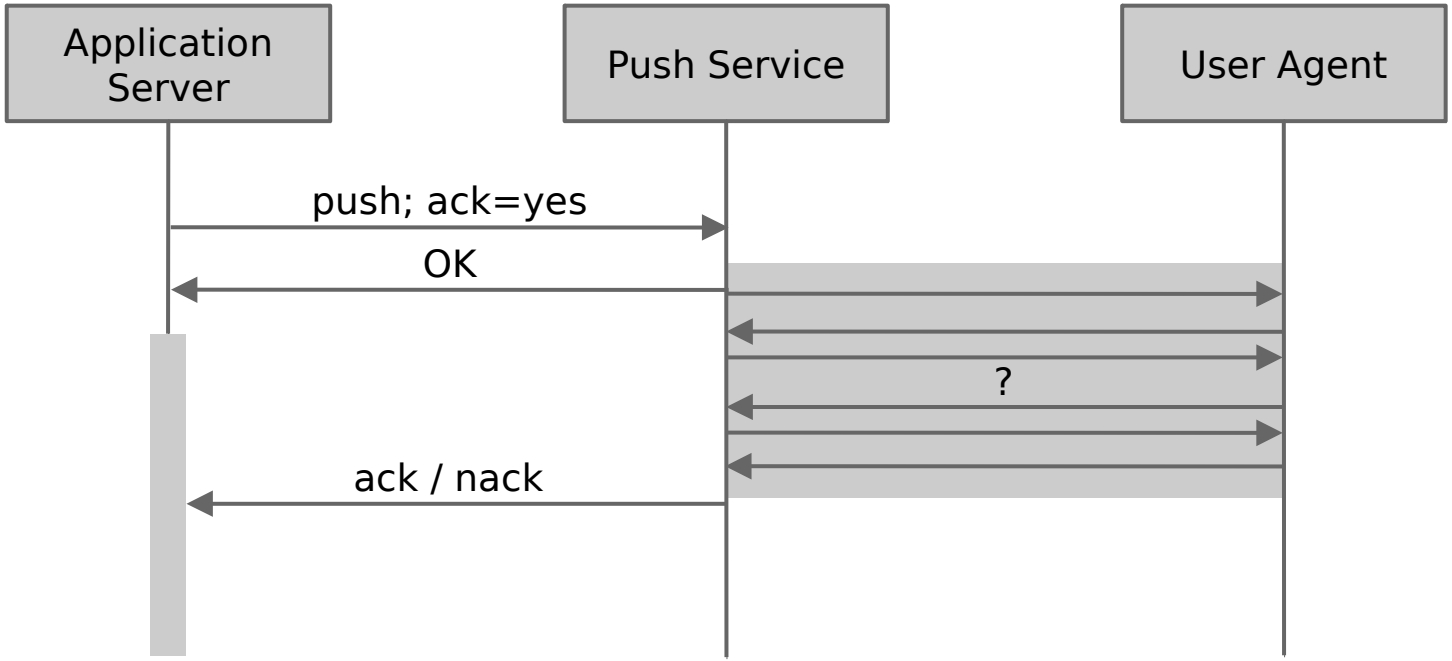
Scenario 3: Push Service Gives Up



Scenario 3: Push Service Gives Up



Scenario 4: Acknowledgment Failure



Reliability for acknowledgments

What level of reliability is needed?
What protocol support does it need?
Acknowledgements provide end-to-end reliability for push messages; do acknowledgments need the same end-to-end guarantees?

Option 1: End-to-end Acknowledgment

No reliability, only end-to-end acknowledgment
If user agent is offline, message is lost
If application server is offline, ack is lost

Demonstrably unacceptable

Option 2: Push Reliability Only

Push messages are stored up to the agreed TTL
... and retried some number of times
... acknowledgments are not
Assumes that the application server is online

Option 3: Full Reliability

Push messages are stored and retried
Acknowledgments are stored and retried
At the TTL, the push service gives up

Note: it makes no sense to have a different TTL for push messages and acknowledgments

Option 4: Push Service Chooses

Let the push service decide between push reliability only and acknowledgment reliability

This could be signaled with the TTL, e.g.,
> TTL: 100; reliable-ack

Proposal

Pick an option:
> no reliability
> push reliability only
> full reliability always
> optional acknowledgment reliability

Proposal

Pick an option:

~~> no reliability~~

~~> push reliability only~~

> full reliability always

~~> optional acknowledgment reliability~~

Negative Acknowledgments

Prior to TTL, if the push service gives up on a message (or an acknowledgment) > signal the error to the application server

Even (or especially) with full reliability, there is no point in signaling the expiration of the TTL

Acknowledging Acknowledgments

This seems absurd
... but acknowledgments can carry data elsewhere

Proposal: not yet