The Direct Project

Implementation Guide for Delivery Notification in Direct

Version 1.0

29 June 2012

Implementation Guide
for Delivery Notification
in Direct Version 1.0, 29
June 2012

Contents

Change Control 3

Status of this Guide 4

Introduction 4
Overview 4

Requirements 4

1.0 Delivery Notification Messages 5

1.1 Positive Delivery Notification Message 5
1.2 Negative Delivery Notification Message 5

1.3 Delivery Notification Request 5

2.0 Notification Responsibilities for STAs 6
2.1 When Sender and Receiver Use the Same STA 6
2.2 When Sender and Receiver Use Separate STAs 6
2.2.1 Responsibilities of the Receiving STA 6
2.2.2 Responsibilities of the Sending STA 7
2.3 Additional Guidance on Interactions Between Sender and Sender's STA 7

3.0 Implementation Considerations 8
3.1 Message Considerations 8

3.2 Delivery Considerations 8

3.3 Sending Edge Client Considerations 8

4.0 Use Cases 9
4.1 Actors 9

4.2 Delivery Success Flows 9
4.2.1 Successful Flow 1 9
4.2.2 Successful Flow 2 10
4.3 Delivery Failure Flows 12
4.3.1 Failure Flow 1 12
4.3.2 Failure Flow 2 13
4.3.3 Failure Flow 3 15
4.3.4 Failure Flow 4 16
4.3.5 Failure Flow 5 18
4.3.6 Failure Flow 6 20

Appendix A: Notification Failure Considerations 23
Notification Failure Flow 1 23

Notification Failure Flow 2 24

Notification Failure Flow 3 25
Notification Failure Flow 4 26

Notification Failure Flow 5 **27**

Change Control

Date	Version	Description of changes
29-Jun-2012	1.0	Published.
22-Jun-2012	DRAFT-2012062201	Incorporated diagrams into flows. Clarification in Section 2.3 on handoff status. Added Appendix A on notification failure considerations.
06-Jun-2012	DRAFT-2012060601	Typo fixed in Section 2.3.
28-May-2012	DRAFT-2012052801	Initial draft.

Status of this Guide

This document is PUBLISHED.

Introduction

Overview

The Direct Project's *Applicability Statement for Secure Health Transport* specifies that Security/Trust Agents (STAs) MUST issue a Message Disposition Notification (MDN, RFC3798) with a disposition of `processed` upon successful receipt, decryption, and trust validation of a Direct message. By sending this MDN, the receiving STA is taking custodianship of the message and is indicating that it will deliver the message to its destination. While the *Applicability Statement* indicates that additional MDNs may be sent to indicate further processing progress of the message, they are not required. The *Applicability Statement* also does not provide guidance in regards to the actions that should be taken by the sending STA in the event an MDN `processed` message is not received or if the receiving STA cannot deliver the message to its destination after sending the initial MDN `processed` message.

Due to the lack of specifications and guidance in the *Applicability Statement* regarding deviations from normal message flow, STAs implementing only requirements denoted as MUST in Section 3 of the *Applicability Statement* cannot provide a high level of assurance that a message has arrived at its destination.

This document provides implementation guidance enabling STAs to provide a high level of assurance that a message has arrived at its destination and outlines the various exception flows that result in compromised message delivery and the mitigation actions that should be taken by STAs to provide success and failure notifications to the sending system.

Requirements

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC2119.

An implementation is not compliant if it fails to satisfy one or more of the MUST, SHALL, or REQUIRED level requirements for the protocols it implements. An implementation that satisfies all the MUST, SHALL, or REQUIRED level and all the SHOULD level requirements for its protocols is said to be "unconditionally compliant"; one that satisfies all the MUST, SHALL, or REQUIRED level requirements but not all the SHOULD level requirements for its protocols is said to be "conditionally compliant."

1.0 Delivery Notification Messages

Notification messages indicate the disposition of a Direct message (e.g., processed, successfully delivered, unsuccessfully delivered). The *Applicability Statement* requires only one notification message – a Message Disposition Notification (MDN) message with a disposition of `processed` (i.e., `processed` MDN), issued by an STA to indicate that it has successfully received, decrypted, and validated trust for a Direct message. The `processed` MDN indicates only that an STA has taken responsibility for delivering a message to its destination – it does not indicate that the message has been successfully or unsuccessfully delivered to that destination. In order to provide sending systems assurance of delivery, STAs will need to issue and accept additional notifications indicating successful or failed delivery to a destination. These notifications and the method for requesting them are defined below.

1.1 Positive Delivery Notification Message

A positive delivery notification message is issued by an STA upon successful delivery to a destination and SHALL take the form of an MDN conforming to RFC3798 with a `disposition-type` of `dispatched` and an `extension-field` of `X-DIRECT-FINAL-DESTINATION-DELIVERY`.

1.2 Negative Delivery Notification Message

A negative delivery notification message is issued by an STA when delivery to a destination has failed or is considered to have failed and SHALL take one of the following forms:

An MDN conforming to RFC3798 with a `disposition-type` of `failed`, or

A negative Delivery Status Notification (DSN).

1.3 Delivery Notification Request

Notification of positive or negative delivery of a Direct message to its destination is requested in the form of an MDN request as specified by Section 2.1 of RFC3798. The MDN request SHALL contain a `Disposition-Notification-Options` header as specified by Section 2.2 of RFC3798 with a parameter named `X-DIRECT-FINAL-DESTINATION-DELIVERY`. This parameter SHALL have an `importance` of `optional` and a value of `true`.

A Direct message containing a request for notification of delivery SHALL also contain a `message-id` header as specified in RFC5322 to permit automatic correlation with its associated `processed` MDN message and delivery notification message.

2.0 Notification Responsibilities for STAs

In order for a sending system to provide to a sender positive assurance that a Direct message has been delivered to its destination, the STAs involved will need to fulfill certain responsibilities as appropriate to their role in the delivery of the message. These are broken down below based on whether the sender and receiver use the same STA versus separate STAs. Additional guidance is provided around interactions between the sender and the STA through which the sender sends messages.

2.1 When Sender and Receiver Use the Same STA

When both the sender and receiver are served by the same STA, the STA itself can positively determine when delivery to the destination (e.g., receiver's system or inbox) has succeeded or failed. In this environment, in order to provide positive assurance of delivery, the STA SHALL notify or indicate back to the sender successful or failed delivery to the destination (see Section 2.3 Additional Guidance on Interactions Between Sender and Sender's STA in this document for more detail).

2.2 When Sender and Receiver Use Separate STAs

When the sender and receiver are served by two different STAs, the sending STA cannot on its own positively determine in all circumstances when delivery to the destination (e.g., receiver's system or inbox) has succeeded or failed; until the sending STA receives a processed MDN or notification of delivery, it can only assume that the receiving STA did not receive and deliver the message or successfully verify security and trust.

In this environment, to provide positive assurance of delivery, each of the STAs – receiving and sending -- has distinct responsibilities.

2.2.1 Responsibilities of the Receiving STA

The Receiving STA SHALL provide delivery notification messages when requested. Once so requested, the Receiving STA:

> SHALL issue a positive delivery notification message to the Sending STA at time of successful delivery of a Direct message to a destination.
>
> SHALL issue a negative delivery notification message to the Sending STA when delivery of a Direct message to a destination fails or is considered to have failed.

The obligation to issue an MDN indicating positive or negative delivery overrides any applicable requirement in Section 2.1 of RFC3798 limiting the number of MDNs that can be issued for a recipient. That is, a Mail User Agent acting as a Receiving STA SHALL issue:

A `processed` MDN once a message has been received and trust and security has been verified as required by the *Applicability Statement*, and

Either a positive delivery notification upon delivery success or a negative delivery notification upon delivery failure.

2.2.2 Responsibilities of the Sending STA

When a use case requires notification of delivery for a particular Direct message, the Sending STA:

SHALL request delivery notification messages from Receiving STAs.
SHALL notify or indicate back to the sender failure to deliver to Receiving STAs.

SHALL notify or indicate back to the sender failed or successful delivery to destinations based on any received positive or negative delivery notification messages it receives from Receiving STAs.

SHALL notify or indicate back to the sender failed delivery to a destination if no processed MDN is received from the Receiving STA within a reasonable timeframe.

SHALL notify or indicate back to the sender failed delivery to a destination if no requested delivery notification messages are received from the Receiving STA within a reasonable timeframe.

For additional detail on Sending STA's notifying or indicating back to the sender delivery status, see Section 2.3 Additional Guidance on Interactions Between Sender and Sender's STA in this document.

2.3 Additional Guidance on Interactions Between Sender and Sender's STA

This guide does not stipulate specific status codes or mechanisms indicating success or failure during the initial handoff of a message from the sender to the sender's STA; these will depend on the edge protocols used between the sender and the sender's STA.

Post-handoff, regardless of whether the sender and receiver share the same STA or are served by two separate STAs, the sender's STA SHALL notify the sender of the successful or failed delivery of the original Direct message by delivering a positive or negative delivery notification message as defined in this guide; this delivery notification message MAY not be the actual positive or negative delivery notification that was originally issued by the receiving STA. Once the sender's STA has notified the sender of positive or negative delivery of a Direct message to a particular destination, the sender's STA SHALL provide no further notification to the sender of positive or negative delivery of that particular Direct message to that destination. Whenever possible, the sender's STA SHOULD notify the sending edge client utilizing the same edge protocol that initiated the message. If the original edge protocol cannot be used, the sender's STA SHOULD attempt to notify the sending edge client using an alternative edge protocol, if available. If there is

no suitable alternative edge protocol, the sender's STA SHOULD implement a "dead letter" destination and offer a protocol enabling edge clients to retrieve delivery notifications from the dead letter destination (the use of a "dead letter" destination in this context deviates from the traditional dead letter concept in that notifications, not the original message, will be held).

3.0 Implementation Considerations

3.1 Message Considerations

MDN `processed` messages are intended to be STA to STA notifications to indicate successful receipt and security and trust validation by the receiving STA; delivery of the `processed` MDN to the sending system is not required to provide assurance that a message has been delivered to its destination

While not required by RFC3798, this guide assumes an STA sending a Direct message that requires notification of delivery will correlate `processed` MDNs and delivery notification messages to the original message using the `message-id` header of the original message.

Per Section 3 of RFC3798, a "particular MDN describes the disposition of exactly one message for exactly one recipient", meaning distinct `processed`, `dispatched`, and `failed` MDN messages will be issued for each recipient.

3.2 Delivery Considerations

The final destination is defined as either:

- The message storage location (for use cases where the STA is responsible for providing message storage), or
- The message being transported successfully over the receiving STA's edge protocol to the recipient's edge client (for use cases where the recipient's systems are responsible for message storage).

Error conditions and semantics at the time an edge client hands off a message to its STA are specific to the edge protocol used. For example, SMTP returns status codes synchronously to the edge client upon message handoff; HTTP implements similar semantics.

3.3 Sending Edge Client Considerations

For sending edge client to STA communication in Direct, the edge client generally assumes that the sender's STA will successfully transport the message to the final destination unless an explicit error status is indicated at the time of message handoff from the edge client to the STA or a negative delivery notification is received from the sender's STA at a

later time. However, when conducting transactions within the scope of this guide, the edge client must receive an explicit delivery notification indicating either positive or negative delivery; the status at time of message handoff is important but not sufficient. Using quality of service terms, the edge client will view the STAs Quality of Service (QoS) as *best effort* with the exception that the sender will be notified of either positive or negative message delivery (somewhat similar to USPS certified mail).

Delivery notification messages will be delivered to the sending edge client asynchronously (i.e., after message handoff from the edge client to STA has occurred).

4.0 Use Cases

The use cases below illustrate various exception flows that result in compromised message delivery and the actions that should be taken by STAs to provide success and failure notifications to the sender.

4.1 Actors

Edge client – An application or service sending and receiving messages to and from an STA over an edge protocol. The edge client may also represent a larger entity such as an HIE, EHR, or an aggregate of multiple systems.

Sending STA – The STA containing the source of the message.
Receiving STA – The STA containing the destination of the message.

4.2 Delivery Success Flows

4.2.1 Successful Flow 1

Description

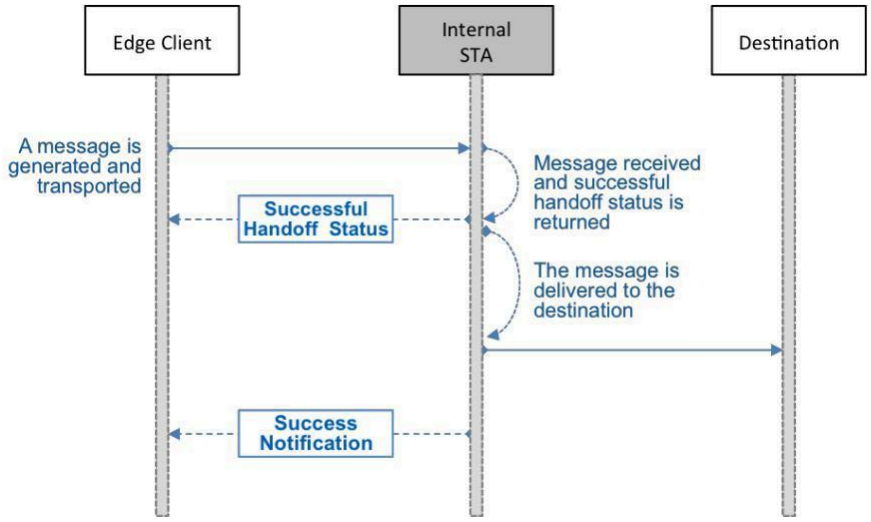
A message is sent from the edge client and successfully delivered to the final destination. In this flow, a single STA serves both the edge client and the recipient.

Applicable Models

Internal STA only

Message Flow

Note: assumes no security and trust processing is necessary



Edge Client	**STA**
1. A message is generated in the edge client and transported to the STA over the edge protocol.	2. The message is received by the STA and a successful handoff status is returned to the edge client.
	3. The message is successfully delivered to the final destination.
	4. A success notification message is delivered to the edge client

4.2.2 Successful Flow 2

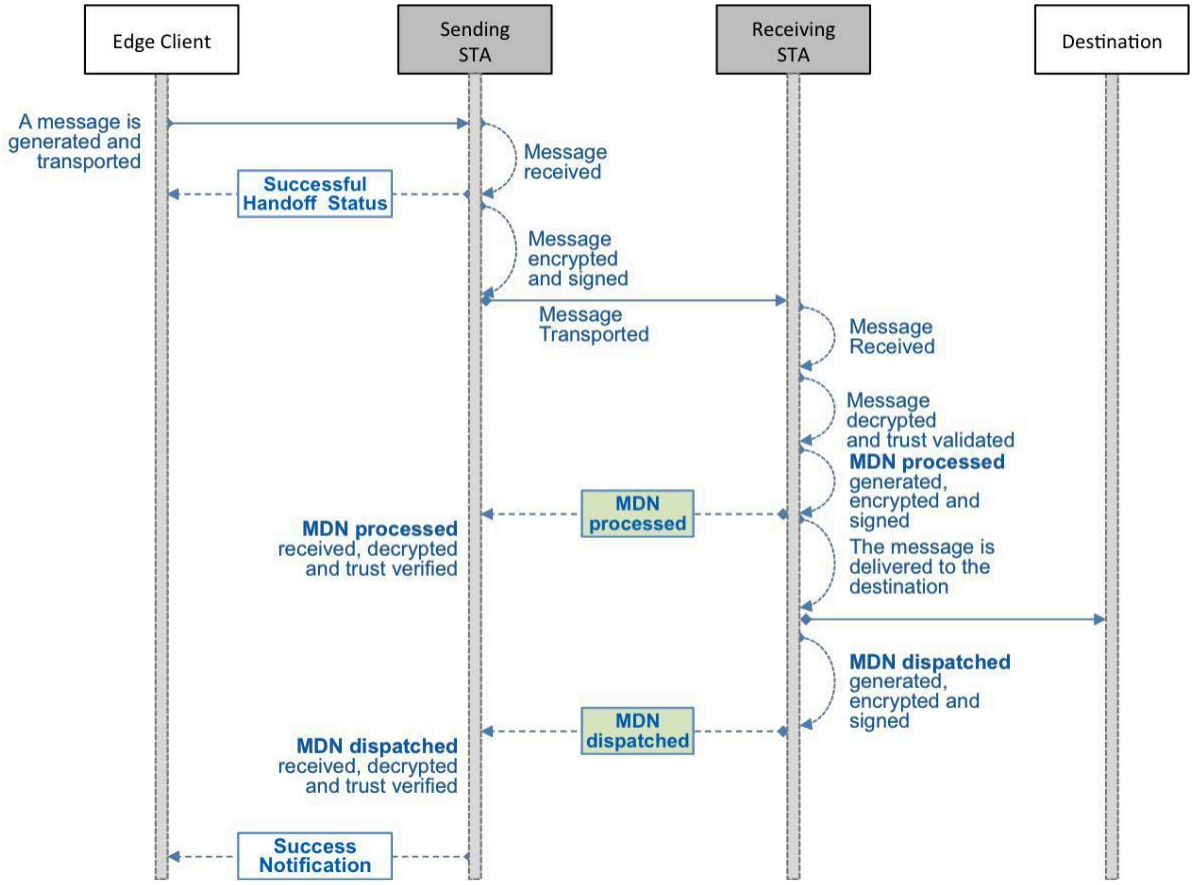
Description

A message is sent from the edge client and successfully delivered to the final destination. In this flow, two STAs, a Sending STA serving the sender and a Receiving STA serving the recipient, are involved.

Applicable Models

STA to STA

Message Flow



Edge Client	**Sending STA**	**Receiving STA**
1. A message is generated in the edge client and transported to the sending STA over the edge protocol	2. The message is received by the sending STA and a successful handoff status is returned to the edge client.	
	3. The message is successfully encrypted and signed.	
	4. The message is transported to the receiving STA.	5. The message is received by the receiving STA for security and trust processing.

		6. The message is successfully decrypted and trust is validated.
	8. The MDN `processed` message is received, decrypted, and trust verified successfully.	7. An MDN `processed` message is created, encrypted, signed, and transported to the sending STA.
		9. The message is successfully delivered to the final destination.
		10. An MDN `dispatched` message is created, signed, and transported to the sending STA.
	11. The MDN `dispatched` message is received, decrypted, and trust verified successfully.	
	12. A success notification message is delivered to the edge client	

4.3 Delivery Failure Flows

4.3.1 Failure Flow 1

Description

The handoff between the edge client and its STA fails.

Applicable Models

STA to STA

Internal STA only

Issue

When the edge client transports the message to its STA over the edge protocol, the STA indicates with an appropriate error condition that it cannot accept the message.

Possible Causes

Edge client is not authenticated or authorized

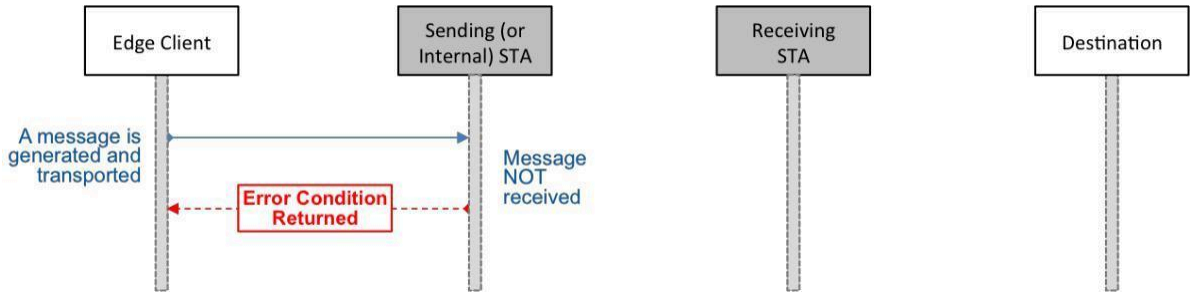
Message is invalid

For internal STA communication, a failure may indicate a message delivery failure if the STA implements synchronous delivery.

Mitigation

At the point the sending STA indicates the error conditions, it is immediately implied that the message cannot be delivered by the sending STA.

Message Flow



Edge Client	**Sending STA**	**Receiving STA**
1. A message is generated in the edge client and transported to the sending STA over the edge protocol	*2. The message is not received successfully by the sending STA and an error condition is immediately returned to the edge client.*	

4.3.2 Failure Flow 2

Description

The sending STA cannot encrypt and/or sign the message or does not trust a recipient due to trust validation issues.

Applicable Models

STA to STA

Issue

Due to an issue in the security and trust process in the sending STA, the message cannot be delivered to the final destination, but the sending STA has already sent a successful handoff status to the edge client.

Possible Causes

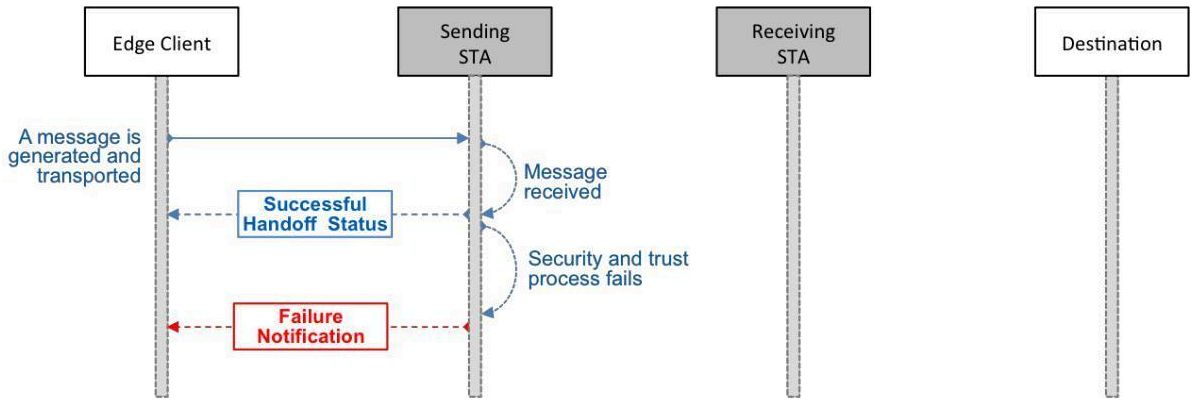
Trust relationship not established with receiving STA

Sender's certificate and/or private key could not be resolved
Sender's certificate is expired or revoked
Recipient's certificate could not be resolved
Recipient's certificate is expired or revoked
Recipient's certificate does not meet receiving STAs certificate policies

Mitigation

Upon failure of the security and trust process, the sending STA must deliver an error notification message to the edge client.

Message Flow



Edge Client	Sending STA	Receiving STA
1. A message is generated in the edge client and transported to the sending STA over the edge protocol	2. The message is received by the sending STA and a successful handoff status is returned to the edge client.	

	3. The security and trust process fails. 4. A failure notification message is generated and delivered to the edge client	

4.3.3

Failure Flow 3

Description

The receiving STA's SMTP infrastructure rejects the message.

Applicable Models

STA to STA

Issue

In some cases, a receiving STA's public facing SMTP server may reject acceptance of a message before ever performing security and trust operations. In these cases, the receiving STA returns an SMTP error code to the sending STA at the time of STA-to-STA transport. The message is never delivered to the final destination.

Possible Causes

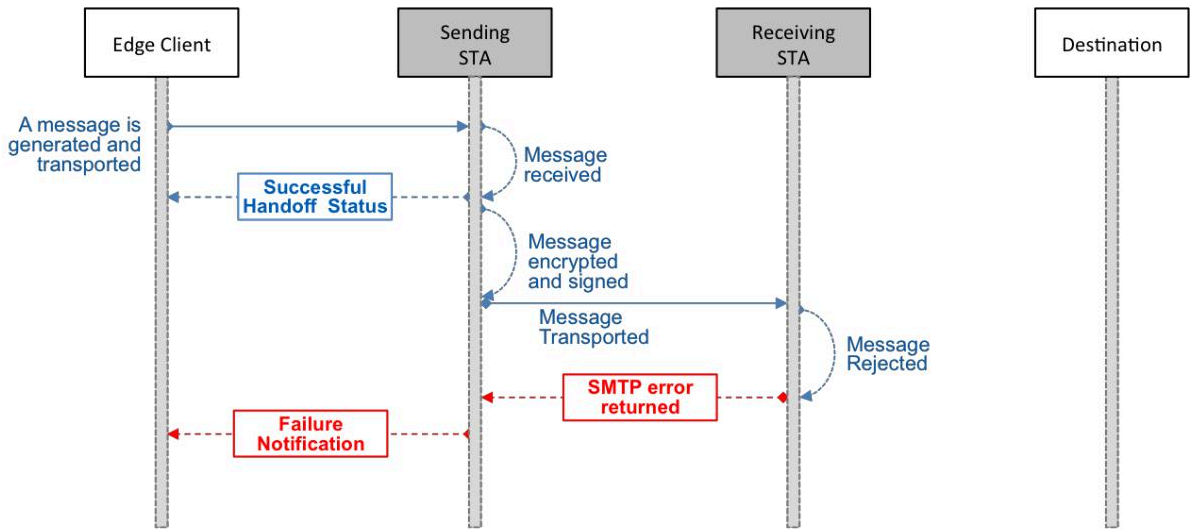
Sending STA has been blacklisted by the receiving STA's SMTP server
Message exceeds size limit
Invalid SMTP header format (invalid address format)
Invalid message format

Mitigation

Mitigation may be dependent on the STA specific deployment model. In some cases, the SMTP error from the receiving STA immediately indicates a failure status to the sending STA, and sending STA can deliver an appropriate error notification to the edge client. In more complex deployment models, the sending STA may not aware of the SMTP error. In these cases, the sending STA will fall back to the mitigation steps in Section 4.3.4 Failure Flow 4.

Message Flow

Note: assumes the sending STA is aware of the SMTP transport error



Edge Client	Sending STA	Receiving STA
1. A message is generated in the edge client and transported to the sending STA over the edge protocol	2. The message is received by the sending STA and a successful handoff status is returned to the edge client.	
	3. The message is successfully encrypted and signed.	
	4. The message is transported to the receiving STA.	*5. The receiving STA rejects the message at time of transport and returns an SMTP error to the sending STA.*
	6. A failure notification message is generated and delivered to the edge client.	

4.3.4 Failure Flow 4

Description

The receiving STA fails to validate the security and trust of the received message.

Applicable Models

STA to STA

Issue

The message cannot be delivered to the final destination because the message does not pass security and trust validation in the receiving STA. Due to the failure, the sending STA is never notified of the failure.

Possible Causes

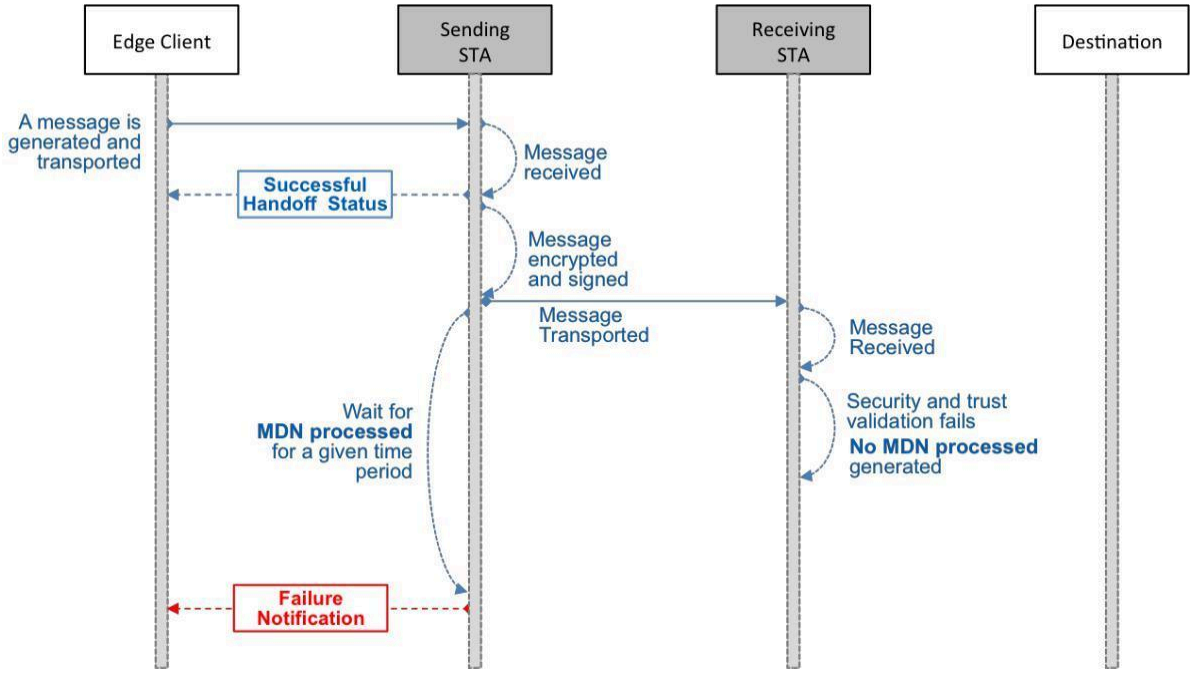
Trust relationship not established with sending STA

Sender's certificate could not be resolved
Sender's certificate is expired or revoked
Sender's certificate does not meet receiving STAs certificate policies
Message is not encrypted or signed

Mitigation

Upon determining that an MDN processed message has not been received after a given time threshold, the sending STA generates error notifications and delivers them to the edge client.

Message Flow



Edge Client	Sending STA	Receiving STA
1. A message is generated in the edge client and transported to the sending STA over the edge protocol	2. The message is received by the sending STA and a successful handoff status is returned to the edge client.	
	3. The message is successfully encrypted and signed.	
	4. The message is transported to the receiving STA.	5. The message is received by the receiving STA for security and trust processing.
		6. Security and trust validation fails. No MDN `processed` *message is sent*.
	7. After a given time period, the sending STA puts the message in a failure status due to the lack of a `processed` MDN. A failure notification is generated and delivered to the edge client.	

4.3.5 Failure Flow 5

Description

The recipient of the message is within the same STA as the sender. This use case assumes that message delivery is not synchronous.

Applicable Models

Internal STA only

Issue

The message cannot be delivered to the final destination within a STA. Depending on the STA implementation, security and trust procedures may not be necessary.

Possible Causes

Delivery components are malfunctioning or unavailable

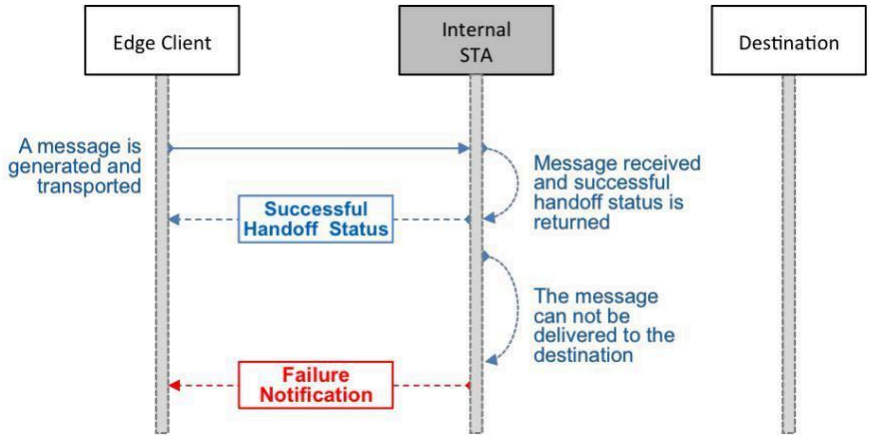
The final destination does not exist (invalid address).
The final destination is full (mail box over quota)

Mitigation

If message delivery fails within a STAs own infrastructure, the STA should be able to unambiguously determine the failure state at any time and deliver a failure notification to the edge client.

Message Flow

Note: assumes no security and trust processing is necessary



Edge Client	**STA**
1. A message is generated in the edge client and transported to the STA over the edge protocol	2. The message is received by the STA and a successful handoff status is returned to the edge client.
	3. The message cannot be delivered to the final destination within the STA.
	4. A failure notification

message is generated and delivered to the edge client.

4.3.6 Failure Flow 6

Description

The receiving STA successfully validates security and trust, but cannot deliver the message to its final destination.

Applicable Models

STA to STA

Issue

Due to a failure condition in the receiving STA, the receiving STA cannot deliver the message to its final destination. The receiving STA has already sent an MDN `processed` message to the sending STA and must notify the sending STA of the new failure condition.

Possible Causes

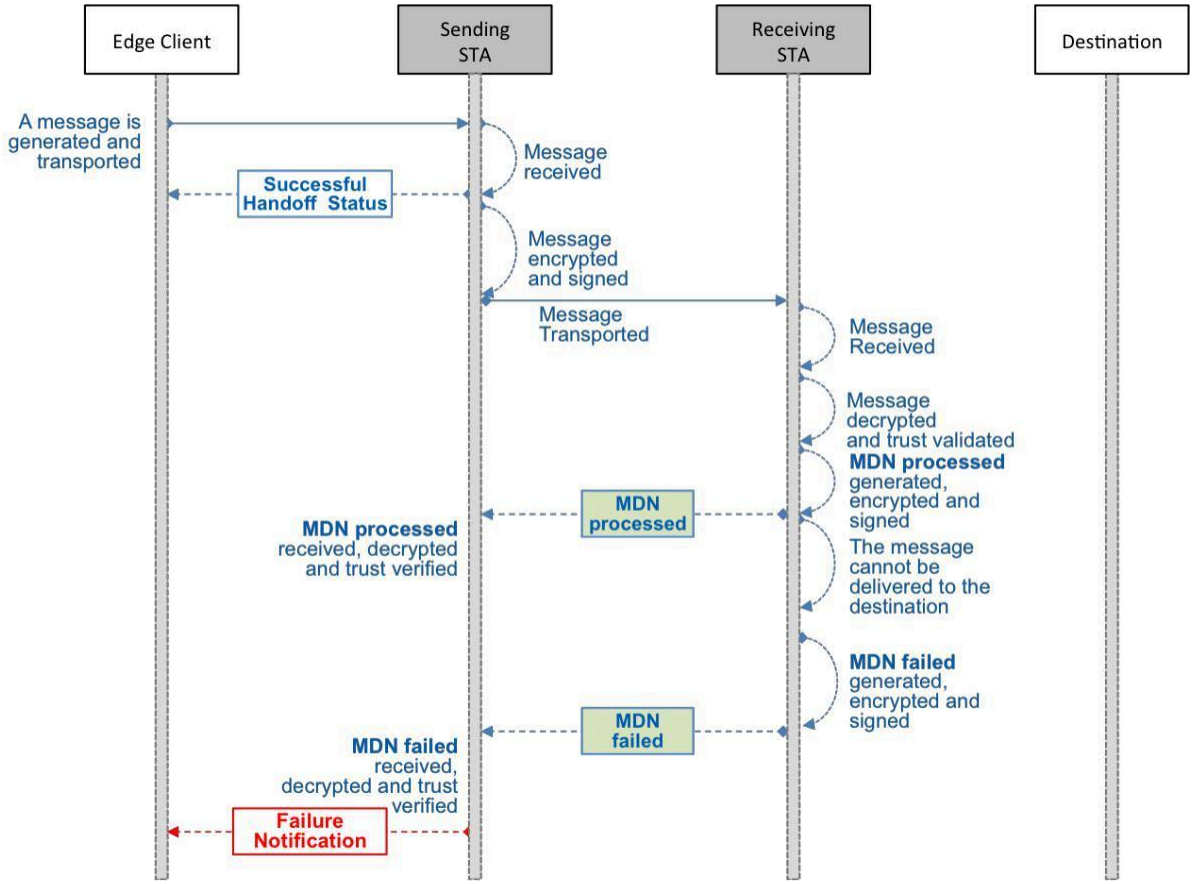
Delivery components are malfunctioning or unavailable

The final destination does not exist (invalid address).
The final destination is full (mail box over quota)

Mitigation

When the receiving STA determines that it cannot deliver the message to the final destination, it generates failure notification and sends it the original sender.

Message Flow



Edge Client	**Sending STA**	**Receiving STA**
1. A message is generated in the edge client and transported to the sending STA over the edge protocol	2. The message is received by the sending STA and a successful handoff status is returned to the edge client.	
	3. The message is successfully encrypted and signed.	
	4. The message is transported to the receiving STA.	5. The message is received by the receiving STA for security and trust processing.
		6. The message is successfully decrypted and trust is validated.

	8. The MDN `processed` message is received decrypted, and trust verified successfully.	7. An MDN `processed` message is created, encrypted, signed, and transported to the sending STA.
		9. The message cannot be delivered to the final destination.
	11. The failure notification message is received decrypted, and trust verified successfully.	10. A failure notification message is created, encrypted, signed, and transported to the sending STA.
	12. A failure notification message delivered to the edge client.	

Appendix A: Notification Failure Considerations

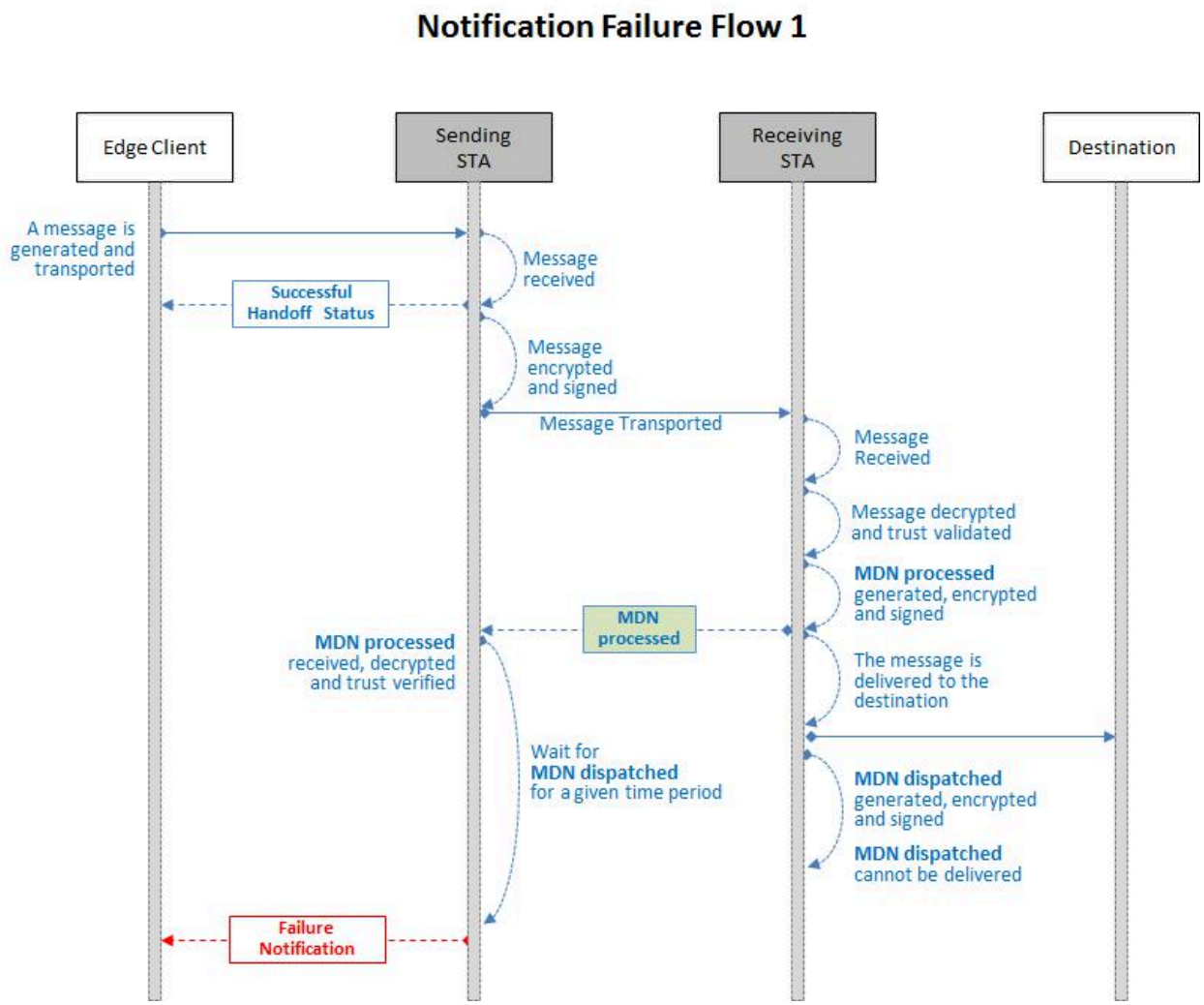
The use cases in section 4 described how sending and receiving STAs should generate delivery notifications. In rare cases, due to the nature of email, the delivery of the notifications themselves can fail or get out of sequence.

Several notification failure flows are described in the sections below. The flows illustrate how STAs should handle notification failures following the guidance on Interactions between sender and sender's STA defined in section 2.3.

Notification Failure Flow 1

> The receiving STA successfully validates security and trust and sends the MDN `processed` message back to the sending STA.
>
> The receiving STA successfully delivers to the final destination, but cannot get the final delivery notification MDN `dispatched` message back to the sending STA.
>
> After a reasonable period of time waiting for the MDN `dispatched` message, the sending STA issues a failure notification to the edge client.

This is a rare case for false negative notification.



Notification Failure Flow 2

The receiving STA successfully validates security and trust but the MDN `processed` message fails to get back to the sending STA.

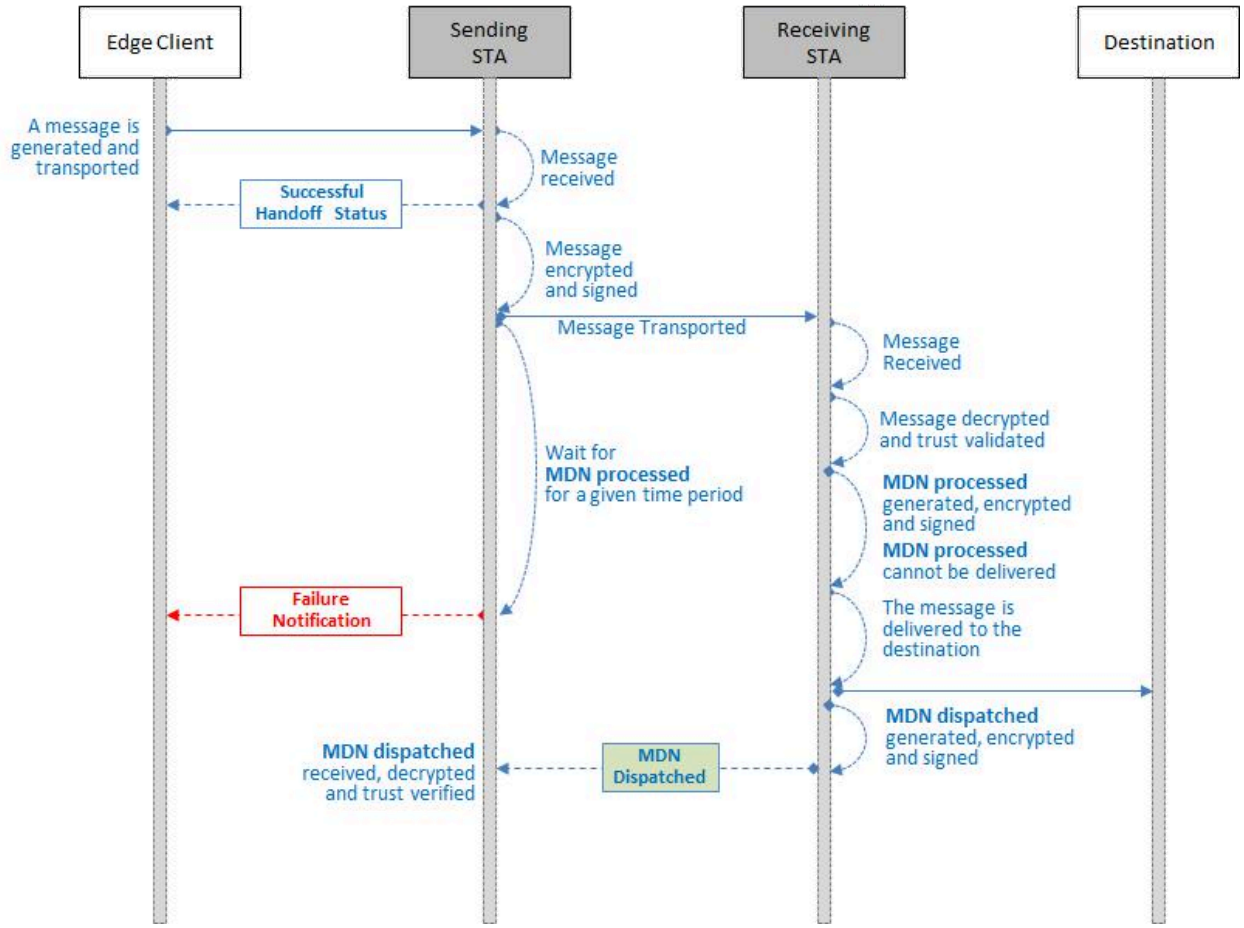
After a reasonable period of time waiting for the MDN `processed` message, the sending STA issues a failure notification to the edge client.

Later on, the message is successfully delivered to the final destination, and a MDN `dispatched` message is sent back to the sending STA.

Since a failure notification has been issued to the edge client, the sending STA will not issue any further notification to the client.

This is a rare case for false negative notification.

Notification Failure Flow 2



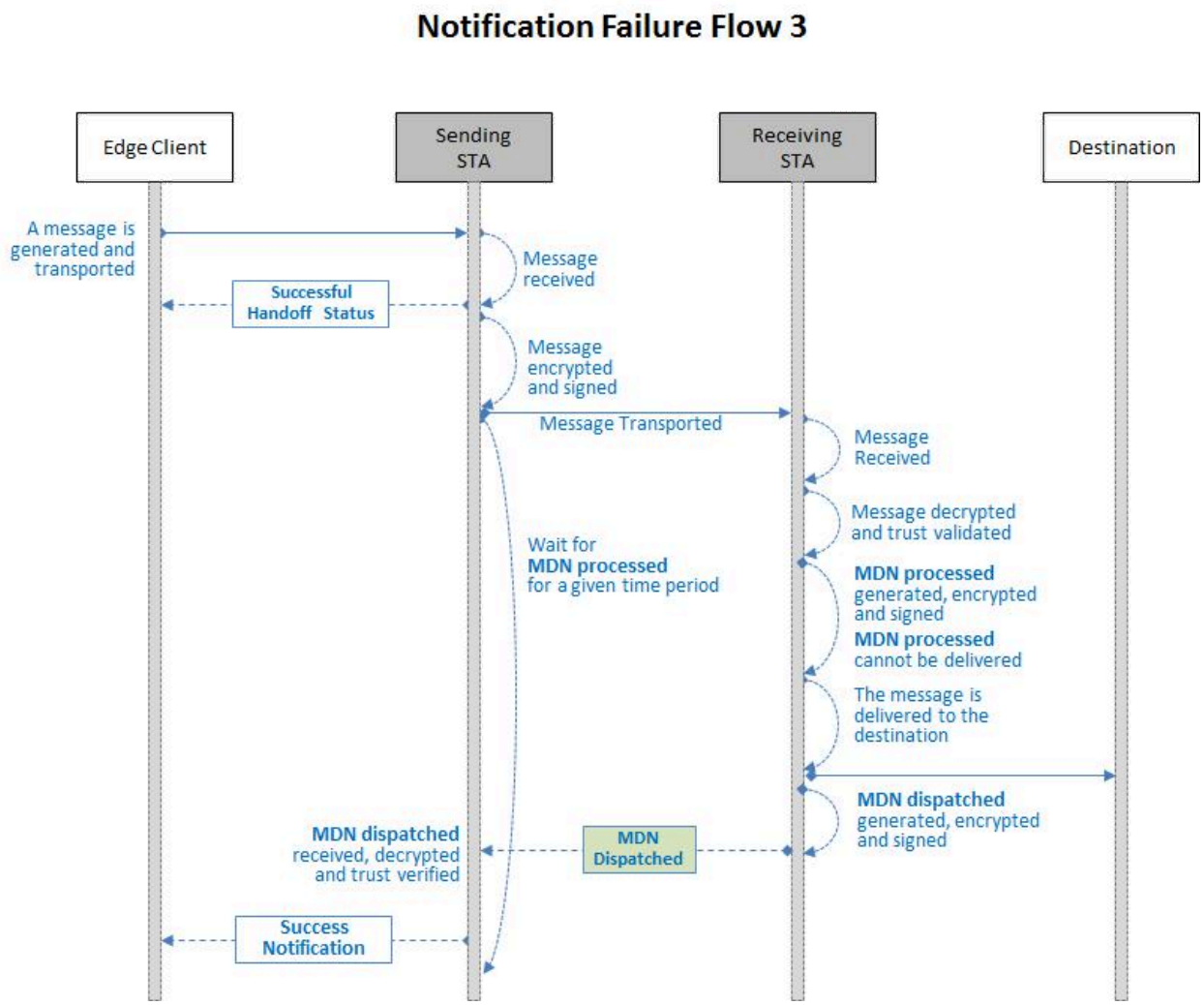
Notification Failure Flow 3

The receiving STA successfully validates security and trust but the MDN `processed` message fails to get back to the sending STA.

Before the wait period for the MDN `processed` message is over, the message is successfully delivered to the destination system and a MDN `dispatched` message is sent back to the sending STA.

The sending STA issues a success notification to the edge client.

No further notification will be issued to the edge client when the wait time for the MDN `processed` message is over.



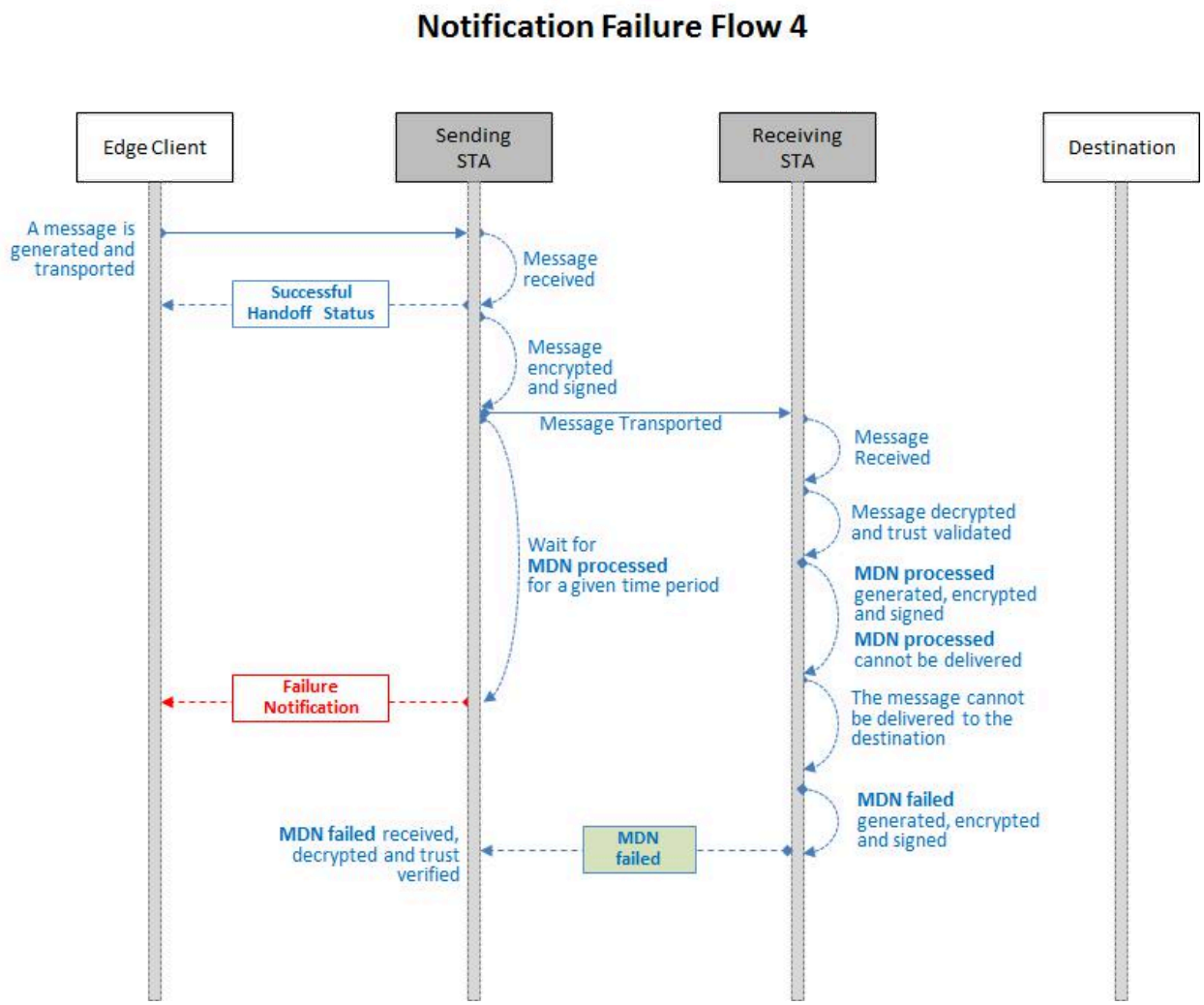
Notification Failure Flow 4

The receiving STA successfully validates security and trust but the MDN `processed` message fails to get back to the sending STA.

After a reasonable period of time waiting for the MDN `processed` message, the sending STA issues a failure notification to the edge client.

Later on, the message fails to be delivered to the destination system and a MDN `failed` message is sent back to the sending STA

Since the sending STA has sent a delivery notification before, no further notification is issued to the edge client.



Notification Failure Flow 5

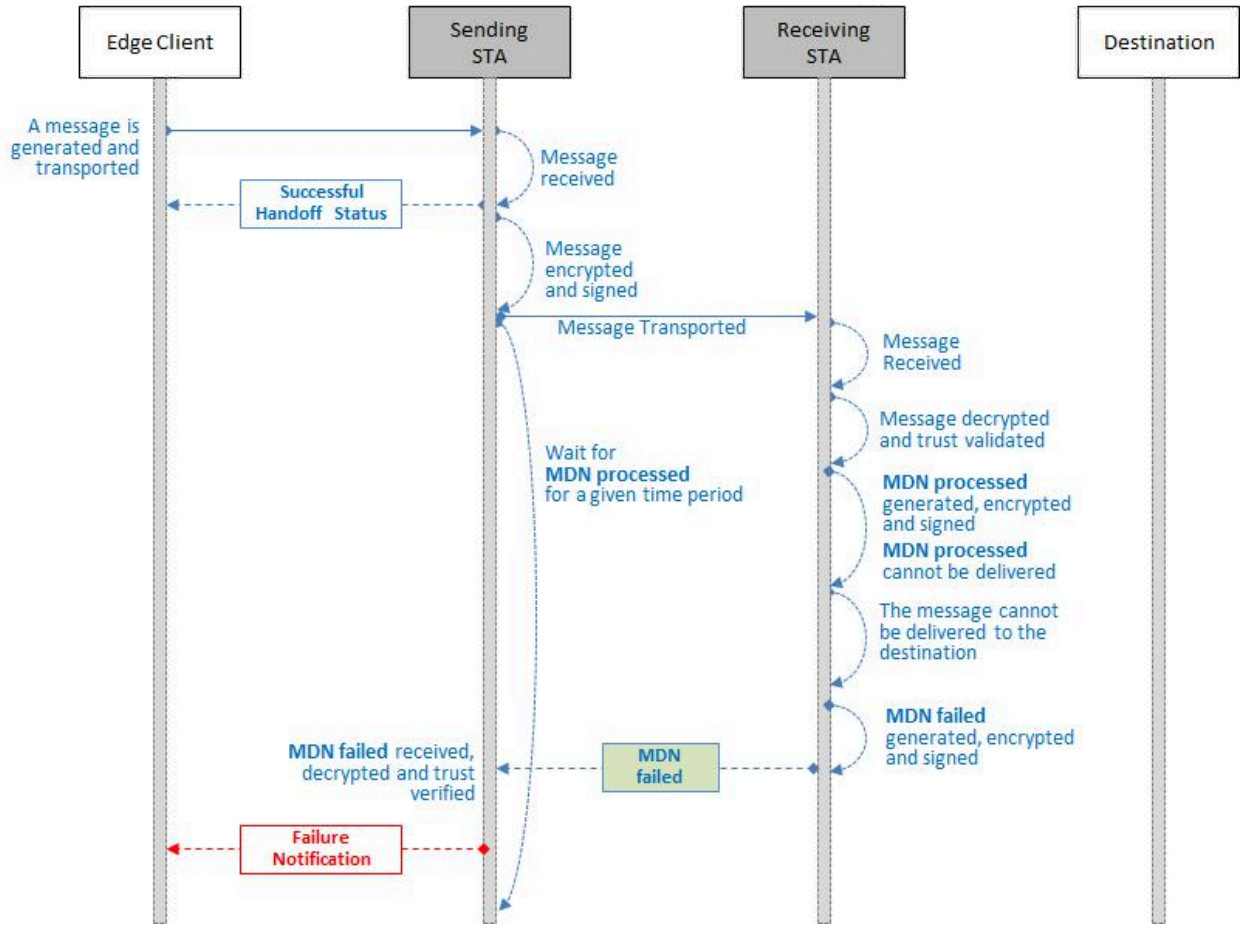
The receiving STA successfully validates security and trust but the MDN `processed` message fails to get back to the sending STA.

Before the wait period for the MDN `processed` message is up, the message fails to be delivered to the destination system and a MDN `failed` message is sent back to the sending STA.

The sending STA issues a failure notification to the edge client.

No further notification will be issued to the edge client when the wait time for the MDN `processed` message is over.

Notification Failure Flow 5



Implementation Guide
for Delivery Notification
in Direct Version 1.0, 29
June 2012