TCG Trusted Network Communications

IF-M Segmentation

Specification Version 1.0
Revision 5
04 April 2016
Published

Contact: admin@trustedcomputinggroup.org

TCG Published

TCG

Acknowledgements

The TCG wishes to thank all those who contributed to this specification. This document builds on considerable work done in the various working groups in the TCG.

Adrien Raffin	AMOSSYS
Henk Birkholz	Fraunhofer
Gerald Maunier	Gemalto
Graeme Proudler	Hewlett Packard Laboratories
Ira McDonald	High North
Andreas Steffen	HSR University of Applied Sciences Rapperswil
Yi Zhang	Huawei
Atul Shah (TNC Co-Chair)	Microsoft
Charles Schmidt (Editor)	MITRE
Steve Hanna	Infineon Technologies
Lisa Lorenzin (TNC Co-Chair)	Pulse Secure
Cliff Kahn	Pulse Secure
Mike Boyle	US Department of Defense
Jessica Fitzgerald-McKay	US Department of Defense
Jonathan Hersack	US Department of Defense
Chris Salter	US Department of Defense
Andrew Cathrow	Verisign

Table of Contents

1 Introduction

1.1 Scope and Audience

The Trusted Network Communications (TNC) Work Group defines an open solution architecture that enables network operators to evaluate and enforce policies regarding endpoint integrity when granting access to a network infrastructure. In this architecture, functional units on endpoints, called Integrity Measurement Collectors (IMCs), send information about an endpoint's system state to corresponding functional units, called Integrity Measurement Verifiers (IMVs), on a Policy Server (such as a Policy Decision Point (PDP) or IETF Network Endpoint Assessment (NEA) Server). IMVs can also send instructions to IMCs indicating the information to collect and/or leading to changes in configuration of the IMC or of the endpoint on which it resides. The information and instructions sent between IMCs and IMVs are expressed in blocks of data called "attributes", which are packaged together as "messages" and sent over the logical IF-M interface that exist between the IMVs and IMCs. This relationship is shown below in Figure 1.

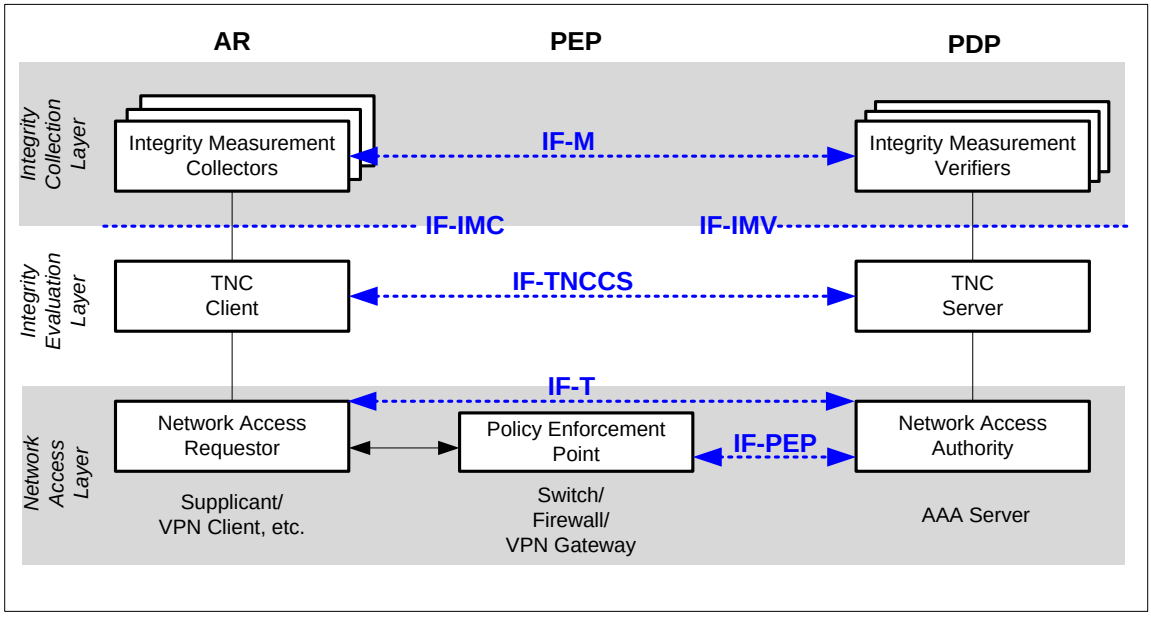

Figure 1 - TNC Architecture

The IF-M interface supports messages up to 4 GB in size[1]. However, messages of this size can cause multiple challenges. First, not all IMCs and IMVs are able to easily handle messages of this size. Even if they are capable of processing messages of such large volume, the processing might be more time consuming than is acceptable, or it might be disruptive of other activities on the computer receiving the message. Similarly, the network connecting an endpoint and a policy server might be low-bandwidth, and transmitting such large messages could end up taking an unacceptably long time and disrupt other communications over the same connection.

The second challenge posed by large messages is related to the multi-level structure of the TNC architecture. As seen in Figure 1, the TNC architecture defines three layers: the Integrity Collection Layer (served by the IF-M interface), the Integrity Evaluation Layer (served by the IF-TNCCS interface), and the Network Access Layer (served by the IF-T interface). Messages are created at

[1] Technically, messages need to be slightly less than 4GB as the size limit includes message headers, as well as headers for the IF-TNCCS interface over which IF-M is sent. However, these headers only consume a few tens of bytes, so for most intents and purposes, their impact on overall message size can be ignored.

the Integrity Collection Layer, such as by an IMC. The complete message is then passed down between successive layers of the TNC architecture until it reaches the Network Access Layer, which is responsible for transmitting the message over the network to the receiving party. The recipient then receives the message at its Network Access Layer, and the message is then passed up the TNC architecture layers to its Integrity Collection Layer, where it is processed. If this exchange between layers is handled by copying messages – a not unreasonable assumption, given that layers might be implemented by separate applications that might not have access to each other's memory space – then this could require up to 12 GB on a single endpoint just to send or receive a single message. In fact, on the recipient side, a single message might be sent to multiple components that operate at the Integrity Collection Layer, and each of these components might be given their own copy of the message. As such, 12 GB would be the minimum amount of memory that would be required by a recipient to handle a 4 GB message. Of course, in practice most messages are nowhere near 4 GB in size, but the fact that the IF-M TLV Binding specification allows such large messages means that all components in the TNC architecture need to be designed to handle such large messages, and users of the TNC architecture cannot safely assume that such large messages will never be encountered.

The IF-M Segmentation specification is designed to help address both of these challenges. To address the first problem, it allows participants in an IF-M exchange to agree upon a maximum allowed message size that is smaller than the theoretical 4 GB limit. To address the second challenge, the specification defines procedures to allow messages to be broken into smaller units, called "segments", which are transmitted one at a time in a controlled manner. This allows both parties to employ a much smaller memory footprint in the TNC architecture stack when handling segmented messages, and allows both parties to control the rate of network bandwidth consumption in their transmission. This specification does not eliminate the possibility of overly large messages – either participant MAY decline (either implicitly, by failing to implement this specification, or explicitly) a request to limit the size of its messages and/or employ segmented message delivery. As such, this specification does not eliminate the need for TNC architecture components or deployments thereof to support the delivery of large messages. However, if the sending and receiving parties do agree upon message size limits and/or on the use of segmented message delivery, this allows both parties to better manage their network and memory usage and reduce the impact of the aforementioned challenges.

IF-M Segmentation is designed to augment other IF-M layer specifications. Other specifications define the attributes used to convey a particular type of information or instruction set over the IF-M interface. This document refers to such specifications as "IF-M binding" specifications. IMC and IMV implementers support a specific set of IF-M bindings. The binding that defines a given attribute is indicated by a field within the IF-TNCCS header called the IF-M Subtype. All attributes within a single message share the same IF-M Subtype and, as such, the message itself can be said to have a single IF-M Subtype. The IF-M Subtype value allows the Integrity Evaluation Layer component (i.e., a TNC Client or a TNC Server) to route the message to the IMCs or IMVs that handle messages of that particular binding. IF-M Segmentation is designed to work "on top of" these IF-M bindings. Specifically, the implementers of IMCs or IMVs might support IF-M Segmentation on top of the specific bindings they support. If they do this, the IMCs and IMVs behave in the way dictated by the binding specification up until a message is to be sent. At this point, if there is an agreement between the IMC and IMV to use IF-M Segmentation, additional size limits are applied and/or the message is transmitted as a series of segments rather than in its entirety. IMCs and IMVs that conform to the IF-M Segmentation specification also support the sending and receiving of special IF-M Segmentation attributes that allow the parties to agree upon size limits. The result of this is that IF-M Segmentation is effectively an optional add-on to any implementation of an IF-M binding. For this reason, use of IF-M Segmentation is specifically designed to only work with the explicit consent of all affected parties. This prevents unexpected behavior if one party in an IF-M binding supports IF-M Segmentation while the other does not.

Before reading this specification any further, the reader should review and understand the TNC architecture as described in TNC Architecture for Interoperability [1]. The reader should also understand the capabilities and requirements common to IF-M interfaces as defined in the TNC IF-

M TLV Binding specification [3]. If the reader is building an IMC that supports IF-IMC, the reader is encouraged to read the TNC IF-IMC Specification [4] prior to reading this specification. If the reader is building an IMV that supports IF-IMV, the reader is encouraged to read the TNC IF-IMV Specification [5] prior to reading this specification.

1.2 Keywords

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this specification are to be interpreted as described in RFC 2119 [2]. This specification does not distinguish blocks of informative comments and normative requirements. Therefore, for the sake of clarity, note that lower case instances of must, should, etc. do not indicate normative requirements.

1.3 Definitions

The following terms are used throughout this document to describe the roles and activities governed by the IF-M Segmentation specification:

IF-M Component - An Integrity Collection Layer component (usually either an IMV or an IMC).

Contracting Party - The IF-M Component that requests a message size limit and/or the use of segmentation on the messages of a particular IF-M Subtype it receives from an Integrity Collection Layer communications partner.

Contracted Party - The IF-M Component that accepts a message size limit and/or the use of segmentation on the messages of a particular IF-M Subtype it sends to a specific Contracting Party.

Segmentation Contract - An agreement between a Contracting Party and a Contracted Party to employ a specific message size limit and/or the use of segmentation on messages of a particular IF-M Subtype sent from the latter to the former. Note that a Segmentation Contract is unidirectional.

Contract Constrained Message - A message generated on a Contracted Party, which is to be sent to a Contracting Party, and which uses the IF-M Subtype identified in a Segmentation Contract between the two parties. Such a message is subject to the limits imposed by this Segmentation Contract.

Base Message - The term given to a complete IF-M message (including the IF-M message header) containing attributes created as part of an IF-M binding OTHER THAN the IF-M Segmentation specification. In other words, the term “Base Message” refers to the complete message that might be broken into segments for transmission in accordance with this specification.

Segmentation Candidate - A Contract Constrained Message whose size is less than or equal to the maximum message size limit of the relevant Segmentation Contract, but is larger than the Segmentation Contract’s maximum segment size.

Message Segment - A portion of a Base Message, smaller than the full Base Message itself, sent from a Contracted Party to a Contracting Party as part of a Segmented Message Exchange.

Segmented Message Exchange - A series of IF-M Segmentation messages exchanged between the Contracted and Contracting Party by which a single Base Message is transmitted as a series of Message Segments.

Null Contract - A Segmentation Contract that imposes no message size limit and prohibits Segmented Message Exchanges. Such a contract is functionally equivalent to there being no Segmentation Contract in place.

Note that the basic units of communication for both the IF-M and IF-TNCCS interfaces are referred to as “messages”. For brevity, in this document, “message” refer to IF-M messages, while IF-TNCCS messages, when mentioned, are always fully identified as “IF-TNCCS messages”.

2 Background

2.1 Role of IF-M Segmentation

IF-M Segmentation serves in a supplementary role to other TNC architecture specifications. As noted before, implementers of IF-M Components that support one or more IF-M bindings might also choose to support IF-M Segmentation to help users manage memory and network bandwidth consumption while sending and receiving messages for those bindings. There is no value to an IF-M Component that supports IF-M Segmentation but does not support other IF-M bindings; without those other bindings, there would be no Base Messages to be subject to any Segmentation Contract.

IF-M Segmentation is designed to be an optional add-on to IF-M Components. For this reason, no IF-M Component can safely assume that a given correspondent supports IF-M Segmentation. The requirements in the IF-M Segmentation specification are designed to ensure that failure by one party to support IF-M Segmentation does not disrupt support for the underlying IF-M binding in unexpected ways.

If IF-M Segmentation is supported, it can help improve resource utilization within the TNC architecture. It allows participants to agree to limits on message size, limit the size of individual network transmissions, limit the amount of memory needed by each layer of the TNC architecture stack, and control how frequently Message Segments are transmitted. Especially for IF-M bindings that can produce large attributes, this ability to manage network and memory consumption can provide a significant benefit in ensuring stable and reliable operations of devices during TNC exchanges and operations.

2.2 Supported Use Cases

This section describes the IF-M Segmentation use cases supported by this specification.

2.2.1 Support Limits on Message Sizes below the Maximum Allowed Size

The IF-TNCCS TLV Binding specification [10] supports IF-M message sizes of up to 4 GB. Sometimes, however, messages of this size would cause unacceptable bottlenecks in transmission or processing. In these situations, an IF-M Components might wish to enter an agreement to limit the size of the messages that they send to each other.

IF-M Segmentation allows IF-M Components to specify an upper size limit for messages sent from one component to another. While a Segmentation Contract is in force, parties can be confident that the messages covered by the contract will be smaller than or equal to the specified message size limits. This allows both components to better manage their memory and network resources so that a single message does not end up disrupting other, possibly more important, activities.

2.2.2 Support Sending Messages Piecemeal

Sometimes it is important to send a large message between IF-M Components, but doing so all at once would be disruptive to either or both IF-M Components. In this situation, it would be advantageous to break the large message into smaller segments and send those segments individually so that the load on the network and components remains manageable.

IF-M Segmentation supports this by allowing a component to indicate a maximum segment size. Messages that exceed the maximum segment size and which cannot be condensed using other methods would be broken into multiple segments and transmitted individually. After the initial segment is sent, subsequent segments are only sent at the request of the recipient, allowing the recipient to control the rate of delivery. This can allow the delivery of large messages without incurring the heavy network and memory loads that might normally be experienced with a single large message.

2.2.3 IF-M Use Cases

IF-M Segmentation attributes are intended to operate over the IF-M interface and, as such, are intended to meet the use cases set out in the IF-M TLV Binding specification.

2.3 Non-supported Use Cases

Some use cases not covered by this version of IF-M Segmentation include:

2.3.1 Unilateral Imposition of Segmentation Contracts

IF-M Segmentation requires support from both participating IF-M Components in order to perform its functions. If only one component is willing or able to support IF-M Segmentation features, a Segmentation Contract cannot be negotiated, so no message size limits can be applied or expected.

2.3.2 Support for Non-Exclusive Delivery

IF-M Segmentation requires exclusive delivery in order to avoid an IF-M Component from inadvertently becoming party to a Segmentation Contract that it did not negotiate. Exclusive delivery was introduced in IF-IMC 1.3, IF-IMV 1.3, and IF-TNCCS 2.0. Consequently, IF-M Segmentation cannot be used in TNC environments implementing versions older than IF-IMC / IF-IMV 1.3 and IF-TNCCS 2.0. See section 2.6 for details.

2.3.3 Support for Contract Initiation or Modification by Contracted Party

IF-M Segmentation does not support would-be Contracted Parties requesting Segmentation Contracts with Contracting Parties or modifying the size limits specified in an existing Segmentation Contract, although Contracted Parties can unilaterally cancel an existing Segmentation Contract. The Contracting Party MUST always be the one that initiates the process of creating or modifying a Segmentation Contract. See sections 3.2 and 3.3 for details.

2.4 Specification Requirements

Below are the requirements that the IF-M Segmentation specification is required to meet in order to successfully play its role in the TNC architecture.

- Efficient

  The TNC architecture enables delay of network access until the endpoint is determined not to pose a security threat to the network based on its asserted integrity information. To minimize user frustration, IF-M Segmentation ought to minimize overhead delays and make IF-M communications as rapid and efficient as possible. In fact, helping to manage memory and network consumption is one of the primary goals of the IF-M Segmentation specification, so care needs to be taken that its mechanisms do not impose significant overhead of their own.

- Loosely Coupled to IF-M Bindings

  One of the goals of IF-M Segmentation is that it be usable with any IF-M binding, including bindings that are vendor-developed. As such, IF-M Segmentation makes no assumptions regarding the structuring or behavior of IF-M attributes associated with other bindings, beyond the assumption that they conform to the relevant requirements set out in the IF-TNCCS [10] and IF-M TLV [3] specifications.

- Interoperable

  This specification defines how IMCs and IMVs can agree upon message size limits and exchange segmented messages. Therefore a key goal for this specification is ensuring that all IMCs and IMVs that conform to this specification, regardless of the vendor who created them, are able to interoperate in their performance of these duties.

- Operate Correctly in the Presence of Non-Compliant IMCs and IMVs

  IF-M Segmentation is an optional add-on for IF-M Components. For this reason, one component cannot automatically assume that a component with which it is communicating supports IF-M Segmentation. When a component that supports IF-M Segmentation attempts to establish a Segmentation Contract, it is important that this be done in a manner that is not disruptive to non-supporting IF-M Components and ensures that all parties have a clear understanding and agreement as to any additional constraints on message delivery that might be in place.

2.5 Non-Requirements

There are certain requirements that the IF-M Segmentation specification explicitly is not required to meet. This list is not exhaustive.

- End to End Confidentiality and Integrity

  IF-M Segmentation does not include mechanisms to ensure the confidentiality or integrity of the messages or Message Segments that are sent in accordance with a given Segmentation Contract. Confidentiality is generally provided by the underlying transport protocols, such as the IF-T Binding to TLS [8] or IF-T Binding to EAP [9]. Should users wish confidentiality protection of assessment instructions or results, this needs to be provided by selecting appropriate transport protocols.

2.6 Assumptions

These are the assumptions that IF-M Segmentation makes about other components in the TNC architecture.

- Reliable Message Delivery

  The TNC Client and TNC Server are assumed to provide reliable delivery for IF-M messages between IF-M Components. In the event that reliable delivery cannot be provided, the TNC Client or TNC Server is expected to terminate the connection.

- Support for Exclusive IF-TNCCS Message Delivery

  IF-IMV [5] and IF-IMC [4] 1.3 introduce the concept of exclusive message delivery when used in conjunction with IF-TNCCS 2.0. Exclusive delivery allows an IF-M message sender to address a message to a specific IF-M Component on the target device. This differs from non-exclusive delivery – the only delivery method possible before IF-IMV 1.3 – where an IF-M message is passed to every IF-M Component that is registered to receive messages of a given IF-M Subtype on the target device. To avoid the possibility of IF-M Components unexpectedly becoming party to a Segmentation Contract, IF-M Segmentation requires support for exclusive message delivery. This means that IF-M Segmentation is only possible on implementations that employ IF-IMV 1.3 or later, IF-IMC 1.3 or later, and IF-TNCCS 2.0 or later.

2.7 IF-M Segmentation Diagram Conventions

This specification defines the syntax of IF-M Segmentation using diagrams. Each diagram depicts the format and size of each field in bits. Implementations MUST send the bits in each diagram as they are shown from left to right for each 32-bit quantity traversing the diagram from top to bottom. Multi-octet fields representing numeric values MUST be sent in network (big endian) byte order.

Descriptions of bit field (e.g. flag) values are described referring to the position of the bit within the field. These bit positions are numbered from the most significant bit through the least significant bit, so a one octet field with only bit 0 set has the value 0x80.

3 IF-M Segmentation Exchanges and Processing

The IF-M Segmentation specification defines the process by which a pair of IF-M Components establish a Segmentation Contract between them to govern the exchange of IF-M messages. IF-M Segmentation also defines the process by which a Base Message might be divided into segments, and those segments transmitted from one IF-M Component to another. This section describes the processes and procedures used to perform these activities.

This section uses the term IF-M Component rather than refer specifically to an IMV or IMC unless the distinction is important. This reflects that IF-M Segmentation allows either an IMV or IMC to be the party that requests message size limits or the use of Segmented Message Exchanges (i.e., the Contracting Party).

3.1 The Segmentation Contract

A Segmentation Contract is an agreement between two IF-M Components to limit the size of messages and/or to employ segmentation to exchange messages. A Segmentation Contract MUST be in place before the use of Segmented Message Exchanges as described in this specification. (Recall, however, that IF-TNCCS imposes its own 4 GB size limits on IF-M messages, and other IF-M bindings can impose message size limits of their own. Messages MUST NOT exceed those size limits regardless of whether a Segmentation Contract is in place.)

A Segmentation Contract exists between a specific IF-M Component on one device, and another IF-M Component on another device with which it exchanges IF-M messages of a given IF-M Subtype. The IF-M Component that requests the Segmentation Contract is called the Contracting Party, while the party that receives and agrees to this request is called the Contracted Party. A Segmentation Contract only constrains messages sent from the Contracted Party to the Contracting Party. Moreover, the Segmentation Contract only applies to messages of a single IF-M Subtype, specified when the Segmentation Contract is established. Thus, one can say that a specific Segmentation Contract is distinguished by three characteristics:

- The identity of the Contracted Party
- The identity of the Contracting Party
- The IF-M Subtype covered by the contract

A Segmentation Contract constrains the maximum size of messages and/or the use of segmentation in the delivery of messages of the specified IF-M Subtype from the Contracted Party to the Contracting Party. The presence of a particular Segmentation Contract does not add constraints on message sizes and/or permit the use of segmentation for any of the following:

- Messages sent by the Contracted Party to IF-M Components other than the Contracting Party.
- Messages sent by the Contracted Party that have an IF-M Subtype other than the IF-M Subtype specified in the Segmentation Contract.
- Messages sent by the Contracting Party to any IF-M Component, including the Contracted Party.
- Messages sent to the Contracting Party by any IF-M Component other than the Contracted Party.

There MUST be no more than one Segmentation Contract with the same triple of Contracting Party, Contracted Party, and IF-M Subtype in effect at any point in time. While there can be many active Segmentation Contracts within a TNC architecture at any given time, each Contract needs to differ in at least one of those three characteristics.

Some IF-M Components might support messages of more than one IF-M Subtype. An IF-M Component that supports IF-M Segmentation for one of those IF-M Subtypes is not required to support IF-M Segmentation for other supported IF-M Subtypes. In other words, an IF-M Component

is allowed to support IF-M Segmentation for a subset of the IF-M Subtypes that it supports. When vendors indicate that a given IMC or IMV conforms to the IF-M Segmentation specification, they MUST identify which IF-M Subtypes supported by their IF-M Component can be managed by IF-M Segmentation.

An IF-M Component conformant with this specification for a given IF-M Subtype MUST support being the Contracted Party in a Segmentation Contract for that Subtype. IF-M Components conformant with this specification for a given IF-M Subtype SHOULD support acting as a Contracting Party for that IF-M Subtype. In other words, an IF-M Component conformant with IF-M Segmentation for a given IF-M Subtype needs to be able to have its own behaviors constrained, but is not required to have the ability to request constraints on others.

An IF-M Component conformant with this specification for a given IF-M Subtype MUST support being party to multiple Segmentation Contracts simultaneously, both as the Contracted Party and (if supported) as the Contracting Party. Upper limits in the number of simultaneously supported Segmentation Contracts SHOULD NOT be hard-coded, and will ideally only be functions of practical considerations on an endpoint, such as available memory. Recall that IMVs in some enterprises might interact with thousands of IMCs. If those IMVs are to support IF-M Segmentation, it might be necessary to efficiently support thousands of simultaneous Segmentation Contracts. Implementers ought to consider the intended operational environment of their products when designing functionality to manage multiple simultaneous Segmentation Contracts to support.

3.2 Establishing a Segmentation Contract

To establish a Segmentation Contract, the IF-M Component that wishes to act as the Contracting Party sends a Segmentation Contract Request attribute to the endpoint with the IF-M Component(s) that it wishes to act as the Contracted Party. The Segmentation Contract Request MAY use exclusive delivery, so that it is only received by a single IF-M Component, but this is not required. (See section 4.6 for an explanation as to why Segmentation Contract Requests are not required to use exclusive delivery.) The Segmentation Contract Request MUST be sent using a method that includes the sender's IMC ID or IMV ID, such as by using the sendMessageLong method as described in section 3.9.5 of the IF-IMV 1.4 specification [5].

There are multiple possible ways a recipient might respond:

1. In the case that the recipient does not recognize the Segmentation Contract Request attribute, it SHOULD respond with a TNC_IFM_ATTRIBUTE_NOT_SUPPORTED IF-M Error attribute.
2. In the case that the recipient does not recognize the Segmentation Contract Request attribute, it MAY silently discard the Segmentation Contract Request attribute as unrecognized. In this case, the sending IF-M Component receives no response.
3. In the case that the recipient conforms to the IF-M Segmentation specification but is unwilling or unable to support the requested Segmentation Contract, it MUST respond with a TNC_IFM_SEG_CONTRACT_REJECTED IF-M Error attribute. This error attribute MAY describe an alternative Segmentation Contract that the recipient would be willing to honor, or MAY simply indicate that the recipient is unwilling to act as a Contracted Party for the given IF-M Subtype at this time. See section 4.12 for more on the TNC_IFM_SEG_CONTRACT_REJECTED IF-M Error attribute.
4. In the case that the recipient conforms to the IF-M Segmentation specification and is willing to be bound by the requested Segmentation Contract, it MUST respond with a Segmentation Contract Response attribute. The Segmentation Contract Response attribute MUST be sent using exclusive delivery, as defined in section 3.3.2.2 of the IF-IMV 1.4 specification [5].

In response scenarios 1 through 3, the attempt to establish the Segmentation Contract fails and does not result in a Segmentation Contract governing the behavior of the request's recipient. Response scenarios 1 and 2 indicate the recipient does not recognize IF-M Segmentation attributes, and thus are an unqualified rejection of any Segmentation Contract. Response scenario 3 could be either a

flat rejection (indicating that the recipient is currently not entertaining requests for Segmentation Contracts for the given IF-M Subtype at this time) or it could include a counter-offer indicating a Segmentation Contract the recipient would be willing to accept. The latter response does not itself establish a Segmentation Contract but allows the would-be Contracting Party to restart the process of establishing a Segmentation Contract with some confidence that the new Segmentation Contract will be acceptable to the recipient.

Only response 4 results in the establishment of a Segmentation Contract, with the sender of the Segmentation Contract Request attribute serving as the Contracting Party, and the sender of the Segmentation Contract Response attribute serving as the Contracted Party. The Segmentation Contract Response attribute describes the actual message and segment size limits of the resulting Segmentation Contract. These limits MUST be less than or equal to the limits identified in the Segmentation Contract Request attribute. (See section 3.7.5 for procedures if this requirement is violated.) See section 4.7 for more details on how/if the Response might provide limits less than those specified in the Request. The important point to remember is that it is the Segmentation Contract Response attribute that provides the actual terms of the Segmentation Contract that binds the Contracted Party.

IF-M Segmentation does not support would-be Contracted Parties initiating establishment of Segmentation Contracts with Contracting Parties. The Contracting Party MUST always be the one that initiates the process of creating a Segmentation Contract.

3.2.1 IF-M Segmentation Attributes and the IF-TNCCS IF-M Subtype

Most IF-M attributes are defined to use a specific IF-M Subtype value, recorded in the IF-TNCCS Message's Vendor ID and Message Type fields. For example, all IF-M attributes that are defined as part of the SWID Message and Attributes for IF-M binding [7] are sent with a Vendor ID value of 0x005597 (the Trusted Computing Group's IANA-registered Private Enterprise Number) and a Message Type of 0x00000003. Similarly, all attributes associated with the PTS Protocol Binding to TNC IF-M [6] have a Vendor ID value of 0x005597 and a Message Type of 0x00000001.

Attributes defined in the IF-M Segmentation specification are different in that they have no fixed IF-M Subtype. The Segmentation Contract Request and Segmentation Contract Response attributes are sent using the IF-M Subtype that is the subject of the Segmentation Contract being created. A Segmentation Contract Response MUST use the same IF-M Subtype as the Segmentation Contract Request to which it is responding. Other attributes defined in this specification use the IF-M Subtype of the Segmentation Contract they are modifying, cancelling, or servicing. This allows IF-M Segmentation attributes to be sent to IF-M Components registered to any IF-M Subtypes. In this aspect, the IF-M Segmentation attributes are similar to the IF-M Error attribute, which also has no fixed IF-M Subtype and is sent using the IF-M Subtype of the binding whose error it is reporting.

3.2.2 Limits on Segmentation Contract Sizes

In order to ensure that IF-M Segmentation attributes can always be delivered under any Segmentation Contract, an IF-M Component MUST NOT send a Segmentation Contract Request that specifies either a maximum message size or a maximum segment size that is less than 64 bytes. (A limit of 64 bytes allows an individual attribute of up to 44 bytes plus the 20 byte fixed IF-M message and attribute header fields.) A component that receives such a request MUST reject it using the TNC_IFM_INVALID_PARAMETER error code, as defined in section 5.2.13 of the IF-M TLV Binding specification [3]. Likewise, an IF-M Component MUST NOT send a Segmentation Contract Response that specifies either a maximum message size or a maximum segment size that is less than 64 bytes, and a component that receives such a request MUST reject it using TNC_IFM_INVALID_PARAMETER. In the latter case, both parties MUST automatically treat the Segmentation Contract that would normally have been established by the Segmentation Contract Response as being cancelled.

IF-M Component implementations conforming to this specification SHOULD allow local administrators to set their own lower limits on message and segment sizes, provided those limits are greater than or equal to 64 bytes. Local administrators ought to be aware that smaller message and

segment sizes, while limiting the size of individual IF-M messages, will ultimately increase the total amount of network traffic due to the increased overhead imposed by handling a larger number of messages in the Segmented Message Exchange. Administrators are encouraged to find the appropriate balance between managing individual message sizes (by applying upper size limits using Segmentation Contracts), and limiting the overhead created by the IF-M Segmentation procedures (by setting lower limits for Segmentation Contract sizes that are requested or accepted).

3.2.3 Allowing Segmented Message Exchanges

Instead of setting a maximum segment size, a Segmentation Contract can prohibit the use of Segmented Message Exchanges regardless of the message size. This is done by using the reserved value of 0xFFFFFFFF as the maximum segment size in a Segmentation Contract Request. (Since all messages need to be smaller than this value, due to size limits set in the IF-M TLV Binding and IF-TNCCS TLV Binding specifications, a Segmentation Contract with such a value would never use Segmented Message Exchanges anyway.)

In the case that a Segmentation Contract Request prohibits the use of Segmented Message Exchanges, the Segmentation Contract Response MUST also prohibit the use of Segmented Message Exchanges by using the same reserved value of 0xFFFFFFFF in the maximum segment size field. Likewise, in the case that a Segmentation Contract Request allows Segmented Message Exchanges (by providing a maximum segment size other than 0xFFFFFFFF), a resulting Segmentation Contract Response MUST also allow Segmentation Message Exchanges. In particular, the maximum segment size given in the Response MUST be less than or equal to the maximum segment size given in the Request. In other words, both the Segmentation Contract Request and the Segmentation Contract Response need to match as to whether Segmented Message Exchanges are permitted. See section 3.7.5 for procedures if these requirements are violated.

3.3 Modifying a Segmentation Contract

The Contracting Party MAY request modification of an existing Segmentation Contract at any time while that Contract remains in force. This is done by sending a Segmentation Contract Request attribute to the Contracted Party of an existing Segmentation Contract for the same IF-M Subtype covered by that Contract. The recipient of this request (a.k.a., the Contracted Party of the Segmentation Contract the Contracting Party wishes to modify) responds in one of two ways:

1. In the case that the recipient is unwilling or unable to support the modified Segmentation Contract, it MUST respond with a TNC_IFM_SEG_CONTRACT_REJECTED IF-M Error attribute. In this case, both parties MUST immediately treat the existing Segmentation Contract that the Contracting Party was attempting to modify as cancelled. (Other existing Segmentation Contracts remain unaffected.)
2. In the case that the recipient is willing to be bound by the modified Segmentation Contract, it MUST respond with a Segmentation Contract Response attribute. The Segmentation Contract Response attribute MUST be sent using exclusive delivery, as defined in section 3.3.2.2 of the IF-IMV 1.4 specification [5].

Note that an attempt to modify an existing Segmentation Contract is an “all-or-nothing” affair. Either the Contracted Party agrees to the modified Contract, which becomes the new Segmentation Contract binding the Contracted and Contracting Parties for the given IF-M Subtype, or the old contract is cancelled, leaving no Segmentation Contract in force for the given Contracting and Contracted Parties for the given IF-M Subtype.

One could view both the Segmentation Contract creation and modification processes as effectively the same:

1. An IF-M Component wants to bind another Component to a (new or modified) Segmentation Contract. It sends the other Component a Segmentation Contract Request attribute.

2. The recipient of this Request determines whether it is willing to be bound by the given Contract (assuming it supports IF-M Segmentation and thus can be bound by a Contract in the first place).
    a. If it agrees, it sends a Segmentation Contract Response attribute. This Response describes the Segmentation Contract to which the recipient (a.k.a., Contracted Party) will adhere. This Contract replaces the existing Segmentation Contract if one exists.
    b. If it declines, it sends a TNC_IFM_SEG_CONTRACT_REJECTED IF-M Error attribute. There is now no Segmentation Contract for which the sender of the Segmented Contract Response attribute is the Contracting Party and the recipient is the Contracted Party for the given IF-M Subtype.

Note that IF-M Segmentation does not provide a way for a Contracted Party to request modification of an existing Segmentation Contract.

3.4 Terminating a Segmentation Contract

Once established, a Segmentation Contract remains in force until one of the following events occurs:

1. The Contracting Party requests modification of the Segmentation Contract, as described in section 3.3. (Note regardless of whether the request is accepted by the Contracted Party, the existing Segmentation Contract is cancelled.)
2. The Contracted Party cancels the contract using a Cancel attribute.
3. The Contracted Party send certain IF-M Errors associated with that Segmentation Contract.

As described in section 3.3, the Contracting Party MAY request modification of an existing Segmentation Contract. This either results in the replacement of the old Contract with a new Contract (if the Contracted Party accepts the request), or the cancellation of the old Contract (if the Contracted Party rejects the request). In either case, the old Contract ceases to be in force. If the Contracting Party wishes to cancel an existing Segmentation Contract but does not want some new contract to take its place, it can do this by requesting modification using a Null Contract as the new contract. A Null Contract imposes no size limits and prohibits Segmented Message Exchanges, and thus is treated by both the Contracting and Contracted Parties as being equivalent to no Segmentation Contract being in force.

The Contracted Party can also cancel an existing Segmentation Contract by sending a Cancel attribute to the Contracting Party. (See section 4.10 for more on the Cancel attribute.) After the Cancel attribute is sent, the Segmentation Contract between the Contracted Party and Contracting Party for the given IF-M Subtype MUST be treated as cancelled by both parties. A Contracting Party cannot reject a call to cancel a Segmentation Contract by the Contracted Party, although it MAY attempt to establish a new contract afterwards using the procedures described in section 3.2.

Finally, some error conditions automatically trigger cancellation of the Segmentation Contract associated with those errors. See section 3.7 for more information about error conditions.

3.5 Applying a Segmentation Contract

Once a Segmentation Contract is in place and active, the Contracted Party MUST abide by its terms until the contract is changed or cancelled. This entails a number of specific obligations on the Contracted Party. By contrast, a Segmentation Contract imposes no obligations on its Contracting Party. This section describes each of the Contracted Party's obligations in detail.

The following discussions use the term Contract Constrained Message. A Contract Constrained Message is a message constrained by a Segmentation Contract. In particular, for a given Segmentation Contract, a Contract Constrained Message is a message sent from the Segmentation Contract's Contracted Party, to that contract's Contracting Party, and which has the IF-M Subtype associated with that Segmentation Contract. (Recall that while an IF-M message can contain multiple IF-M attributes, all attributes within a single IF-M message are required by the IF-M TLV Binding

specification to have the same IF-M Subtype, and thus one can speak of an IF-M Message having a single IF-M Subtype.)

When measuring the size of a Contract Constrained Message, the message's IF-M header is included. The IF-M header is considered part of the associated Base Message.

3.5.1 Sending Messages

In the case that a Contracted Party generates a Contract Constrained Message whose size is less than or equal to both the maximum message size and maximum segment size allowed by the Segmentation Contract, the message MUST be sent without using IF-M Segmentation attributes. In other words, if a message is smaller than both of the Segmentation Contract's size limits, no further action is necessary before the Contracted Party sends the message.

In the case that a Contracted Party generates a Contract Constrained Message that exceeds the maximum message size allowed by the Segmentation Contract, the following procedure MUST be followed. This is the case regardless of whether the message is generated in direct response to a request, or if the message is generated spontaneously by the Contracted Party, such as due to the detection of a local change in state that the endpoint is required to report.

In the case that the oversized message contains multiple attributes, the Contracted Party MUST divide those attributes among multiple messages. Attributes MUST remain be complete (it is not permissible to place part of one attribute in one message and another part of the same attribute in another message), but otherwise IF-M Segmentation imposes no constraints on how the attributes are divided among messages (although some IF-M bindings might include such constraints). This process MUST continue until either all messages are less than or equal to the maximum message size given in the contract, or the only messages that exceed the maximum message size contain only a single attribute.

After this redistribution, sometimes there will be an individual attribute that, when placed alone in an IF-M message, causes that message to exceed the Segmentation Contract's size limits. Such an attribute is referred to as an "Oversized Attribute". A Contracted Party MUST use one of the following three methods to deal with Oversized Attributes:

1. The Contracted Party MAY turn the Oversized Attribute into multiple smaller attributes using mechanisms directly supported by the IF-M Subtype definition. This is not the same as segmenting the attribute and does not employ IF-M Segmentation. This is simply a repackaging of the original attribute in a manner consistent with the attribute's underlying IF-M Subtype binding specification so that the same information is sent via multiple attributes. Not all IF-M Subtypes support this action. Once the Oversized Attribute has been replaced by these smaller attributes, the Contracted Party can redistribute the smaller attributes among multiple messages and then restart the message sending process for each of these new messages.
2. The Contracted Party MAY remove information from the Oversized Attribute until it fits within the contract's size limits. This MUST only be done in the case that doing so does not violate either the relevant IF-M Subtype specification or the sender's responsibilities to the recipient. For example, the sender might need to fulfill a subscription where the subscription requires that the relevant information be sent using a verbose format. In such a case, the sender could not reduce the Oversized Attribute size by switching to a more concise format, since this would violate the terms of the subscription.
3. The Contracted Party MAY send an Oversized Message attribute instead of the message containing the Oversized Attribute. The Oversized Message attribute includes the IF-M message and attribute headers of the Oversized Attribute, but the rest of Oversized Attribute is not sent. The Oversized Message attribute tells the recipient that the Contracted Party had information to share, but that it was prevented from doing so by the limits imposed by the Segmentation Contract. The Contracting Party has the option to request delivery of that message and attribute by granting that particular message an exemption from the Segmentation Contract. For this reason, the Contracted Party MUST retain the message

containing the Oversized Attribute for at least 300 seconds (5 minutes) or until the Contracting Party successfully retrieves the message (whichever comes first), after which the message MAY be discarded. See section 3.5.1.1 for more information on retrieving Oversized Attributes. The Oversized Message attribute is discussed in more detail in section 4.11. Because the message containing the Oversized Attribute might be delivered if the Contracting Party grants that message an exemption from the Segmentation Contract, the Message ID of that message MUST NOT be re-assigned to some other IF-M message until after the message has been discarded.

Figure 2 diagrams the procedure described above.

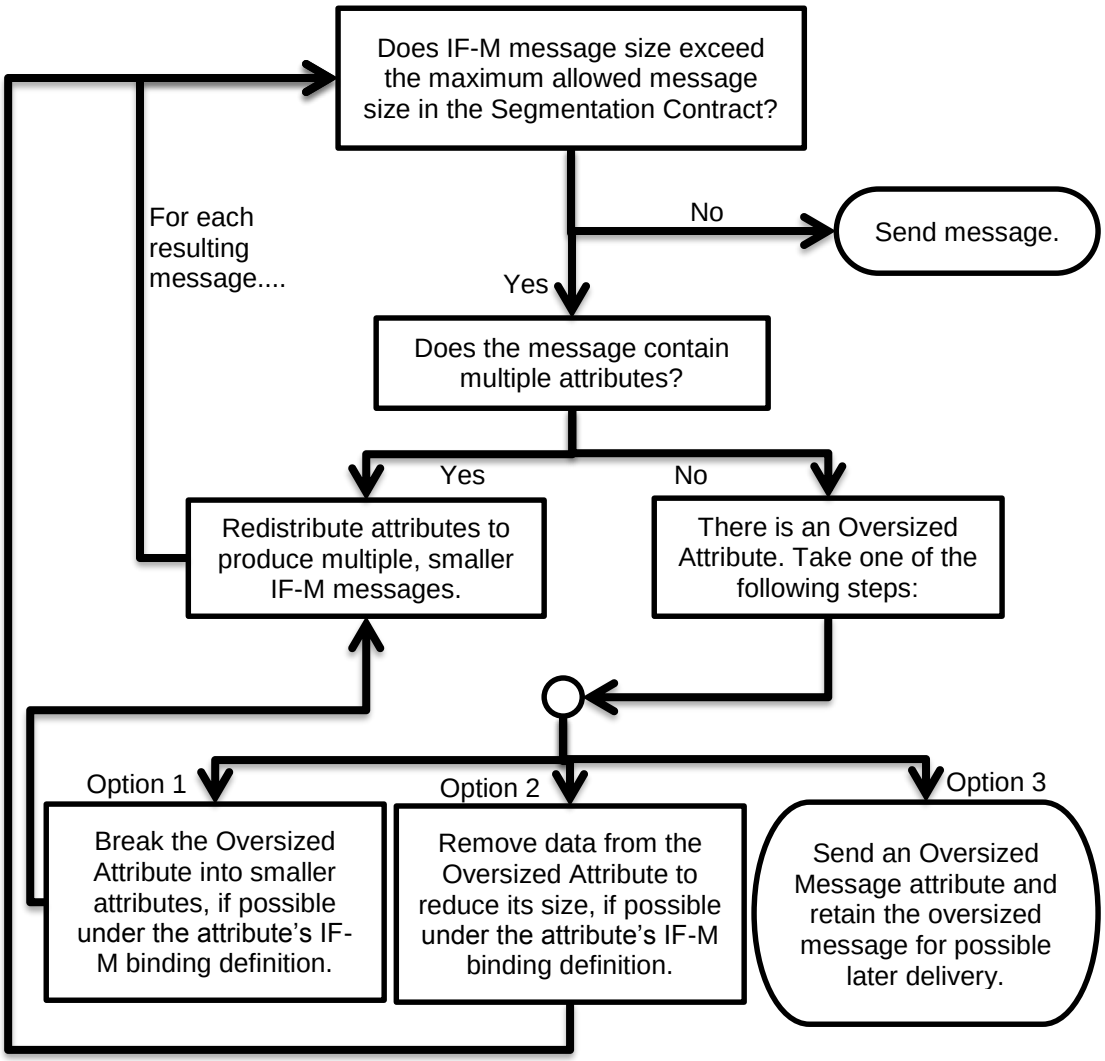

Figure 2 - Processing of Oversized Messages

In general, Contracted Parties SHOULD employ option 1 or 2 if at all possible. These options ensure the Contracting Party receives the relevant information from the Oversized Attribute in a manner that complies with the Segmentation Contract. As noted, these options can only be employed if doing so does not violate the relevant IF-M Subtype specification or the Contracted Party's specific obligations to the Contracting Party. In the case that options 1 and 2 are not possible, the endpoint MUST resort to option 3. Implementers MAY allow enterprise admins to specify circumstances under which each option is used.

IF-M Components are not required to support all three of the options described. However, IF-M Components MUST always at least support option 3, since that is the only option that is always possible, and SHOULD support options 1 and 2 for flexibility in the case that they are possible under the relevant IF-M binding. IF-M Components are not required to use the same option each time they have an Oversized Attribute; they are permitted to select from the available options on an instance-by-instance basis.

The Contracted Party SHOULD log any oversized messages it generates and which action is employed to deal with an oversized message. The Contracting Party SHOULD log receipt of an Oversized Message attribute. The Contracting Party will not necessarily know that there was an oversized message if the Contracted Party employs option 1 or option 2.

Note that, even if a message is less than or equal to the maximum message size allowed by the Segmentation Contract, it might still be larger than the permitted maximum segment size. In this case, the Contracted Party MUST follow the steps described in section 3.5.2. This is true regardless of whether or not the message underwent the above process prior to it becoming less than or equal to the maximum message size.

3.5.1.1 Retrieving Oversized Messages

As described in section 3.5.1, a Contracted Party sends an Oversized Message attribute if it has a Contract Constrained Message that exceeds the maximum message size allowed by a Segmentation Contract and which has not been sufficiently reduced in size by other steps outlined in the described procedure. An Oversized Message attribute alerts the Contracting Party of the presence of the message, as well as its IF-M Subtype, the attribute type, and the total size of the oversized message. (The latter information is captured in the message and attribute headers that are contained in the Oversized Message attribute.) The Contracting Party then has the option of ignoring this oversized message (in which case it will never receive this message) or of granting a Contract Exemption for the given message. The Contracting Party signals the latter course by sending the Contracted Party a Contract Exemption attribute that contains the Message Identification of the oversized message, as captured in the IF-M message header sent in the Oversized Message attribute. The Contract Exemption attribute includes a flag to indicate whether the maximum segment size given in the Segmentation Contract ought to be honored, or if the oversized message is to be sent without segmentation regardless of the Segmentation Contract limits.

Upon receiving a Contract Exemption attribute, the Contracted Party MUST do one of the following:

1. Send the identified message to the Contracting Party.
    a. In the case that the Segmentation Contract allows Segmented Message Exchanges, the Contract Exemption attribute indicates that the maximum segment size given in the Segmentation Contract is to be honored, and the message is larger than the maximum segment size given in the Segmentation Contract, the Contracting Party MUST deliver the message using a Segmented Message Exchange.
    b. Otherwise, the Contracting Party MUST deliver the message in its entirety without using a Segmented Message Exchange.
2. In the case that the Contracted Party has no record of the IF-M message identified in the Contract Exemption attribute, the Contracted Party MUST send a TNC_IFM_MESSAGE_NOT_FOUND IF-M Error attribute to the Contracting Party. This IF-M Error attribute is described in section 4.13.2.3.

Note that the Oversized Message attribute is unable to report messages that exceed the 4 GB size limit imposed on all IF-M messages by the IF-M TLV Binding specification. Likewise, it is possible that a message might exceed size limits defined within a specific IF-M binding. Messages that exceed either such size limit MUST NOT result in an Oversized Message attribute but instead MUST employ the procedure dictated by their IF-M binding definition for dealing with such a situation. In other words, an Oversized Message attribute is not to be used to circumvent size limits imposed by other IF-M bindings.

3.5.2 Segmented Message Exchanges

In the case that the size of a Contract Constrained Message is less than or equal to the maximum message size limit of the relevant Segmentation Contract, but is larger than the Segmentation Contract's maximum segment size, the message is a candidate for a Segmented Message Exchange. Such a message is called a Segmentation Candidate. In the case that a Contract Constrained Message is less than or equal to both the maximum message size limit and the maximum segment size limit of the relevant Segmentation Contract, it MUST be sent as a complete Base Message.

A Segmentation Candidate first goes through a process similar to that of an oversized message. In the case that the Segmentation Candidate contains multiple attributes, the Contracted Party MUST divide those attributes among multiple messages. Attributes MUST remain complete, but otherwise IF-M Segmentation imposes no constraints on how the attributes are divided among messages (although some IF-M bindings might include such constraints). At the end of this process it MUST be the case that either all messages are under the maximum segment size given in the Segmentation Contract, or the only messages that exceed the maximum segment size limit each contain only a single attribute.

After this redistribution, sometimes there will be an individual attribute that, when placed alone in an IF-M message, causes that message to exceed the Segmentation Contract's segment size limits. The Contracted Party MAY attempt to reduce the size of this attribute or divide this attribute into smaller attributes, providing that these actions are permitted by the attribute's IF-M binding. These steps are identical to methods 1 and 2 for attribute size reduction described in section 3.5.1. In the case that these methods are not employed, or are not able to reduce the message size below the Segmentation Contract's maximum segment size limit, the Contracted Party MUST employ a Segmented Message Exchange to deliver the oversized Base Message.

Segmented Message Exchanges are used to send a Base Message (that is, an IF-M message, including its IF-M message header, IF-M attribute header, and IF-M attribute) from a Contracted Party to a Contracting Party as a series of Message Segments rather than all at once. In a Segmented Message Exchange, the Contracted Party first divides the Base Message into Message Segments. Each Message Segment MUST be less than or equal to the maximum segment size limit given in the Segmentation Contract minus 20 bytes. (Subtracting 20 from the maximum segment size limit is necessary to ensure that the Message Segment, the Segment Envelope attribute that contains it, the Segment Envelope IF-M attribute header, and the IF-M message header have a combined size that is less than or equal to the Segmentation Contract's maximum segment size.) In other words, the resulting Segment Envelope complete with its attribute and message header MUST be less than or equal to the Segmentation Contract's maximum segment size limit, and thus MUST NOT itself be subject to segmentation.

All Message Segments MUST be 1 byte or larger. In the case that a Contracting Party receives a Message Segment attribute with a 0-byte segment, it MUST respond with a TNC_IFM_SEG_VIOLATION IF-M Error attribute. Message Segments SHOULD be as close to the Segmentation Contract's maximum segment size limit as possible, in order to minimize the total number of Message Segments and thus the total number of exchanges necessary to deliver the whole Base Message. Message Segments do not need to be equal in size.

Once the Base Message has been divided into Message Segments, the first Message Segment is sent from the Contracted Party to the Contracting Party using a Segment Envelope attribute, described in section 4.8. The Contracting Party receives this Message Segment and sometime later (possibly immediately, but possibly after other processing occurs) it sends the Contracted Party a Next Segment attribute (described in section 4.9) requesting the next Message Segment. The Contracted Party then sends the second Message Segment of the Base Message. This process continues until the Contracted Party sends the final Message Segment of the Base Message, at which point the Contracting Party has received the complete Base Message (as evidenced by receipt of a Segment Envelope attribute with its More flag not set - see section 4.8). Figure 3 shows this exchange.

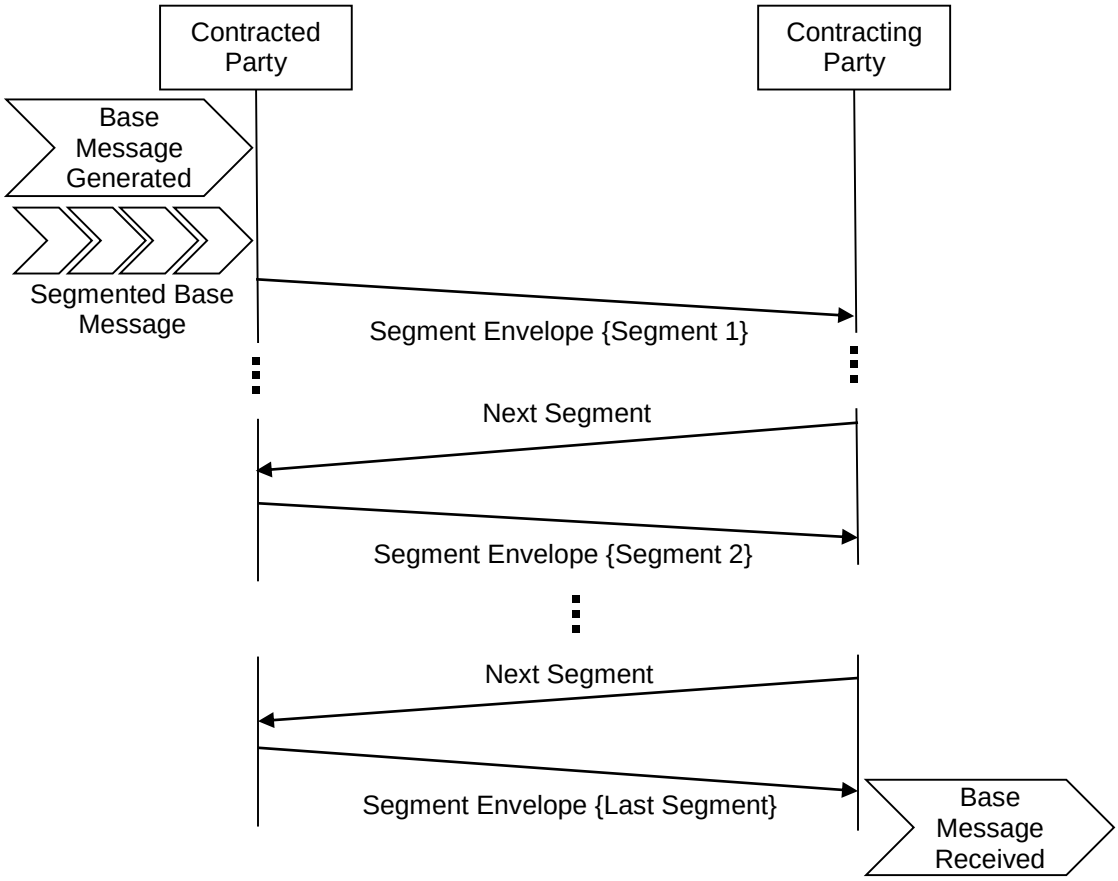

Figure 3 - Segmented Message Exchange

The Contracted Party MUST NOT send more than a single Message Segment of a given Base Message at a time. (A single IF-TNCCS batch might contain multiple Message Segments, but each Message Segment needs to be from a different Base Message.) Likewise the Contracting Party MUST only send a Next Segment attribute after it receives a Message Segment, and can only send a single Next Segment attribute for any given Base Message at a time. In other words, for a given Base Message, the Contracted and Contracting Party engage in a half-duplex exchange, alternating sending messages to each other until the final Message Segment is delivered.

The Contracted Party MUST deliver Message Segments of a given Base Message in order. Specifically, the first Message Segment delivered MUST be the earliest bytes of the Base Message, including the Base Message's IF-M message header. Each subsequent Message Segment delivered MUST represent the next consecutive bytes of the Base Message. The Contracted Party MUST NOT add any padding or other material to each Message Segment. Thus, the Base Message can be reconstructed by the Contracting Party simply by concatenating the received Message Segments in the order in which they are received.

Both Segment Envelope and Next Segment attributes MUST be sent using exclusive delivery, as defined in section 3.3.2.2 of the IF-IMV 1.4 specification [5].

As described in the above exchange, when a Contracting Party receives a Message Segment, it can then request the delivery of the next Message Segment. It MAY do this immediately, but it MAY also delay this request. There are several reasons why it might take the latter course. For example, it might delay requesting the next Message Segment for resource management reasons, to give other endpoints more opportunity to use the network, or to allow other local applications to perform activities before the next Message Segment is delivered. The Contracting Party might also wait until the receiving IF-M Component has performed as much processing as possible on the Message Segment

it has received. This IF-M Component might even conclude that it has no need for the remaining segments of the Base Message and forego requesting any further Message Segments, although if it reaches this conclusion it SHOULD formally cancel the Segmented Message Exchange, as described in section 3.5.2.1.

If the Contracting Party delays requesting the next Message Segment for too long, a Contracted Party MAY declare the Segmented Message Exchange stalled and cancel the exchange by sending a Cancel attribute. This specification does not dictate when a Contracted Party ought to declare a Segmented Message Exchange to be stalled, but it SHOULD wait at least 300 seconds (5 minutes) after having sent its most recent Segment Envelope attribute before doing so.

By contrast, the Contracted Party MUST provide the next Message Segment in a Segmented Message Exchange as soon as possible following the Contracting Party's Next Segment attribute.

IF-M Components MAY conform to the IF-M Segmentation specification without supporting Segmented Message Exchanges. An IF-M Component that supports acting as a Contracted Party but which does not support Segmented Message Exchanges MUST reject any Segmentation Contract Request that allows Segmented Message Exchanges. An IF-M Component that supports acting as a Contracting Party but which does not support Segmented Message Exchanges MUST NOT send Segmentation Contract Request attributes for contracts that allow Segmented Message Exchanges.

IF-M Components that support Segmented Message Exchanges MUST support multiple simultaneous Segmented Message Exchanges under a single Segmentation Contract. This can happen if the Contracted Party generates multiple Contract Constrained Messages that require segmentation under that Segmentation Contract. Each Segmented Message Exchange would deliver the segments of a different Base message; these Segmented Message Exchanges can be distinguished by an identification number assigned by the Contracted Party. This is described in more detail in section 4.8.

3.5.2.1 Cancelling Segmented Message Exchanges

IF-M Components that support Segmented Message Exchanges MUST support the ability to cancel an ongoing Segmented Message Exchange, and to accept the cancelling of this exchange by the other party. A Contracting Party cancels an ongoing Segmented Message Exchange by setting a special flag in the Next Segment attribute it sends to the Contracted Party. (See section 4.9 for more on the Next Segment attribute.) A Contracted Party cancels an exchange using the Cancel attribute. (See section 4.10 for more on the Cancel attribute.) In both cases, cancellation of a Segmented Message Exchange does not necessarily cancel the Segmentation Contract that led to this exchange. (In the case of a Contracted Party sending a Cancel attribute, it is possible to include in a single Cancel attribute a combination of flags that cancels both a Segmentation Contract and an ongoing Segmented Message Exchange. This is not possible for the Contracting Party, since it uses different attributes to cancel an exchange and to cancel a Segmentation Contract.) Either party MAY cancel an ongoing Segmented Message Exchange for any reason. For example, a Contracting Party might cancel an exchange because it has received all the information it needed in earlier Message Segments and does not need the rest of the Base Message. A Contracted Party might cancel an exchange because it has been several minutes since the Contracting Party requested the next Message Segment and the Contracted Party is declaring the exchange to be stalled. Either party might cancel the exchange due to changes in their environment that render further delivery of the Base Message problematic or obsolete.

As noted in section 3.4, either a Contracting Party or a Contracted Party has the ability to terminate an active Segmentation Contract at any time. In the case that a Segmentation Contract is cancelled while there is an ongoing Segmented Message Exchange in support of that Contract, the Segmented Message Exchange MUST continue to completion as if the Segmentation Contract were still in place (i.e., with the same segment size limits) unless some party explicitly cancels that Segmented Message Exchange as well. In other words, cancellation of Segmentation Contracts and cancellation of Segmented Message Exchanges are completely separable activities, and one does not automatically imply the other.

3.5.3 Violations of Segmentation Contracts

While a Segmentation Contract is in place and active, the Contracted Party MUST NOT knowingly violate the contract. (Since the Segmentation Contract does not restrict the activities of the Contracting Party, the Contracting Party cannot violate the contract.) It is conceivable that the Contracted Party might lose track of a Segmentation Contract that binds it, e.g. due to a crash or data corruption. If this happens the Contracted Party might send a message that exceeds the size limits imposed by the Segmentation Contract. This could come in the form of a message exceeding the maximum message size limit, or by an unsegmented message exceeding the maximum segment size limit. The Contracting Party MUST be capable of receiving and correctly processing messages that violate the active Segmentation Contract's size limits. This is necessary to keep IF-M Segmentation from becoming a point of failure for other IF-M bindings - a message exchange that does not use IF-M Segmentation attributes ought not to fail only because the recipient thought that IF-M Segmentation was supposed to be employed. Moreover, receipt of a message that exceeds either limit in the Segmentation Contract MUST NOT result in an IF-M Segmentation IF-M Error attribute to the Contracted Party (although other errors in the received message might result in an IF-M Error for other reasons). In other words, the violating message MUST be received and processed by the appropriate IF-M Component as if there was no Segmentation Contract in place. In the case that the Contracting Party receives a message that exceeds either of the size limits of a Segmentation Contract, it MUST treat the violated Segmentation Contract as cancelled. The violation of the Segmentation Contract indicates that either the Contracted Party is non-conformant with IF-M Segmentation, or that either the Contracted or Contracting Party are no longer in alignment as to the requirements of the contract (possibly reflecting state corruption). In either case, the Contracting Party can no longer rely on its understanding of the Segmentation Contract to be enforced, and thus needs to treat it as inoperable (i.e., cancelled). The (former) Contracting Party MAY wish to establish a new Segmentation Contract with the message sender if this happens.

In the case that a Contracting Party receives a Message Segment that is larger than permitted by the current Segmentation Contract, or if the recipient of a Segment Envelope or Next Segment attribute does not have any record of a Segmentation Contract that permits Segmented Message Exchanges, then the recipient of the Message Segment MUST respond with a TNC_IFM_SEG_VIOLATION IF-M Error attribute. (See section 4.13.2.4 for more on this error attribute.) Both the error sender and recipient MUST treat this error as cancelling the Segmented Message Exchange and also cancelling any Segmentation Contract that would have been associated with the violating attribute. Other Segmentation Contracts and ongoing Segmented Message Exchanges between the relevant parties, including other Segmented Message Exchanges under the cancelled Segmentation Contract, are unaffected by this error. All Contracting Parties MUST be capable of detecting the receipt of an unexpected segment and responding with the TNC_IFM_SEG_VIOLATION IF-M Error attribute even if that Contracting Party does not support Segmented Message Exchanges.

Note that there is an unequal response to these two situations. If a complete message is received that exceeds one or more of a Segmentation Contract's size limits, either by being larger than the maximum permitted message size or by exceeding the maximum segment size (in which case compliance with the contract required that it be sent via a Segmented Message Exchange), the Contracting Party silently (without sending an IF-M Error) and unilaterally treats the Segmentation Contract as cancelled. In contrast, if a Segment Envelope attribute is received that violates a party's understanding of the Segmentation Contract, either by containing a too-large segment, or by occurring when the party believes Segmented Message Exchanges are prohibited, the party responds with an IF-M Error that explicitly terminates that Segmented Message Exchange as well as the associated Segmentation Contract. The reason for the differing responses is that they represent different types of problems. In the former case, an oversized message indicates that the Contracted Party has forgotten about the Segmentation Contract and is attempting to exchange messages using regular IF-M exchanges. The message itself is otherwise valid and correct. On the other hand, if a bad segment is received, the sending party believes that there is a Segmentation Contract, but that it differs from the receiving party's understanding of this contract. Under this circumstance it is necessary to resync the two parties, which necessitates cancelling the relevant Segmentation Contract.

In any of these cases, all parties who become aware of the error SHOULD log the error to aid in addressing the source of the problem.

3.6 Multi-Component Parties

It is possible to register multiple IF-M Components on a single device (Endpoint or PDP) to receive messages of a given IF-M Subtype. This type of situation is referred to as having a **Multi-Component Party** for the given IF-M Subtype and requires some extra clarification. When this happens, a message of that IF-M Subtype sent to that device would be received by both of these IF-M Components unless the message is sent using exclusive delivery, as described in Section 3.3.2.2 of the IF-IMV 1.4 specification [5].)

Without exclusive delivery of IF-M Segmentation attributes, many problematic situations could arise, since IF-M Components could become subject to a Segmentation Contract without their knowledge or consent. Consider a case where there are two IMVs on a PDP both registered to receive the same IF-M Subtype, where IMV 1 supports IF-M Segmentation and IMV 2 does not. Suppose IMV 1 seeks to establish a Segmentation Contract with some endpoint's IMC. That IMC accepts the contract, sending a Segmentation Contract Response attribute. If the IMC later starts a Segmented Message Exchange but does not use exclusive delivery, IMV 2 would receive Segment Envelopes that it cannot understand or process.

To avoid circumstances like this, all IF-M Segmentation attributes except the Segmentation Contract Request attribute MUST be sent using exclusive delivery. (Segmentation Contract Request attributes MAY be sent using exclusive delivery but are not required to be sent this way. For more on why this is allowed, see section 4.6) Exclusive delivery ensures that only a single IF-M Component receives a given message, regardless of the number of IF-M Components registered to receive that message's IF-M Subtype. By requiring exclusive delivery of Segmentation Contract Responses and most other attributes, Segmentation Contracts and attributes sent in fulfillment of those contracts are only exchanged between one specific pair of IF-M Components. The requirement to use exclusive delivery for IF-M Segmentation attributes other than Segmentation Contract Request attribute avoids the danger of implicit binding to a Segmentation Contract. Note that, because exclusive delivery is communicated through information in the PB-PA message type (as defined in RFC 5793 [11]) which encapsulates the IF-M message body, an IF-M message is either exclusively delivered or not. This means all attributes in that message share the same treatment regarding exclusive delivery. For this reason, while attributes' senders MAY send multiple attributes within an IF-M message that contains IF-M Segmentation attributes, the sender MUST ensure that these attributes not only share the same IF-M Subtype, but also share the same treatment with regard to exclusive delivery and their receiving IF-M Component.

Despite the use of exclusive delivery, however, some special circumstances can arise in the presence of Multi-Component Parties. Consider a situation where an endpoint has two IF-M Components (C1 and C2) registered to receive messages of a given IF-M Subtype. Some other party seeks to establish a Segmentation Contract for this IF-M Subtype with the endpoint and sends it a Segmentation Contract Request attribute without using exclusive delivery. Both IF-M Components on the endpoint receive the Segmentation Contract Request but they respond differently. Component C1 accepts the suggested Segmentation Contract while component C2 rejects the initial contract with a TNC_IFM_SEG_CONTRACT_REJECTED IF-M Error attribute suggesting alternate size limits. In this case, the Contracting Party receives two responses, one accepting and one rejecting, to its single Segmentation Contract Request. Would-be Contracting Parties MUST recognize when this situation is caused by multiple components on the receiving endpoint and not treat this as an error condition. (The Contracting Party will be able to distinguish between senders by comparing the sender's IMV ID or IMC ID value, as appropriate, as this will uniquely identify the sender of each message.)

The Contracting Party has four different ways it can respond to this situation:

1. The Contracting Party MAY choose to send a new Segmentation Contract Request to the endpoint without using exclusive delivery. This would replace the Segmentation Contract that was successfully established on component C1 and would offer a new, hopefully acceptable contract on component C2. Assuming both components accept the new contract, this would

mean both components on the endpoint were bound to the same size limits for the given IF-M Subtype. Implementers need to be aware that it is possible that there might be no mutually acceptable Segmentation Contract for both components C1 and C2 and avoid creating an infinite loop where each proposed contract is alternately rejected by one component or the other.

2. The Contracting Party MAY choose to use exclusive delivery to send component C2 a new Segmentation Contract Request specifying limits acceptable to C2. Component C1 would never receive this Segmentation Contract Request and its existing Segmentation Contract would be unaffected. Assuming component C2 accepts the new contract, this means both components on the endpoint associated with the given IF-M Subtype are bound by Segmentation Contracts with the same Contracting Party, but those Segmentation Contracts impose different size limits. In this case, the Contracting Party MUST ensure that communications from each of those parties is only judged against the appropriate party's Segmentation Contract. For example, a message from component C1 cannot be treated as a violation of component C2's Segmentation Contract or vice versa.
3. The Contracting Party MAY choose to take no action. In this case, component C1 remains bound by the established Segmentation Contract while component C2 is bound by no Segmentation Contract. As before, the Contracting Party MUST ensure that each component's messages are only judged against the relevant contract (or non-contract, as the case may be).
4. The Contracting Party MAY choose to cancel the Segmentation Contract with component C1. In this case, no Segmentation Contract constrains the endpoint's use of the relevant IF-M Subtype.

In a slight variation of this scenario, an endpoint might have two IF-M Components bound to the same IF-M Subtype where one supports IF-M Segmentation and the other does not. In this case, the non-conformant IF-M Component might respond to the Segmentation Contract Request with a TNC_IFM_ATTRIBUTE_NOT_SUPPORTED IF-M Error attribute, or it might silently ignore the Segmentation Contract Request. In the former case, the would-be Contracting Party MUST recognize that the receipt of multiple responses to their Segmentation Contract request reflects a Multi-Component Party situation and not treat this as an error condition. In the latter case, where the non-conformant component silently ignores the request, the Contracting Party might not realize it is in a Multi-Component Party situation until it starts receiving messages from the non-conformant party. As before, the Contracting Party MUST ensure that each component's messages are only judged against the appropriate Segmentation Contract, if one is present.

Note that IF-M Components that are not bound by a Segmentation Contract might send messages using methods that do not include the sender's IMC ID or IMV ID. In a worst-case scenario, this can make it impossible to distinguish between a situation where a Contracted Party has forgotten its contract, and a Multi-Component Party situation where messages are sent from an IF-M Component other than the Contracted Party. In the case that a Contracting Party starts receiving a large amount of traffic with no IMV ID or IMC ID to identify the sender, indicating the possibility that the Contracted Party has forgotten its Segmentation Contract, it MAY send a Segmentation Contract Request to re-establish the Segmentation Contract it believes is already in place. If the Contracted Party had not forgotten its contract, this will simply replace the old contract with an identical contract and produce no change in behavior. If the Contracted Party had lost its Segmentation Contract, this would re-establish the contract.

The examples above describe Multi-Component Party scenarios where a single endpoint has two IF-M Components bound to the same IF-M Subtype. However, there is no limit on the number of IF-M Components a single endpoint might have that are registered to receive messages of the same subtype. For this reason, implementers MUST NOT limit their components' abilities to properly handle Multi-Component Parties to just recognizing and handling two parties.

3.7 Error Handling

The IF-M Segmentation specification defines several new IF-M Error attributes:

- TNC_IFM_SEG_CONTRACT_REJECTED - Sent to indicate rejection of a proposed Segmentation Contract
- TNC_IFM_NO_NEXT_SEGMENT - Sent by a Contracted Party to a Contracting Party if the Contracting Party sends a Next Segment attribute for a given Segmented Message Exchange but there is no additional segment waiting for delivery.
- TNC_IFM_UNEXPECTED_SEGMENT - Sent by the recipient of a Segment Envelope attribute if the Segment Envelope is not flagged as being the first segment of an exchange but the recipient has no record of prior segments for the exchange.
- TNC_IFM_MESSAGE_NOT_FOUND - Sent by the recipient of a Contract Exemption attribute if the recipient of that attribute is unable to deliver the indicated oversized message.
- TNC_IFM_SEG_VIOLATION - Sent by either a Contracting or Contracted Party if they receive a Segment Envelope or a Next Segment attribute, respectively, which indicates the sender is not conforming to the recipient's understanding of the use of Segmented Message Exchanges as permitted by a Segmentation Contract.

All of the IF-M Segmentation IF-M Error attributes defined in this specification MUST be sent using exclusive delivery.

3.7.1 Errors that Cancel Segmentation Contracts

In the case that a Segmentation Contract exists between two parties and one of those parties sends the other a TNC_IFM_SEG_VIOLATION using the contract's IF-M Subtype, both parties MUST treat that Segmentation Contract as cancelled. In the case that a Segmentation Contract Request attribute results in a TNC_IFM_SEG_CONTRACT_REJECTED, then not only is no new contract created, but an existing Segmentation Contract that the request would have replaced is also cancelled. In the case that a Segmentation Contract Response attribute generates any IF-M Error, instead of creating a Segmentation Contract, this is treated as a failure to agree upon a contract - no new Segmentation Contract is created, and any existing Segmentation Contract that would be replaced by the Segmentation Contract Response is automatically cancelled. Segmentation Contracts other than that specific Segmentation Contract associated with the IF-M Error MUST NOT be cancelled by that IF-M Error.

3.7.2 Errors that Cancel Segmented Message Exchanges

In the case that there is an ongoing Segmented Message Exchange and one party in that exchange sends a TNC_IFM_UNEXPECTED_SEGMENT, TNC_IFM_NO_NEXT_SEGMENT, or a TNC_IFM_SEG_VIOLATION which identifies an attribute sent in fulfillment of that Segmented Message Exchange, both parties MUST treat that Segmented Message Exchange as cancelled. Segmented Message Exchanges not associated with the given IF-M Error MUST NOT be cancelled.

3.7.3 Reaching Limits on Supported IF-M Segmentation Activities

In the case that an IF-M Component has reached the limit of the number of Segmentation Contracts for which it can act as a Contracting Party for a given IF-M Subtype, that Component MUST NOT attempt to establish a new Segmentation Contract until at least one of those existing Segmentation Contracts is cancelled. In the case that an IF-M Component has reached the limit of the number of Segmentation Contracts for which it can act as a Contracted Party for a given IF-M Subtype, that Component MUST respond to any requests to establish a new Segmentation Contract with a TNC_IFM_SEG_CONTRACT_REJECTED error with both the Acceptable Max Message Size and the Acceptable Max Segment Size set to 0xFFFFFFFF. (See section 4.12 for more on TNC_IFM_SEC_CONTRACT_REJECTED errors.) In either case, an IF-M Component can still request modification of an existing Segmentation Contract, as described in section 3.3. Likewise,

such IF-M Components can still request cancellation of existing Segmentation Contracts, as described in section 3.4.

3.7.4 Errors in the Base Message Delivered by a Segmented Message Exchange

It is possible that a Base Message delivered via a Segmented Message Exchange leads to an error on the recipient. For example, the attribute in the Base Message might have been constructed with an invalid parameter. In the case that this happens, the IF-M Error attribute generated MUST indicate the Base Message as the cause of the error, rather than the IF-M message containing the Segment Envelope that conveyed it to the recipient. This is generally done using the value of the Message Identifier field in the Base Message's IF-M message header, which would be part of the payload of the first Segment Envelope. Some IF-M Error attributes identify the erroneous attribute by providing an offset in bytes between the start of the IF-M message header and the start of the erroneous attribute's IF-M attribute header. In this case, the offset MUST be calculated using the reassembled Base Message (or at least reassembled up to and including the detection of the error).

In addition, some IF-M Error codes include copies of fields from the erroneous attribute's IF-M attribute header to aid identification. When constructing such an error, the IF-M Component MUST use the IF-M attribute header of the erroneous attribute within the Base Message (not the IF-M attribute header of the Segment Envelope).

The actual error code generated in these circumstances MUST NOT be one of the error codes defined in the IF-M Segmentation specification. This is because IF-M Segmentation attributes cannot themselves be segmented. (This rule is enforced by the requirement that messages containing Segment Envelope attributes need to be smaller than the maximum segment size limit, as described in section 3.5.2, and the fact that all other IF-M Segmentation attributes are smaller than 44 bytes and thus would never exceed the maximum segment size limit of any valid Segmentation Contract.)

3.7.5 IF-M Segmentation Attributes and Externally Defined IF-M Errors

An IF-M Segmentation attribute might elicit an IF-M Error attribute other than one of the IF-M Errors defined in the IF-M Segmentation specification. Such errors are referred to in this specification as "externally defined IF-M errors", reflecting that they are defined in specification other than the IF-M Segmentation specification. For example, if an IF-M Segmentation attribute contained an invalid parameter, this would result in a TNC_IFM_INVALID_PARAMETER IF-M Error, which is defined in the IF-M TLV Binding specification [3]. In particular, a TNC_IFM_INVALID_PARAMETER error code MUST be returned in all of the following circumstances:

- A Segmentation Contract Request specifies a maximum segment size or a maximum message size that is less than 64 bytes.
- A Segmentation Contract Response specifies a maximum segment size or a maximum message size that is less than 64 bytes.
- A Segmentation Contract Response specifies either a maximum segment size or a maximum message size that is larger than the corresponding field of the Segmentation Contract Request to which it is responding.
- A Segmentation Contract Response allows Segmented Message Exchanges, but the Segmentation Contract Request to which it is responding prohibits such exchanges.
- A Segmentation Contract Response prohibits Segmented Message Exchanges, but the Segmentation Contract Request to which it is responding allows such exchanges.

In the case that a Segmentation Contract Response results in an externally defined IF-M error, both parties MUST automatically treat the Segmentation Contract that that would normally have been established by the Segmentation Contract Response as being cancelled. If that Segmentation Contract Response was replacing an existing Segmentation Contract, that existing Segmentation Contract MUST be cancelled in the same way as if a request to modify an existing Segmentation Contract was rejected.

In the case that a Segment Envelope or a Next Segment attribute results in an externally defined IF-M error, then both parties MUST effectively backtrack the state of the Segmented Message Exchange to the point immediately before the erroneous attribute was sent. In other words, the sender of the erroneous attribute remains the next party that needs to send an attribute in the back-and-forth Segmented Message Exchange. The sender of the original erroneous attribute MUST either resend the original attribute with the error fixed or, in the case that it is unable to fix the error, cancel the Segmented Message Exchange using the appropriate method as described in section 3.5.2.1. To avoid infinite loops where the same, erroneous attribute is constantly resent, after the 5th attempt to send the same attribute, the recipient SHOULD respond by cancelling the Segmented Message Exchange rather than responding with an IF-M Error. Similarly, the sender of an attribute MUST cancel the Segmented Message Exchange upon receiving the 5th consecutive IF-M Error to its attempts to send a particular attribute in that Segmented Message Exchange.

In the case that a Cancel attribute results in an externally defined IF-M error, then the indicated Segmentation Contract and/or Segmented Message Exchange is not cancelled. Note that a Cancel attribute can only result in an IF-M Error if the recipient is unable to correctly process the attribute. (For example, if an unrecognized set of flags is used, or if the attribute is not the right size.) Recipients of Cancel attributes MUST NOT reject calls to cancel Segmentation Contracts of Segmented Message Exchanges for any other reason. Similarly, if the recipient of a Cancel attribute does not have any record of the associated Segmentation Contract and/or Segmented Message Exchange, it MUST NOT treat that as an error condition. Instead, such a situation ought to be treated as successful, since, in the end, both parties will be in agreement that there is no active contract or exchange matching those identified in the Cancel attribute.

In the case that an Oversized Message or Contract Exemption attribute result in an externally defined IF-M error, the attempt to create the Contract Exemption fails; the Contracting Party needs to send a new Contract Exemption attribute to restart the process of sending the oversized message.

4 IF-M Segmentation Message and Attributes

This section describes the format and semantics of IF-M Segmentation.

4.1 IF-M Subtype (AKA IF-M Component Type)

The TNC IF-TNCCS protocol provides a general message-batching protocol capable of carrying one or more IF-M messages between the TNC Client and TNC Server. When IF-TNCCS is carrying an IF-M message, the IF-TNCCS message headers contain a 32-bit identifier called the IF-M Subtype. The IF-M Subtype field indicates the IF-M Component associated with all of the IF-M attributes carried by the IF-TNCCS message.

This specification does not add a new IF-M Subtype. Instead, all attributes defined in this specification are sent with the IF-M Subtype of some other IF-M binding. This allows the IF-M Segmentation attributes to be delivered to the IF-M Components registered to retrieve attributes associated with that binding. If those IF-M Components do not support IF-M Segmentation, the IF-M Segmentation attributes will not be recognized, and the components will either respond with an IF-M Error or silently discard the IF-M Segmentation attribute. If the component supports IF-M Segmentation, it MUST process that attribute according to the procedures described in this specification.

4.2 IF-TNCCS and IF-M Messages

An IF-M message is wrapped within an IF-TNCCS message. Figure 4 shows the relationship between IF-M and IF-TNCCS messages. A single IF-M message might contain one or more IF-M attributes. All of these attributes within a single IF-M message use the same IF-M Subtype value. Note, however, that a single IF-TNCCS batch might contain multiple IF-TNCCS and IF-M messages, and each of those messages might use different IF-M Subtypes.

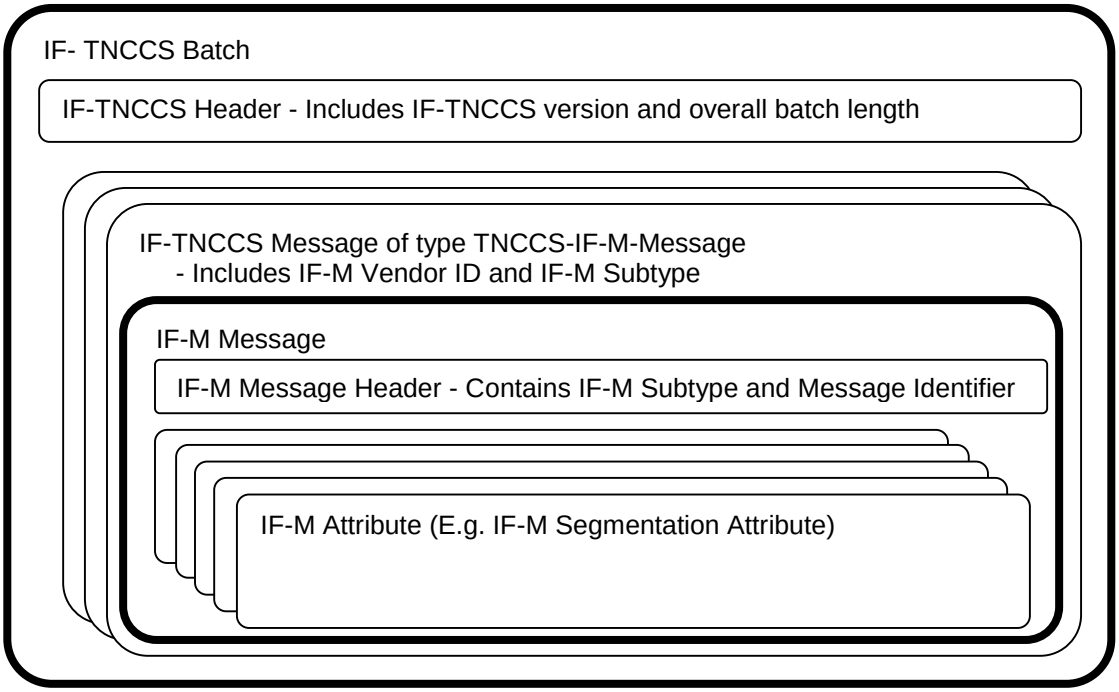

Figure 4 - IF-TNCCS and IF-M Messages

For more information on IF-TNCCS and IF-M messages and message headers, see the TNC IF-TNCCS TLV Binding [10] and TNC IF-M TLV Binding [3] specifications, respectively.

4.3 IF-M Attribute Header

IF-M Segmentation is an extension of the IF-M protocol described in the TNC architecture. IF-M is designed to be very flexible in order to carry many types of IF-M attributes. Figure 5, reproduced from the IF-M TLV Binding specification [3], shows the format of an IF-M header and attribute.

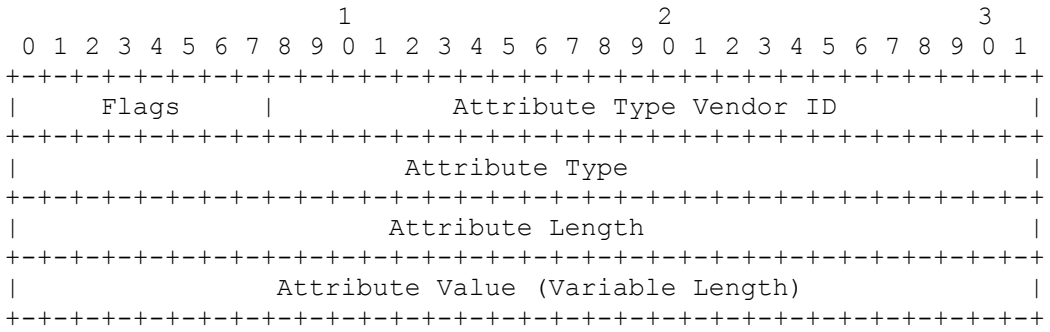

Figure 5 - IF-M Header and Attribute Format

TLV Field	Description
Flags	This field defines flags affecting the processing of the IF-M attributes. Permissible flags are given in the IF-M TLV Binding specification. [3]
Attribute Type Vendor ID	This field indicates the owner of the name space associated with the Attribute Type. Attributes defined in the IF-M Segmentation specification have a value corresponding to the TCG SMI Private Enterprise Number value (0x005597). The IF-M Error attribute is defined in the IF-M TLV Binding specification and uses the IETF SMI Private Enterprise Number Value (0x000000). See Table 2 for more information.
Attribute Type	This field defines the type of the Attribute. The values corresponding to IF-M Segmentation attributes are given in Table 2.
Attribute Length	This field contains the length in octets of the entire Attribute, including the Attribute's header.
Attribute Value	This field contains the IF-M Attribute.

Table 1 - Fields of the IF-M Header and Attribute

4.4 IF-M Segmentation Attribute Overview

The attributes defined in this specification appear below with a short summary of their purposes. Each attribute is described in greater detail in subsequent sections.

- **Segmentation Contract Request** - This attribute is sent by a would-be Contracting Party to request the establishment of a Segmentation Contract with the recipient, who will become the Contracted Party if it accepts the contract. This attribute MAY be sent using exclusive delivery, but doing so is not required. This attribute MUST be sent in a manner that includes the sender's IMC ID or IMV ID, as appropriate, such as by using the sendMessageLong method as described in section 3.9.5 of the IF-IMV 1.4 (or 1.3) specification [5] or section 4.3.12 of the IF-IMC 1.3 specification [4].
- **Segmentation Contract Response** - This attribute is sent in response to a Segmentation Contract Request to indicate that the request's recipient will be bound by a Segmentation Contract with message and segment size limits as described in the response attribute. This attribute MUST be sent using exclusive delivery.

- **Segment Envelope** - This attribute is used to convey one segment of a Base Message from a Contracted Party to a Contracting Party during a Segmented Message Exchange. This attribute MUST be sent using exclusive delivery.
- **Next Segment** - This attribute is sent by a Contracting Party in response (although possibly not an immediate response) to a Segment Envelope in order to request the delivery of the next segment of the Segmented Message Exchange. This attribute MUST be sent using exclusive delivery.
- **Cancel** - This attribute is sent by a Contracted Party to a Contracting Party in order to cancel the Segmentation Contract between them and/or to cancel a specific Segmented Message Exchange between those parties. The attribute can perform either action or both simultaneously. This attribute MUST be sent using exclusive delivery.
- **Oversized Message** - This attribute is sent by a Contracted Party to a Contracting Party to indicate that the Contracted Party has generated a Base Message but is unable to deliver it without violating the Segmentation Contract between these parties. This attribute MUST be sent using exclusive delivery.
- **Contract Exemption** - This attribute is sent by a Contracting party to a Contracted Party to grant an exemption to a Segmentation Contract to allow a specific oversized Base Message to be delivered. This attribute MUST be sent using exclusive delivery.

All IF-M Components conformant with this specification MUST be capable of serving as a Contracted Party and therefore MUST be capable of receiving Segmentation Contract Request and Contract Exemption attributes, and MUST be capable of sending Segmentation Contract Response, Cancel, and Oversized Message attributes. In the case that such a component also supports Segmented Message Exchanges, it MUST support sending Segment Envelope attributes and MUST support receiving Next Segment attributes.

In the case that an IF-M Component supports acting as a Contracting Party, it MUST support sending Segmentation Contract Request attributes, and MUST be capable of receiving Segmentation Contract Response, Cancel, and Oversized Message attributes. In the case that such a component also supports Segmented Message Exchanges, it MUST support receiving Segment Envelope attributes and MUST support sending Next Segment attributes. Contracting Parties MAY support sending Contract Exemption attributes, but are not required to do so.

4.5 IF-M Segmentation Attribute Enumeration

IF-M attribute types are identified in the IF-M attribute header (see section 4.2) via the Attribute Type Vendor ID and Attribute Type fields. Table 2 identifies the appropriate values for these fields for each attribute type defined in the IF-M Segmentation specification.

Attribute Name	Attribute Type Vendor ID	Attribute Type	Description
Segmentation Contract Request	0x005597	0x00000021	Request to establish a Segmentation Contract.
Segmentation Contract Response	0x005597	0x00000022	Indicate acceptance of a Segmentation Contract.
Segment Envelope	0x005597	0x00000023	Send a Message Segment as part of a Segmented Message Exchange.
Next Segment	0x005597	0x00000024	Request the next Message Segment in a Segmented Message Exchange.
Cancel	0x005597	0x00000025	Cancel a Segmentation Contract and/or an ongoing Segmented Message Exchange.

Oversized Message	0x005597	0x00000026	Report a message that cannot be delivered due to limits imposed by a Segmentation Contract.
Contract Exemption	0x005597	0x00000027	Grant an exemption to a Segmentation Contract to allow an oversized Base Message to be delivered.
Reserved	0x005597	0x00000028 - 0x0000002F	These attribute types are reserved for future use in revisions to IF-M Segmentation.
IF-M Error	0x000000	0x00000008	An error attribute as defined in the IF-M TLV Binding.

Table 2 - IF-M Segmentation Attribute Enumeration

4.6 Segmentation Contract Request

A would-be Contracting Party sends this attribute to request the establishment of a Segmentation Contract with the attribute's recipient.

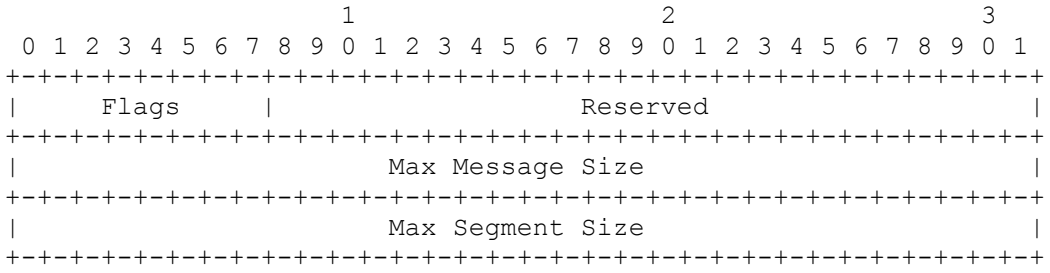

Figure 6 - Segmentation Contract Request Attribute

Field	Description
Flags	**Bit Encoding**: Bit 0-7 - Reserved **Description**: Reserved for future use. This field MUST be set to zero on transmission and ignored upon reception.
Reserved	Reserved for future use. This field MUST be set to zero on transmission and ignored upon reception.
Max Message Size	A 4-byte unsigned integer indicating the largest message size (including its IF-M message header fields) allowed under this Segmentation Contract. This value MUST be greater than or equal to 64. A value of 0xFFFFFFFF indicates that the Segmentation Contract imposes no upper limit on message size.
Max Segment Size	A 4-byte unsigned integer indicating the largest Message Segment size allowed under this Segmentation Contract. This value MUST be greater than or equal to 64. A value of 0xFFFFFFFF indicates that Segmented Message Exchanges are not permitted under this Segmentation Contract.

Table 3 - Segmentation Contract Request Attribute Fields

The Max Message Size field specifies the upper limit on message size measured in bytes imposed by the requested Segmentation Contract. The IF-M message header of a message contributes to its total size for the purpose of this calculation, as do all contained IF-M attributes and their headers. Note that some IF-M bindings impose their own upper limits on attribute and/or message sizes. A Segmentation Contract would not absolve a Contracted Party from conforming to size limits imposed by other parties regardless of the value specified in the contract itself. A value of 0xFFFFFFFF (i.e., the largest possible value for this field) indicates that the Segmentation Contract imposes no upper limit on message sizes. Such a contract MAY still impose limits on segment size.

The Max Segment Size field specifies the upper limit on segment size imposed by the requested contract. Any message larger than this maximum segment size MUST be broken into segments and transmitted via a Segmented Message Exchange as described in section 3.5.2. A message smaller than or equal to this maximum segment size MUST NOT be sent using a Segmented Message Exchange. A value of 0xFFFFFFFF indicates that the Segmentation Contract does not allow Segmented Message Exchanges. Such a contract MAY still impose limits on message size.

Because there can only be a single Segmentation Contract in effect between a given Contracting Party and Contracted Party for a given IF-M Subtype, there MUST NOT be more than one Segmentation Contract Request within a single IF-M message.

A request for a Null Contract would have a value of 0xFFFFFFFF in both the Max Message Size and the Max Segment Size fields. A Null Contract is equivalent to their being no Segmentation Contract between the Contracting and Contracted Parties for the given IF-M Subtype.

The Segmentation Contract Request is the only IF-M Segmentation attribute that MAY be sent without using exclusive delivery. (As described in section 3.6, for all other IF-M Segmentation attributes, exclusive delivery is required to avoid problems when dealing with Multi-Component Parties.) There are two reasons that Segmentation Contract Requests are allowed to be sent without using exclusive delivery. First, as described in section 3.6, a Segmentation Contract Request will often reveal the existence of Multi-Component Parties, and this specification describes how to detect and respond to these situations. As such, a “broadcast” Segmentation Contract Request (i.e., a request that is not sent using exclusive delivery) is the best way to determine if one has a Multi-Component Party, allowing the would-be Contracting Party to respond appropriately. Secondly, a would-be Contracting Party might not have the component identifiers necessary to send a Segmentation Contract Request exclusively, but they would receive those identifiers in any Segmentation Contract Response they receive. As such, allowing Segmentation Contract Requests to be broadcast allows would-be Contracting Parties to bootstrap IF-M Segmentation relationships that might not otherwise be possible due to a lack of available identity information.

Note, however, that Segmentation Contract Requests MAY be sent using exclusive delivery as well. A would-be Contracting Party could do this if it knew the component identifier of the targeted Contracted Party and had no need to check for other components on the recipient endpoint that might be registered to use the same IF-M Subtype.

4.7 Segmentation Contract Response

This attribute is sent to indicate acceptance of a Segmentation Contract. The sender then becomes the Contracted Party for that contract.

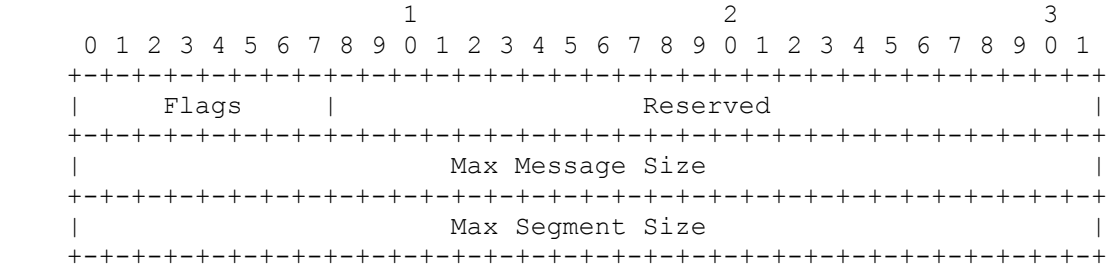

Figure 7 - Segmentation Contract Response Attribute

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>Flags</td><td>
<table>
<tr><th>Bit Encoding</th><th>Description</th></tr>
<tr><td>Bit 0-7 - Reserved</td><td>Reserved for future use. This field MUST be set to zero on transmission and ignored upon reception.</td></tr>
</table>
</td></tr>
<tr><td>Reserved</td><td>Reserved for future use. This field MUST be set to zero on transmission and ignored upon reception.</td></tr>
<tr><td>Max Message Size</td><td>A 4-byte unsigned integer indicating the largest message size (including its IF-M header fields) allowed under this Segmentation Contract.

This value MUST be greater than or equal to 64.

A value of 0xFFFFFFFF indicates that the Segmentation Contract imposes no upper limit on message size (although the IF-M TLV Binding specification and other IF-M bindings can impose their own size limits).</td></tr>
<tr><td>Max Segment Size</td><td>A 4-byte unsigned integer indicating the largest Message Segment size allowed under this Segmentation Contract.

This value MUST be greater than or equal to 64.

A value of 0xFFFFFFFF indicates that Segmented Message Exchanges are not permitted under this Segmentation Contract.</td></tr>
</table>

Table 4 - Segmentation Contract Response Attribute Fields

The Max Message Size field specifies the upper limit on message size imposed by the accepted Segmentation Contract measured in bytes. Note that it is the value of this field, rather than the value of the corresponding field in the Segmentation Contract Request, that sets the actual limits of the resulting Segmentation Contract. The sender of this attribute MUST specify a Max Message Size value that is less than or equal to the value of the corresponding field in the Segmentation Contract Request. The sender MAY specify a value less than the request's Max Message Size, but MUST always specify a value greater than or equal to 64.

The Max Segment Size field specifies the upper limit on segment size imposed by the accepted contract. As with the Max Message Size field, it is the value of the field in this attribute that specifies the actual limits of the Segmentation Contract. In the case that the Segmentation Contract Request provides a value of 0xFFFFFFFF in its Max Segment Size field, the Segmentation Contract Response MUST provide a value of 0xFFFFFFFF in its Max Segment Size field. (See section 3.7.5 for procedures if this requirement is violated.) In other words, if the Segmentation Contract Request disallows Segmented Message Exchanges, the actual contract specified by the Segmentation Contract Response MUST also disallow Segmented Message Exchanges. In the case that the Segmentation Contract Request provides a Max Segment Size value other than 0xFFFFFFFF, the Segmentation Contract Response's Max Segment Size MUST be less than or equal to the corresponding value from the request and MUST be greater than 64. (Again, see section 3.7.5 for instructions on handling violations of this requirement.)

4.8 Segment Envelope

A Segment Envelope is used to transmit a Message Segment from a Contracted Party to a Contracting Party during a Segmented Message Exchange.

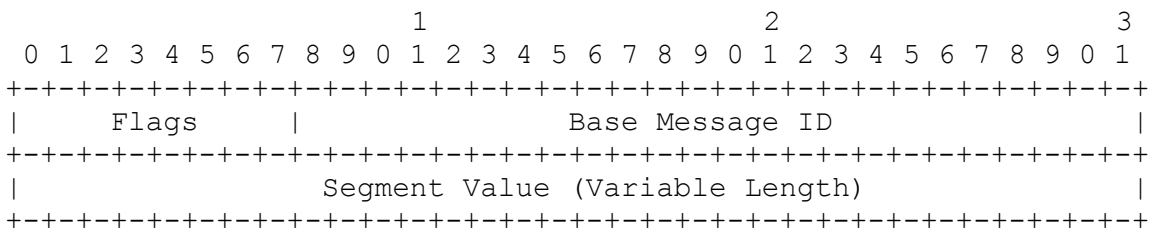

Figure 8 - Segment Envelope

Field	Description
Flags	**Bit Encoding** / **Description** Bit 0 - Start: If set (1), indicates that this is the first Message Segment of a given Base Message to be associated with the given Base Message ID. If unset (0), indicates that there has been at least one preceding Message Segment for the given Base Message. Bit 1 - More: If set (1), indicates that there are additional segments of the Base Message following this one. If unset (0), indicates that this is the final Message Segment for the given Base Message. Bit 2-7 - Reserved: Reserved for future use. This field MUST be set to zero on transmission and ignored upon reception.
Base Message ID	This field contains an identifier that is used to link individual Message Segments to their Base Message. The Contracted Party assigns this value in the first segment sent and all segments of the same Base Message are sent with the same Base Message ID. The Base Message ID value of 0 (0x000000) is reserved and MUST NOT be used in this field.
Segment Value	This field contains a Message Segment.

Table 5 - Segment Envelope Fields

The Start flag MUST be set when sending the first segment of a given Base Message. The Start flag MUST be unset for all subsequent segments of the same Base Message. In the case that a Contracting Party receives a Segment Envelope with the Start flag set, it MUST clear any prior associations with the named Base Message ID for the given Segmentation Contract. In other words, even if the Contracting Party had previously associated the Base Message ID with some prior Segmented Message Exchange, that prior association is cancelled and the Base Message ID becomes associated with the new Segmented Message Exchange.

The More flag MUST be set in Segment Envelopes for all but the final segment of a Segmented Message Exchange. The More flag MUST be unset when conveying the final segment of the Segmented Message Exchange. The Contracting Party MUST NOT send a Next Segment attribute in response to a Segment Envelope with the More flag unset.

The Base Message ID is used to uniquely identify a particular Segmented Message Exchange relative to a given Segmentation Contract. Each time a Contracted Party starts a new Segmented Message Exchange in fulfillment of a Segmentation Contract, it assigns that Segmented Message Exchange a new Base Message ID. This Base Message ID value is used in all Segment Envelope attributes sent during that particular Segmented Message Exchange (and is also used in other attributes, such as the Next Segment attribute and the Cancel attribute, to refer to this particular exchange). Note that a particular Base Message ID is only unique relative to a given Segmentation

Contract. A single IF-M Component might be a Contracting Party for multiple Segmentation Contracts, and at any given time some of those contracts might have ongoing Segmented Message Exchanges that have the same Base Message ID. IF-M Components conformant with this specification MUST be able to track multiple Segmented Message Exchanges for different Segmentation Contracts even when the exchanges' Base Message IDs collide. By contrast, in the case that there are multiple active Segmented Message Exchanges for a single Segmentation Contract, each of those exchanges MUST have a different Base Message ID. Starting a new Segmented Message Exchange using a previously used Base Message ID immediately terminates any prior Segmented Message Exchange for the same Segmentation Contract previously associated with that Base Message ID.

Note that, while implementers might be tempted to base the Base Message ID on the Message Identifier field from the Base Message's header doing so is not recommended. The Base Message ID is 3 bytes and reserves the value of 0 while the Message Identifier is 4 bytes and 0 is not reserved. Since IF-M Segmentation functions might have no influence over how the Message Identifier is generated, any attempt to use the larger Message Identifier as a basis for the smaller Base Message ID would need to include special functions to avoid value collisions or use of the reserved 0 value. Given this, it is likely simpler to generate Base Message IDs independently. For this reason, the Contracting and Contracted party MUST NOT assume any relationship between the Base Message ID and the Message ID from the header of that Base Message.

The Segment Value contains the Message Segment conveyed in the Segment Envelope. All Message Segments MUST be divided along byte boundaries and MUST NOT be padded. The size of the Segment Value field varies, but is always equal to 16 bytes less than the Attribute Length field of the Segment Envelope's IF-M header. (The fixed-length fields in the IF-M attribute header and the Segment Envelope come to 16 bytes). Note that this refers to the IF-M attribute header of the Segment Envelope itself, and not to the IF-M header of some attribute in the Base Message that would be included in the Segment Value field. The size of the Segment Value field MUST also be at least 20 bytes less than the maximum segment size of the Segmentation Contract under which it is sent. This ensures that the Segment Envelope does not itself require segmentation under the Segmentation Contract.

4.9 Next Segment

The Next Segment attribute is sent from a Contracting Party to a Contracted Party in order to request the next Message Segment of a Segmented Message Exchange.

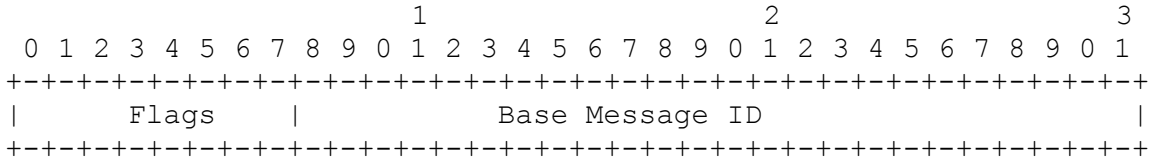

Figure 9 - Next Segment Attribute

Field	Description	
Flags	**Bit Encoding**	**Description**
	Bit 0 - Cancel Exchange	If this bit is set (1), the Contracting Party is cancelling the Segmented Message Exchange associated with this Base Message ID.
	Bit 1-7 - Reserved	Reserved for future use. This field MUST be set to zero on transmission and ignored upon reception.
Base Message ID	A copy of the Base Message ID from the Segment Envelope.	

Table 6 - Next Segment Attribute Fields

The Cancel Exchange flag is used by the Contracting Party to cancel an ongoing Segmented Message Exchange. The Contracted Party MUST NOT send either a Segment Envelope or a TNC_IFM_NO_NEXT_SEGMENT IF-M Error in response to a Next Segment attribute with the Cancel Exchange flag set. In the case that the Cancel Exchange flag is unset, the Contracted Party MUST either respond with a Segment Envelope containing the next Message Segment of the Segmented Message Exchange or with an appropriate IF-M Error if an error condition prevents the Contracted Party from sending the next Message Segment.

The Base Message ID contains a copy of the Base Message ID associated with the given Segmented Message Exchange, as assigned in the first Segment Envelope of that exchange.

4.10 Cancel

The Cancel attribute is sent by a Contracted Party to a Contracting Party to cancel a Segmented Message Exchange and/or to cancel a Segmentation Contract.

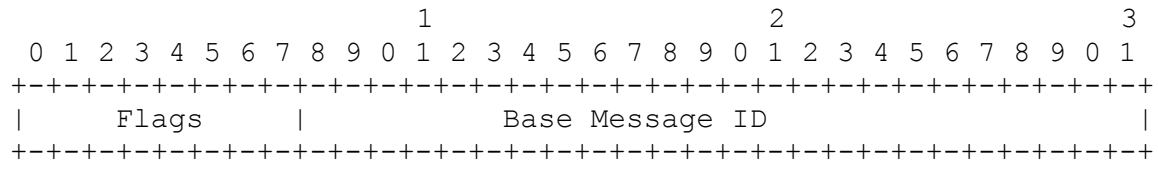

Figure 10 - Cancel Attribute

Field	Description	
Flags	**Bit Encoding**	**Description**
	Bit 0 - Cancel Contract	If set (1), this attribute cancels the associated Segmentation Contract. If unset (0), no Segmentation Contract is cancelled by this attribute.
	Bit 1-7 - Reserved	Reserved for future use. This field MUST be set to zero on transmission and ignored upon reception.
Base Message ID	A copy of the Base Message ID of the Segmented Message Exchange to be cancelled. If this field is set to 0 (0x000000), no Segmented Message Exchange is cancelled by this attribute.	

Table 7 - Cancel Attribute Fields

The Cancel attribute can be used by a Contracted Party to cancel an ongoing Segmented Message Exchange, to cancel a Segmentation Contract, or both. The field values determine which of these actions is taken. Both parties MUST immediately treat the indicated contract and/or exchange as cancelled.

The Cancel Contract flag indicates whether the associated Segmentation Contract is to be cancelled. If the Cancel Contract flag is set, the Segmentation Contract associated with the attribute (i.e., the Segmentation Contract with the Cancel attribute sender as the Contracted Party, the Cancel attribute recipient as the Contracting Party, and associated with the IF-M Subtype used to send this attribute) is immediately canceled. No other Segmentation Contracts are cancelled. If the Cancel Contract flag is unset, this attribute does not cancel any Segmentation Contract.

The Base Message ID indicates whether the Cancel attribute cancels a Segmented Message Exchange. In the case that the Base Message ID field is not set to 0x000000, the Segmented Message Exchange associated with the given Base Message ID within the associated Segmentation Contract MUST be treated as cancelled by both the Contracted and Contracting Parties. In the case that the Base Message ID field is set to 0x000000, no Segmented Message Exchange is cancelled by this attribute. (Recall that the value of 0x000000 is reserved and MUST NOT be associated with

any Base Message ID.) Note that it is only possible to cancel a single Segmented Message Exchange in a single Cancel attribute.

A Contracting Party receiving a request to cancel a non-existent Segmentation Contract or non-existent Segmented Message Exchange MUST NOT treat this as an error.

Barring errors, all Cancel requests automatically succeed. The recipient of a Cancel request MUST NOT reject the request, although they MAY attempt to re-establish a new Segmentation Contract or Segmented Message Exchange after cancelling the old one.

4.11 Oversized Message

The Oversized Message attribute is sent from a Contracted Party to a Contracting Party when the Contracted Party generates a message, but this message is too large to send without violating the relevant Segmentation Contract.

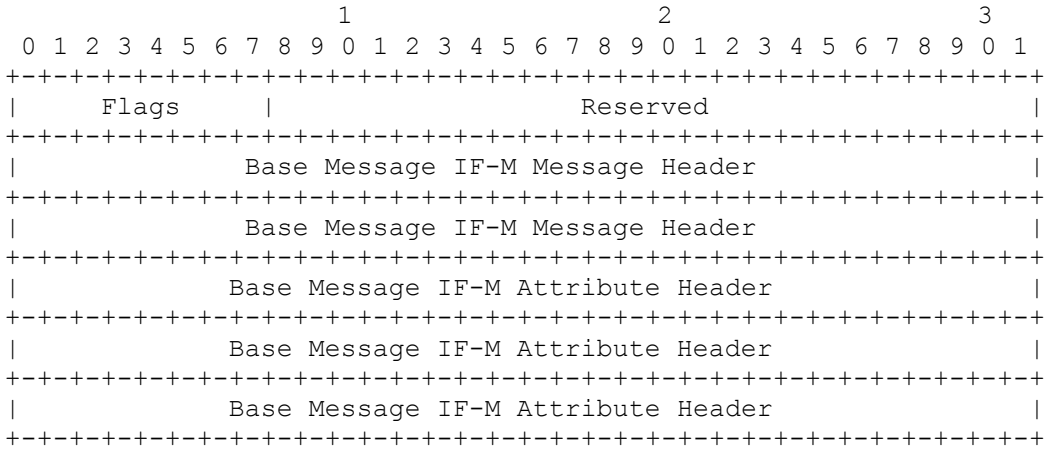

Figure 11 - Oversized Message Attribute

Field	Description
Flags	**Bit Encoding**: Bit 0-7 - Reserved **Description**: Reserved for future use. This field MUST be set to zero on transmission and ignored upon reception.
Reserved	Reserved for future use. This field MUST be set to zero on transmission and ignored upon reception.
Base Message IF-M Message Header	These fields contain an exact copy of the oversized Base Message's IF-M message header.
Base Message IF-M Message Header	These fields contain an exact copy of the attribute header of the IF-M attribute contained in the oversized Base Message.

Table 8 - Oversized Message Attribute Fields

The purpose of this attribute is to inform the Contracting Party of the presence and size of the unsent message, as well as the type and size of the oversized attribute that resulted in the oversized message. Recall from section 3.5.1 that oversized messages with multiple attributes MUST attempt to redistribute those attributes among multiple messages prior to resorting to sending an Oversized Message attribute. As such, an oversized message MUST only identify a single IF-M attribute.

The Flags field and subsequent Reserved field are reserved for use in future version of the IF-M Segmentation specification. The remaining fields of this attribute duplicate the values of the unsent message's IF-M message header fields and the Oversized Attribute's header fields.

4.12 Contract Exemption

The Contract Exemption attribute MAY be sent in response to an Oversized Message attribute to indicate that the Contracting Party is willing to waive the Segmentation Contract to allow a specific oversized message to be delivered.

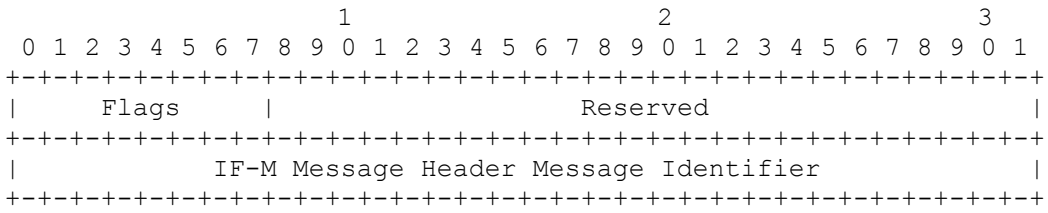

Figure 12 - Contract Exemption Attribute

Field	Description	
Flags	**Bit Encoding**	**Description**
	Bit 0 - Honor Segmentation	If set (1), the Maximum Segment Size limits of the Segmentation Contract MUST be honored with the oversized message is sent. If unset (0), the oversized message MUST NOT be sent using a Segmented Message Exchange.
	Bit 1-7 - Reserved	Reserved for future use. This field MUST be set to zero on transmission and ignored upon reception.
Reserved	Reserved for future use. This field MUST be set to zero on transmission and ignored upon reception.	
IF-M Message Header Message Identifier	This field contains an exact copy of the Message Identifier field from the IF-M Message Header within the body of the Oversized Message attribute to which the Contract Exemption applies.	

Table 9 - Contract Exemption Attribute Fields

The Contract Exemption attribute can be used by a Contracting Party to request the delivery of an oversized message by granting that message an exemption from the Segmentation Contract. Each Contract Exemption attribute only applies to a single oversized message; if there are multiple oversized messages awaiting delivery, a Contract Exemption needs to be sent for each one the Contracting Party wishes to have delivered.

A Contract Exemption attribute automatically exempts a particular message from the Max Message Size limit of a Segmentation Contract. The Honor Segmentation flag determines whether there is also an exemption to the Max Segment Size limit of the Segmentation Contract. In the case that the flag is unset (0), the oversized message MUST NOT be sent using a Segmented Message Exchange, regardless of whether the Segmentation Contract would normally require segmentation. In the case that the flag is set (1), the Max Segment Size limit of the Segmentation Contract MUST be honored. Note that setting the Honor Segmentation flag does not automatically mean that the oversized message will be sent using a Segmented Message Exchange - it just means that the Segmentation Contract governs the decision as to whether a Segmented Message Exchange is used. If the Segmentation Contract forbids Segmented Message Exchanges, then the value of this flag is moot, in that a Segmented Message Exchange is prohibited regardless of the flag value. A Contracting Party might choose to avoid the use of Segmented Message Exchanges if the oversized message is

so large that it would take a large number of Message Segments to deliver it and the Contracting Party wishes to avoid the associated network overhead.

The IF-M Message Header Message Identifier field contains the Message Identifier value from the Base Message IF-M Message Header fields encapsulated within the Oversized Message attribute. This identifies the specific oversized message to which the contract exemption is granted. Note that this is not the Message Identifier from the IF-M Message containing the Oversized Message attribute itself. In other words, the IF-M Message Header Message Identifier field identifies the oversized message itself, rather than the Oversized Message attribute that reports the existence of the oversized message.

When the Contracted Party receives a Contract Exemption attribute it MUST respond in one of two ways:

- In the case that it does not have a record of an oversized message whose Message Identifier matches the IF-M Message Header Message Identifier provided in the Contract Exemption attribute, it MUST respond with a TNC_IFM_MESSAGE_NOT_FOUND IF-M Error attribute.
- Otherwise, it MUST deliver the indicated message. Delivery will either be as a complete message or via a Segmented Message Exchange, as dictated by the combination of the Honor Segmentation flag in the Contract Exemption attribute and the ongoing Segmentation Contract.

A Contract Exemption MUST NOT cancel an existing Segmentation Contract.

All Contracted Parties MUST support receiving and correctly handling a Contract Exemption attribute. Contracting Parties MAY support sending a Contract Exemption attribute, but are not required to do so.

4.13 IF-M Error as Used by IF-M Segmentation

The IF-M Error attribute is defined in the IF-M TLV Binding specification [3], and its use here conforms to that specification. An IF-M Error can be sent due to an error in the IF-M exchange, as described in the IF-M TLV Binding, and might also be sent in response to error conditions specific to IF-M Segmentation. The latter case utilizes error codes defined below.

Table 10 lists the Error Code values for the IF-M Error attribute specific to IF-M Segmentation. In all of these cases, the Error Code Vendor ID field MUST be set to 0x005597, corresponding to the TCG SMI Private Enterprise Number. Note that the Error Code Vendor ID is not the same as the Attribute Type Vendor ID, which MUST be set to 0x000000, as shown in Table 2. The Error Information structures for each error type are described in the following subsections.

Note that IF-M Segmentation attributes might also result in an error condition covered by the IF-M Error Codes defined in the TNC IF-M TLV Binding. (E.g., an IF-M Segmentation attribute might have an invalid parameter, leading to an error code of TNC_IFM_INVALID_PARAMETER.) In this case, the appropriate IF-M Error Code value as defined in Section 5.2.13 of the IF-M TLV Binding specification MUST be used.

Error Code Value	Description
0x00000030	TNC_IFM_SEG_CONTRACT_REJECTED - Sent in response to a Segmentation Contract Request to indicate rejection of the proposed Segmentation Contract.
0x00000031	TNC_IFM_NO_NEXT_SEGMENT - Sent by a Contracted Party to a Contracting Party if the Contracting Party sends a Next Segment attribute for a given Segmented Message Exchange, but there is no additional segment waiting for delivery.

0x00000032	TNC_IFM_UNEXPECTED_SEGMENT - Sent by the recipient of a Segment Envelope if the Segment Envelope is not flagged as being the first segment of an exchange but the recipient has no record of prior segments for the exchange.
0x00000033	TNC_IFM_MESSAGE_NOT_FOUND - Sent by the recipient of a Contract Exemption attribute if it is unable to provide the indicated oversized message.
0x00000034	TNC_IFM_SEG_VIOLATION - Sent by either a Contracting or Contracted Party if that party receives an attribute associated with Segmented Message Exchange, but which violates the recipient's understanding of the relevant Segmentation Contract.
0x00000035 - 0x0000003F	RESERVED. These Error Codes are reserved for use by future revisions of the IF-M Segmentation specification. Any IF-M Error attribute using one of these Error Codes MUST be treated as indicating a fatal error on the sender without further interpretation.

Table 10 - IF-M Error Codes for IF-M Segmentation

The following subsections describe the structures present in the Error Information fields.

4.13.1 TNC_IFM_SEG_CONTRACT_REJECTED Information

The TNC_IFM_SEG_CONTRACT_REJECTED error code is used to indicate that a given IF-M Component is rejecting a request to form a Segmentation Contract. The fields of the error information structure indicate alternative size limits that the sender of the error might be willing to accept, or can indicate that the sender is unwilling to accept any Segmentation Contract at the current time.

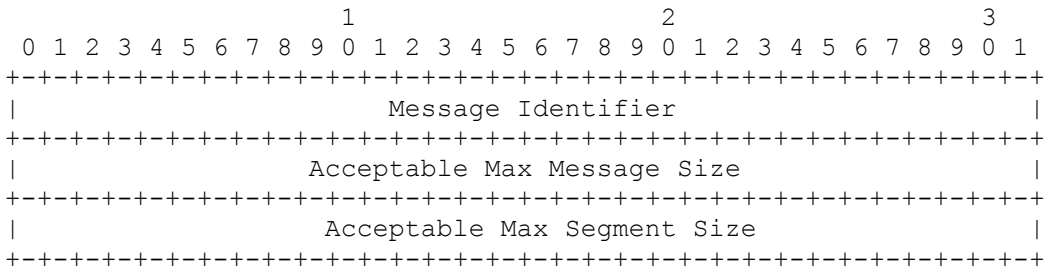

Figure 13 - Segmentation Contract Rejected Error Information

Field	Description
Message Identifier	This field MUST contain an exact copy of the Message Identifier field in the IF-M message header containing the Segmentation Contract Request that is being rejected.
Acceptable Max Message Size	This field contains the smallest Max Message Size the sender of the error is willing to accept. In the case that this field contains its largest possible value (0xFFFFFFFF), this indicates that the sender is unwilling to limit the size of the messages it sends.
Acceptable Max Segment Size	This field contains the smallest Max Segment Size the sender of the error is willing to accept. In the case that this field contains its largest possible value (0xFFFFFFFF), this indicates that the sender is unwilling to engage in Segmented Message Exchanges.

Table 11 - Segmentation Contract Rejected Error Information Fields

The Message Identifier is used to identify the Segmentation Contract Request to which this error is a response. It is included to avoid issues in the case where a would-be Contracting Party sends one

Segmentation Contract Request in one IF-M message and then sends a second message containing a request before the recipient has responded to the first request. While this situation is highly unlikely in most cases, it is possible over very slow networks or if the would-be Contracted Party is computationally constrained (such as in the case of some low-power embedded systems). In either case, it could be some time before the would-be Contracted Party is able to respond to a request, and the would-be Contracting Party might conclude that the relevant IF-M Component was not active when the first attempt went out and try again. Note that while a single IF-M message might contain multiple attributes for the given IF-M Subtype, a single IF-M message MUST NOT contain more than one Segmentation Contract Request, so it is not necessary to identify the specific attribute within the identified IF-M message.

The Acceptable Max Message Size and Acceptable Max Segment Size fields indicate the smallest acceptable values for a Segmentation Contract Request's Max Message Size and Max Segment Size fields, respectively, that the error sender would find acceptable. The values in these two fields can be used by a would-be Contracting Party to craft a new Segmentation Contract Request that is more likely to be acceptable to the IF-M Component that sent this error code. The smallest acceptable message size is provided here (rather than the largest acceptable message size) because, as described in section 3.2, the Contracted Party has the option of specifying contract size limits smaller than those given in the Segmentation Contract Request in its Segmentation Contract Response, and it is the latter attribute that sets the actual Segmentation Contract size limits. (There are some caveats to the Contracted Party's ability to do this. See sections 3.2.2 and 3.2.3 for these cases.) In other words, the would-be Contracted Party has no reason to reject a Segmentation Contract Request whose limits are too large because it can unilaterally reduce those limits. However, the would-be Contracted Party might face a Segmentation Contract Request whose limits are too small, in which case the would-be Contracted Party would want to inform the would-be Contracting Party about the smallest acceptable size limits in its rejection of that contract.

The sender and recipient of this error code MUST NOT treat this error code as establishing a Segmentation Contract. A would-be Contracting Party instead needs to send another Segmentation Contract Request, and the recipient respond with a Segmentation Contract Response, before a Segmentation Contract is in place.

If there was an existing Segmentation Contract for the associated IF-M Subtype where the error sender is the Contracted Party and the error recipient is the Contracting Party, both parties MUST treat that existing Segmentation Contract as cancelled.

The sender of this error code ought to be willing to accept a Segmentation Contract with the limits specified in the information fields at the time that it sends the error code. That said, the sender of this error code MAY reject a subsequent Segmentation Contract Request even if it uses the limits specified in this error code information. In other words, the sender of this error code is not committing itself to accepting a new Segmentation Contract Request, since circumstances can change between the time that the error code is sent and the time the new Segmentation Contract Request is received.

A value of 0xFFFFFFFF in the Acceptable Max Message Size indicates that the sender of the error code is unwilling to accept limits on the size of messages it sends. However, depending on the value of the Acceptable Max Segment Size field, it might still be willing to accept Segmentation Contracts that limit segment size.

A value of 0xFFFFFFFF in the Acceptable Max Segment Size indicates that the sender of the error code is unwilling to engage in Segmented Message Exchanges. However, depending on the value of the Acceptable Max Message Size, it might still be willing to accept Segmentation Contracts that limit message size.

If both the Acceptable Max Message Size and Acceptable Max Segment Size fields are set to 0xFFFFFFFF, then the sender of this error code is currently unwilling to accept any Segmentation Contract.

4.13.2 TNC_IFM_NO_NEXT_SEGMENT, TNC_IFM_UNEXPECTED_SEGMENT, TNC_IFM_NO_SUCH_MESSAGE, and TNC_IFM_SEG_VIOLATION Information

The TNC_IFM_NO_NEXT_SEGMENT, TNC_IFM_UNEXPECTED_SEGMENT, TNC_IFM_NO_SUCH_MESSAGE, and TNC_IFM_SEG_VIOLATION errors all share the same error information structure.

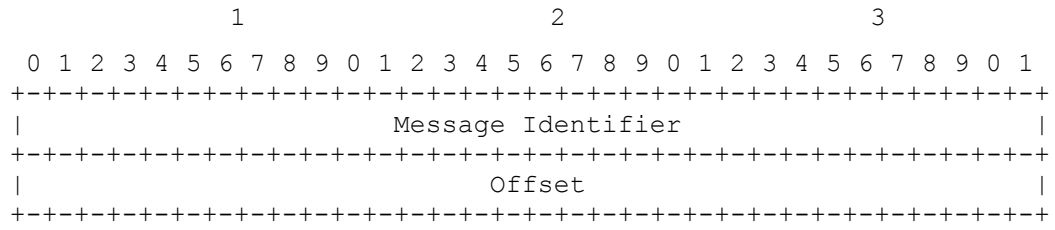

Figure 14 - No Next Segment et al. Error Information

Field	Description
Message Identifier	This field MUST contain an exact copy of the Message Identifier field in the IF-M Message Header of the IF-M message that caused this error.
Offset	This field MUST contain an octet offset from the start of the IF-M message header of the IF-M message that contained the attribute that caused this error, to the start of the offending IF-M attribute.

Table 12 - No Next Segment et al. Error Information Fields

The Message Identifier field is used to identify the IF-M message containing the attribute that led to this error condition. The Offset field is used to identify the specific attribute within the message that caused the error, since it is possible for a single IF-M message to contain multiple attributes that could be responsible for the indicated error.

4.13.2.1 TNC_IFM_NO_NEXT_SEGMENT

The TNC_IFM_NO_NEXT_SEGMENT error code is sent by a Contracted Party in response to a Next Segment attribute when there are no remaining segments associated with the Base Message ID given in that Next Segment attribute. This could be due to the Segmented Message Exchange having completed, having been cancelled, or because there never was a Segmented Message Exchange associated with that Base Message ID. Since Contracted Parties are not required to retain Base Message IDs after completion or cancellation of the associated Segmented Message Exchanges, the Contracted Party might not be able to distinguish between these causes and is not required to do so.

Both the sender and recipient of this IF-M Error SHOULD log this error.

The recipient of a No Next Segment error MUST treat the Segmented Message Exchange, specified in the Next Segment attribute that precipitated this error, as cancelled. (The sender of this error code already considers the exchange completed, cancelled, or non-existent.)

This error MUST NOT cancel any Segmentation Contract.

4.13.2.2 TNC_IFM_UNEXPECTED_SEGMENT

The TNC_IFM_UNEXPECTED_SEGMENT error code is sent by a Contracting Party in response to a Segment Envelope that is unexpected or that cannot be processed. This could be due to the Segmented Message Exchange having completed, having been cancelled, or because there never was a Segmented Message Exchange associated with that Base Message ID given in the Segment Envelope. Since Contracting Parties are not required to retain Base Message IDs after completion or

cancellation of the associated Segmented Message Exchanges, the Contracting Party might not be able to distinguish between these causes and is not required to do so.

This IF-M Error MUST NOT be sent in response to a Segment Envelope with the Start flag set, since such a Segment Envelope is initiating a new Segmented Message Exchange.

Both the sender and recipient of this IF-M Error SHOULD log this error.

The recipient of an Unexpected Segment error MUST treat the Segmented Message Exchange, specified in the Segment Envelope attribute that precipitated this error, as cancelled. (The sender of this error code already considers the exchange completed, cancelled, or non-existent.)

This error MUST NOT cancel any Segmentation Contract.

4.13.2.3 TNC_IFM_MESSAGE_NOT_FOUND

The TNC_IFM_MESSAGE_NOT_FOUND error code is sent by a Contracted Party in response to a Contract Exemption attribute, in the case that the Message Identifier given in that attribute is not associated with an oversized message awaiting delivery. This could occur because the Contracted Party has discarded the oversized message because it was already delivered, or because it timed out waiting for a Contract Exemption, or because the Contracted Party never had any oversized message with that Message Identifier. Because the Contracted Party is not required to retain information about an oversized message after that message has been discarded, it might not be able to distinguish between these causes and is not required to do so.

Both the sender and recipient of this IF-M Error SHOULD log this error.

This error MUST NOT cancel any Segmentation Contract.

4.13.2.4 TNC_IFM_SEG_VIOLATION

The TNC_IFM_SEG_VIOLATION error code is sent by either a Contracted or Contracting party in response to a Next Segment or Segment Envelope attribute, respectively, that violates the recipient's understanding of the Segmentation Contract. Causes of this error include:

- The recipient of a Next Segment attribute believes that it has no Segmentation Contract that supports Segmented Message Exchanges associated with the Next Segment attribute.
- The segment in a received Segment Envelope is larger than allowed by the active Segmentation Contract or is 0-length.
- The recipient of a Segment Envelope believes that it does not currently have a Segmentation Contract with the envelope sender that allows Segmented Message Exchanges.

This error code is only used when attributes associated with Segmented Message Exchange violate the attribute recipient's understanding of the Segmentation contract. In particular, the error code MUST NOT be sent by a Contracting Party upon receipt of a complete message (that is, outside of a Segmented Message Exchange) that exceeds the maximum segment size limit set out in the relevant Segmentation Contract. (In fact, Contracting Parties are required to accept such oversized messages without raising any error defined in the IF-M Segmentation specification, per section 3.5.1.)

Both the sender and recipient of this IF-M Error SHOULD log this error.

Both the sender and recipient of this error MUST treat the Segmented Message Exchange associated with the attribute that precipitated this error as cancelled. Moreover, both the sender and recipient of this error MUST treat any associated Segmentation Contract (i.e., one with the appropriate Contracting Party, Contracting Party, and IF-M Subtype) as cancelled.

5 Security Considerations

This specification adds relatively few security considerations. This is due to the fact that the practical impact of this specification is to make slight structural modifications to existing IF-M bindings, specifically by allowing them to be transmitted in segments. As such, IF-M Segmentation inherits the security considerations of the IF-M bindings that it conveys, but adds few new considerations. Implementers need to consult the security considerations of the IF-M binding that is to be supported by IF-M Segmentation and determine if any of those considerations affect or are affected by the IF-M Segmentation capabilities.

The following sections look at specific threats that can arise through the use of IF-M Segmentation and how to address them.

5.1 Impractical Size Limits

Implementers of IF-M Segmentation need to be aware of the possibility of a malicious Contracting Party requesting an extremely small maximum message size and/or maximum segment size. For example, a Contracting Party could specify a maximum message size of 64 bytes, making most of the Segmentation Contract's Contracted Party's messages too large to send. Alternately, a Contracting Party could specify a maximum segment size of 64 bytes, forcing the Segmentation Contract's Contracted Party to send many IF-TNCCS batches to convey even a small message, thus contributing to network congestion. As such, while the specification provides lower limits for both message and segment sizes that ensure that a Segmentation Contract cannot make communication impossible, permissible message and segment sizes can still be made impractical for most circumstances.

One way to avoid this problem is to ensure that no IF-M Component accepts a Segmentation Contract with impractical message or segment size limits. Implementers SHOULD allow enterprise administrators to specify lower size limits for messages and segments of acceptable Segmentation Contracts, where all Segmentation Contracts with smaller limits are rejected. Implementers SHOULD ensure that the default bounds are large enough not to cause problems. (10 MB would be a reasonable default as the smallest acceptable message size, and 1 MB would be a reasonable default as the smallest acceptable segment size.) Different environments can have different definitions of practical size limits, so implementers MUST allow administrators to change the lower size limits as necessary, providing it is not set below the minimum acceptable limit of 64 bytes. IF-M Components MUST NOT accept Segmentation Contracts that are smaller than the specified size limits.

Note that, since the Contracted Party is the one that actually sets the limits of a Segmentation Contract, a malicious Contracted Party could take a Segmentation Contract Request with reasonable size limits and then specify impractically small size limits in its Segmentation Contract Response, effectively creating the same outcome. However, this is not really a new attack, since a malicious Contracted Party could withhold messages as "oversized" with or without a Segmentation Contract.

5.2 Withholding Segments

During a Segmented Message Exchange, a malicious Contracted Party could send most of a segmented Base Message, but then withhold the final segments. The Contracting Party could then be left with most of a large message in memory but lacking the final segments that would allow it to process this message. This could cause resource depletion on the Contracting Party.

To address this possible threat, Contracting Parties SHOULD employ a timeout that detects when the Contracted Party has not responded to a Next Segment attribute within a reasonable amount of time. Given that a Contracted Party MUST respond immediately to a Next Segment attribute, a timeout of 30 seconds is probably a reasonable default, although implementers SHOULD allow administrators to tailor this value, since some network environments might make longer or shorter timeouts reasonable. If a timeout is employed, the Contracting Party MUST explicitly cancel the Segmented Message Exchange by sending another Next Segment attribute with the Cancel

Exchange flag set. This is the only situation under which a Contracting Party is permitted to send two consecutive Next Segment Attributes without receiving a Segment Envelope attribute between them.

6 Privacy Considerations

As with the security considerations, IF-M Segmentation inherits the privacy considerations of the IF-M bindings that it conveys, but it adds no privacy considerations of its own. In fact, IF-M Segmentation can help with privacy, by making large messages more difficult to spot if they are conveyed using Segmented Message Exchanges. This said, implementers need to consult the privacy considerations of the IF-M binding that is to be supported by IF-M Segmentation and determine if any of those considerations impact or are impacted by the IF-M Segmentation capabilities.

7 References

7.1 Normative References

[1] Trusted Computing Group, *TNC Architecture for Interoperability*, Specification Version 1.5, Revision 3, May 2012.

[2] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", Internet Engineering Task Force RFC 2119, March 1997.

[3] Trusted Computing Group, *TNC IF-M: TLV Binding*, Specification Version 1.0, March 2010.

7.2 Informative References

[4] Trusted Computing Group, *TNC IF-IMC*, Specification Version 1.3, February 2013.

[5] Trusted Computing Group, *TNC IF-IMV*, Specification Version 1.4, August 2013.

[6] Trusted Computing Group, *PTS Protocol: Binding to TNC IF-M*, Specification Version 1.0, August 2011.

[7] Trusted Computing Group, *SWID Message and Attributes for IF-M*, Specification Version 1.0, August 2015.

[8] Trusted Computing Group, *TNC IF-T: Binding to TLS*, Specification Version 2.0, February 2013.

[9] Trusted Computing Group, *TNC IF-T: Protocol Bindings for Tunneled EAP Methods*, Specification Version 1.1, May 2007.

[10] Trusted Computing Group, *TNC IF-TNCCS: TLV Binding*, Specification Version 2.0, January 2010.

[11] Sahita, R., S. Hanna, R. Hurst, K. Narayan, PB-TNC: A Posture Broker (PB) Protocol Compatible with Trusted Network Connect (TNC), RFC 5793, March 2010, IETF.