MMUSIC
Internet-Draft
Intended status: Standards Track
Expires: January 03, 2014

R. Penno, Ed.
P. Martinsen
D. Wing
A. Zamfir
Cisco

July 02, 2013

Meta-data Attribute signaLling with ICE
draft-martinsen-mmusic-malice-00

Abstract

It can be useful for applications to provide flow metadata information to on-path devices to influence flow treatment in the network. Provided that the network is able to provide useful feedback, this can also influence path selection if an application have multiple flow paths to choose from.

This draft describes how this can be achieved by adding metadata to the STUN packets sent during the ICE connectivity checks or a slightly modified version of the keep-alive mechanism. Devices on the media path can use the metadata information to prioritize the flow, perform traffic engineering, or provide network analytics and notifications as requested by the endpoints. On-path devices can append or modify the existing metadata information in the STUN/ICE messages to enable feedback to other on-path devices or the applications in both ends of the media session.

This document describes a framework mechanism for how such metadata can be transported by STUN when ICE is in use and it covers the endpoint and on path device processing. The functionality described here is referred to as MALICE.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

This Internet-Draft will expire on January 03, 2014.

Copyright Notice

Table of Contents

1. Problem Statement

In the context of Content, Mobile, Fixed Service, Service Providers, Enterprise and Private networks have a need to prioritize packet flows end-to-end. These flows are often dynamic, time-bound, encrypted, peer-to-peer, possibly asymmetric, and might have different priorities depending on network conditions, direction, time of the day, dynamic user preferences and other factors. These factors may be time variant, and thus need to be signalled. Moreover, in many cases of peer-to-peer communication, flow information is known only to the endpoint. These considerations, coupled with the trend to use encryption for browser-to-browser communication [I-D.ietf-rtcweb-security-arch], imply that access lists, deep packet inspection and other static prioritization methods cannot be employed successfully to prioritize packet flows. It can also be useful for the endpoints to provide flow metadata and receive network feedback in order select an optimal media communication path. This specification describes how these problems can be solved at

different points in the network by using either STUN [RFC5389] packets sent during ICE's [RFC5245] connectivity check phase during establishment of a media session, or as part a slightly modified keep-alive mechanism after the session is established. Devices on the media path can use the metadata information to prioritize the flow, perform traffic engineering, or provide network analytics and notifications as requested by the endpoints. On-path devices can append or modify the existing metadata information in the STUN/ICE messages. The ICE agents may use this information to learn about the status of their requests at on-path devices.

This document describes a framework mechanism for how such metadata can be transported by STUN when ICE is in use with UDP based media and it covers the endpoint and middlebox processing. The functionality described here is referred to as MALICE.

2. Terminology

Metadata - Information and actions associated with a flow but not used for matching. For example, firewall and NAT actions, application name, Diffserv marking actions, media-type, amongst others.

Flow - 5-tuple composed on source and destination IP addresses, IP protocol, source and destination ports.

MALICE Agent - An ICE agent [RFC5245] that supports this specification

MALICE Check - An ICE connectivity check that includes client metadata and that may include the results from network elements that have processed the request.

MALICE Message - An ICE connectivity check message (STUN Binding request or response) that carries metadata attributes.

Metadata Attribute - A STUN attribute that contains a set of information elements in the form of type-lenght-values (TLVs).

Information Elements - Information elements (IE) are TLVs that contain the actual metadata such as minimum bandwidth, delay tolerance, firewall action, etc.

Network Elements - Devices such as middleboxes, routers, Wireless Access LAN controller, amongst others. The terms network element and node are used interchangeably in the text.

3. Overview of MALICE

In a typical ICE deployment there are two endpoints, known as agents in ICE terminology, that attempt ICE message exchanges in order to discover one or more paths over which they can send and recieve media. The ICE exchange protocol is defined in [RFC5245]. This specification proposes an extension to the ICE protocol that allows applications to request services from the network, and learn about the status of these requests and of the media paths they use. This is achieved by signaling flow and network metadata attributes between endpoints and network elements (NEs).

The means by which an implementation determines the metadata IEs to be signaled is out of the scope of this specification. Section 9 covers different scenarios where metadata may be of use. This specification defines three types of transaction that can be signaled by a MALICE agent and acted upon by NEs.

- Binding Transaction (REQ-RESP): Endpoint requests flow prioritization, e.g. by signaling the desired service class (Section 9) that includes the minimum and maximum bandwidth, loss and delay tolerance. The following are examples of services that could be offered by network elements:
  - IntServ: Network elements on path may perform admission control against the desired service class. If resources are not available, a middlebox may return an error (or allocated BW = 0) or it may try to admit the flow in a lower service class. In the latter case, the middlebox will update the response with the new service class. If resources are available, they are allocated for the flow and guaranteed (in a stable network) for the lifetime of the flow.
  - DiffServ: A middlebox may perform flow classification. Flows are guaranteed QoS as long as there is no oversubscription. If the corresponding service queue becomes full, drops and delays affect all flows in that service class.
- Advisory Transaction (REQ-RESP):
  - Notification Subscription: An endpoint may request the network to send notifications when certain conditions occur. One example described in Section 9 is notification when congestion is about to occur in the class of service associated with the flow. Other services in this category may be defined in the future.

    * Query : Endpoints may request information from the network. One example described in Section 9 is an endpoint requesting the currently available bandwidth, delay and loss tolerance of the service class associated with the flow. Network elements update the response STUN attributes if local values are more restrictive than the ones carried in the message. At the end of the request/response check, the endpoint has the information about the end-to-end b/w, delay and loss characteristics of the path.

- Informational Transaction (INFO-ONLY):

    * Endpoints send INFO-ONLY attributes to describe their flows. This service can be used in managed environments like enterprise or data center.

The following new comprehensive-optional STUN attributes are defined in order to support this functionality:

- MD-AGENT: includes client agent metadata information for the flow described by the 5-tuple identified in the STUN/ICE header.

- MD-RES-UP: contains the result of the request processing by the network elements on upstream path.

- MD-RES-DN: includes the result of the request processing by the network elements on downstream path.

- MD-PEER-CHECK-RES: contains the result of the MALICE check performed by the peer agent.

- MD-INFO: contains flow descriptive information.

The client agent includes a combination of MD-AGENT, MD-RESP-UP and MD-RESP-DN to create one of the three transaction types described above. In addition, the FLOWDATA sub-TLV is defined to support flow prioritization through a Binding Transaction.

3.1. Metadata Attributes

The main focus of this specification is around the services described in the previous section which are implemented through REQ-RESP attribute signaling. For these services, most of the actions described here apply.

3.1.1. Sending and Receiving

Sending metadata can be done early in the connectivity check phase of ICE [RFC5245] section-7 and the result of metadata processing may be taken into account by the controlling agent during the nomination process. Once a candidate pair is selected to be used for media, MALICE agents use the consent freshness mechanism described in [I-D.muthu-behave-consent-freshness] to signal metadata attributes.

If a server agent supports MALICE, it MUST reflect back in the STUN Binding Response message the metadata attributes that were received in the STUN Binding Request. It is up to the server agent whether to use the metadata present in the binding request for its own purposes, for example adjusting the metadata it will put in its own binding request.

Network Elements on the path that are MALICE capable may intercept and read the metadata attributes from the connectivity or consent freshness checks. They may also update the message with the result of a REQ-RESP request. When doing so, the NEs MUST NOT add significant delay while attribute processing is in progress and SHOULD wait for the next refresh message for result update.

3.1.2. Directionality and Asymmetry

It is important to mention that some attributes may be bidirectional in nature, while others may be associated with a given direction. A bi-directional attribute is represented by individual upstream and downstream attributes.

In order to take into account directionality and routing asymmetry the following rules are proposed for the STUN Binding request/ response messages used in connectivity check and consent freshness mechanism:

STUN Request On-path devices only process upstream attributes and if necessary update the original request message with the result.

STUN Response On-path devices only process downstream attributes and if necessary update the original response message with the result.

Due to asymmetric routing, a NE may see only binding request or response messages for a given candidate pair and therefore it may read and process metadata for upstream only, downstream only or both. In some cases, upstream and downstream paths may span the same node but over different interfaces and in this case a middlebox may need to use different ingress and/or egress interface policies for the two directions of the media.

3.1.3. Network Element Processing

When processing MALICE messages, NEs generally perform the following steps:

1. Intercept and read the metadata attributes from the connectivity or consent freshness checks.

2. Depending on the metadata information elements carried in the message and on the current state (e.g. resource availability, policies, etc.), a node may perform certain actions (e.g. install local policies for the flow described by the message, start monitoring the flow, perform marking, etc.).

3. If the results of these actions are readily available, the network element should include them in the currently intercepted message. Otherwise any required response is conveyed in the next refresh message.

4. Forwards the MALICE message downstream.

The current specification makes sure that network elements do not have to change the STUN message size, instead the MD-RESP-* attributes are inserted as place holders for updates from network.

3.1.4. MALICE Client and Server Processing

The MALICE client agent includes metadata information elements in the new MD-AGENT STUN attribute defined in this specification. The MD-AGENT attribute MUST be included before INTEGRITY. If a response is required for all or a subset of these information elements, the client agent may also include the new MD-RESP-DN (before INTEGRITY) and MD-RESP-UP (after INTEGRITY) as place holders that can be used by on-path devices to provide a response.

When a MALICE server agent receives a Binding Request, it copies the MD-AGENT and the MD-RESP-UP TLV in the response, adds the INTEGRITY attribute and then inserts the MD-RESP-DN attribute to be filled by on path nodes for the downstream direction. When forming the response (success or error), the agent running the server follows the rules of Section 6 of [RFC5389]. It MUST NOT send an ’Error Response’ message class if the processing of metadata attributes is the only one that has failed. Instead the MALICE error indications are included in the MD-RESP-UP to communicate to the client the success/error indications for the metadata processing.

3.2. Connectivity Checks

Connectivity checks are extended by this specification to include metadata attributes in both request and response messages. In the presence of REQ-RESP metadata attributes, a MALICE agent may consider the connectivity check successful if responses for the check received indicate success. It is not necessary that the metadata attribute results, if present, also indicate success.

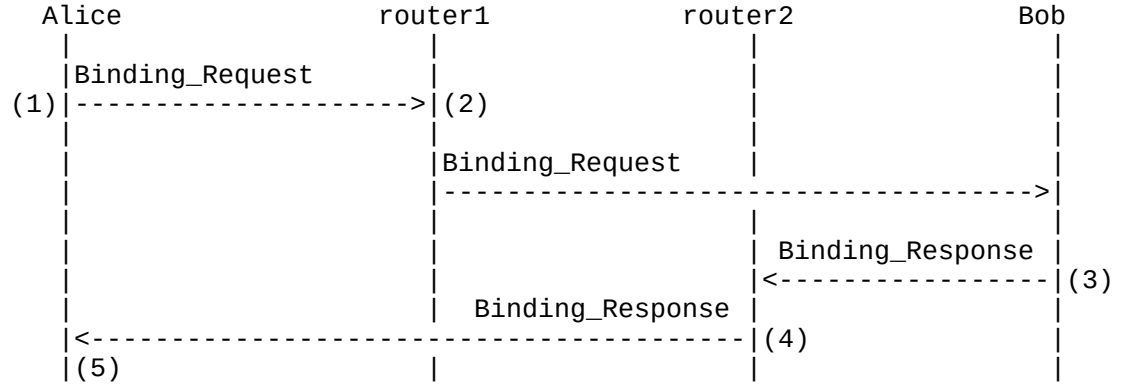
The MALICE Server agent MAY also include the new MD-PEER-CHECK-RES TLV defined in this specification if it has already performed a MALICE check and has the result available. This is useful if the MALICE Server is the controlled agent and wishes to influence the nomination process at MALICE Client (controlling agent).

3.2.1. MALICE to non-MALICE

A MALICE client agent does not have prior knowledge if the peer supports this specification. If the peer agent is not MALICE capable, it will not reflect back the metadata STUN attributes. Therefore a MALICE client agent will know if peer is MALICE capable after the first exchange of the connectivity check. The client may choose to continue to signal the metadata attributes to benefit from possible upstream network element processing but should not expect any results from the network.

3.2.2. MALICE to MALICE

A remote MALICE agent echoes back in the Binding Response message all metadata received in the request. In the example below MALICE upstream network elements (router1 in the diagram below) processes MD-AGENT and MD-RESP-UP attributes present in the STUN binding request while MD-AGENT and MD-RESP-DOWN attributes present in the STUN binding response are processed by network elements (router2) in the downstream path.



FLOW-METADATA MALICE to MALICE

1. Alice creates a Binding Request, adds MD-AGENT and result (MD-RESP-UP and MD-RESP-DN) attributes with desired metadata information elements.

2. Router1 inspects the Request message and, if allowed (based on realm, security and policy considerations), reads MD-AGENT attribute and its information elements. If the result of processing is available, router1 writes the result in the MD-RESP-UP attribute. It then forwards the request.

3. Bob processes the Binding Request as described in the ICE RFC [RFC5245](Section 7.2). When Bob builds the response, it copies the metadata attribute MD-AGENT and the MD-RESP-UP attributes into the Binding Response and adds MD-RESP-DN after the integrity attribute. Bob then transmits the message.

4. Router2 (first MALICE network element for the downstream direction) inspects the Response message, reads the metadata attribute and MAY change the result (MD-RESP-DN) including the local results if available. It then transmits the message.

5. When Alice receives the Binding Response message, the same processing described in ICE RFC [RFC5245] (Section 7.1.3) applies. Then it extracts the metadata upstream and downstream attributes. If Alice’s agent has the controlling role, it may take into account this information during the candidate pair selection step (if this check was part of the initial connectivity check sequence).

3.3. Keepalives

This specification proposes the use of consent freshness messages [I-D.muthu-behave-consent-freshness] in place of indications in order to have up to date results on the MALICE checks used by media. This is required since network conditions may change during the lifetime of a flow resulting in changes, including new failure indications, in MALICE responses.

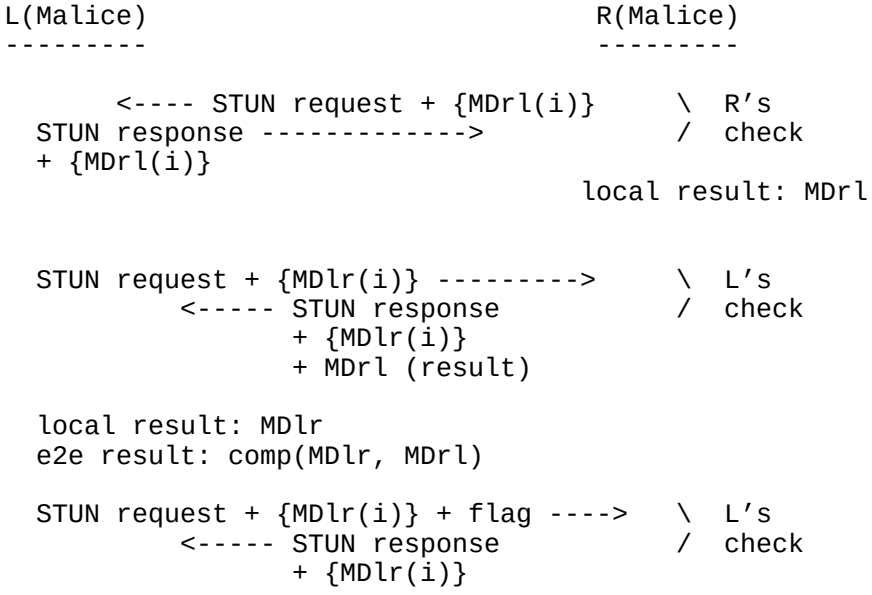
3.4. Aggressive Nomination

With aggressive nomination, the controlling agent includes the nominated flag in every connectivity check it sends for all media components. Once the first check for a component succeeds, it is added to the valid list with the nominated flag set. The nominated candidate pair may start being used by the media at any time after. This lowers the chance of MALICE results to be collected. Therefore,

if the controlling MALICE agent expects to consider the metadata attribute processing result into the candidate pair selection process, it SHOULD NOT use aggressive nomination. The controlled MALICE agent does not have a way to influence the peer with respect to the nomination procedure used. If the peer is non-MALICE, the agent SHOULD NOT signal any MD attributes. If a MALICE agent chooses to use the aggressive nomination, the endpoints should be prepared for transient candidate selection as described in Section 8.1.1.2 of [RFC5245]. Using aggressive nomination is an implementation trade-off between quick call initiation versus waiting to determine the best path (using regular nomination and waiting until MALICE checks finish).

3.5. Implications on Concluding ICE

When the MALICE client agent receives the STUN binding response it extracts the metadata results. A controlling agent may choose to ignore the received metadata information or consider it in the decision process. The figure below shows MALICE used in a regular nomination process.



Notations:

L is the controlling agent.

{MDrl(i)} is the set of metadata attributes sent from R to L in the request. In the (2nd, 3rd,..) response back they will also include the result. Similar notation for the checks in the other direction.

MDrl is a an overall success/fail type of indication for the MALICE check R->L

comp(MDlr, MDrl) - is a function that determines the overall end to end MALICE result based on both local check result and the one from the peer.

If a connectivity check response is received for an already nominated pair, the controlling agent may inform the application but MUST NOT restart the nomination process. In the case where the result of a MALICE check is not available in the response at the time of nomination, any subsequent MALICE results become informative.

3.6. Lite Implementations and MALICE

As described in [RFC5245], lite ICE implementations do not send connectivity checks but only reply to them. A lite ICE implementation may be extended to become a lite MALICE implementation by adding the functionality associated with the MALICE Server. When a lite MALICE server agent receives a STUN binding request, it copies the metadata related attributes as described in earlier sections. A lite MALICE implementation will never include an MD-PEER-CHECK-RES attribute in the STUN binding response, since it never runs ICE or MALICE checks.

4. Performing Connectivity Checks

This section describes how MALICE agents perform connectivity checks and how network elements process and modify the information in the connectivity check messages.

4.1. MALICE Client Procedures

4.1.1. Building the MALICE Request

This section describes how STUN and ICE are extended to include metadata attributes and refers to them in generic terms. The new attributes and their usage defined in Section 9 are included in the connectivity checks performed by MALICE agents.

The Client agent starts the connectivity check by sending a STUN binding request following the procedures described in Section 7.1.2 of [RFC5245]. A MALICE client MAY include metadata attributes in the request. The way the application determines the attributes to be

sent to the MALICE agent for signaling is outside the scope of this specification. The client agent may reduce the attribute set based on other factors (e.g. MTU considerations).

The client encodes metadata information in the MD-AGENT attribute. It then builds the MD-RESP-UP and MD-RESP-DN attributes, including an information element for each REQ-RESP attribute for which a response is desired. The values in these IEs are initialized as described in the corresponding metadata information element section. MD-AGENT and MD-RESP-DN MUST be included before INTEGRITY, and MD-RESP-UP after INTEGRITY so that it can be changed by on-path devices.

4.1.2. Processing MALICE Responses

A MALICE agent processes a STUN binding response and depending on the presence of metadata attributes, their contents, and the procedures of [RFC5245] section 7.1.3.1 the result of MALICE connectivity check is considered unknown, failure or success as described below

4.1.2.1. Unknown

If the STUN response message does not include any metadata related STUN attributes, this is an indication that the peer is not MALICE capable. In this case the client should change the pair state to Succeeded.

It is possible that the STUN Client receives a response that includes metadata STUN attributes, but doesn’t include any valid results from NEs or STUN Server. This can happen if NEs are not MALICE enabled.

4.1.2.2. Failure

In the presence of a MALICE peer, a MALICE check is considered failed if either of the following is true:

- the ICE check has failed as described in Section 7.1.3.1 of [RFC5245].
- the client determines that the metadata included by an on-path device in the Binding response does not meet its criteria for success. The success criteria is application dependent and outside the scope of this specification.

4.1.2.3. Success

A MALICE check is considered successful if all of the following are true:

- the ICE check as described in Section 7.1.3.1 of [RFC5245] has succeeded.
- the Binding response indicates that MALICE NEs have satisfactorily processed all the RESP-REQ information elements.

4.2. MALICE Network Element Procedures

A MALICE network element intercepts ICE request and response messages, reads metadata information from the MD-AGENT attribute and triggers corresponding processing. When the result of this processing is available, the MALICE node MAY update the MD-RESP-xx attribute carried in the message. As a consequence, it is recommended (and stated [RFC5245]) that the agent perform a few identical checks in order to allow NEs to react to and communicate the result of the metadata processing.

MALICE NEs consume router resources to maintain per flow state and, depending on the information elements and requests, to enforce per flow QoS or perform monitoring. State and associated attributes are considered alive as long as periodic refresh messages that include those attributes are received. In the absence of refreshes [I-D.muthu-behave-consent-freshness] or if attributes cease to be present in those refreshes, attributes time out, associated resources are released and state may be removed.

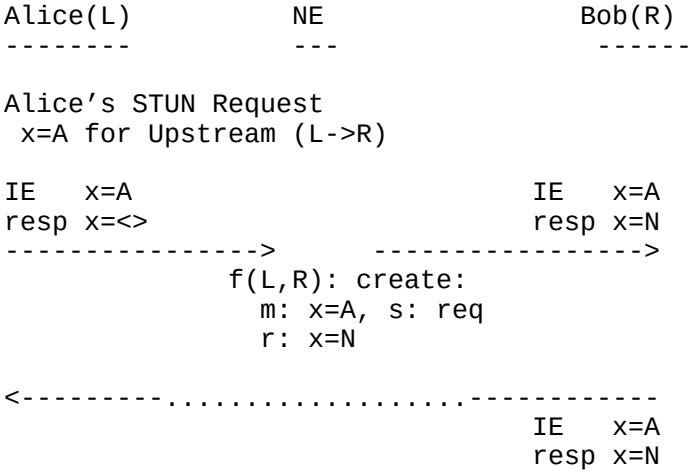
MALICE agents can signal the same metadata information elements for a flow. Therefore it is possible that different STUN messages types containing the same information elements, with same or different values, are seen by NEs. It is also possible that the two agents signal different metadata for the same flow.

During the lifetime of a session, agents can change the values of information elements, remove or add new IEs. It is also possible that a NE changes the result values over the lifetime of a session. A NE should determine if a newly intercepted STUN message indicates a refresh versus a change as compared to the previously intercepted message. A refresh resets the lifetime of an IE and state. A change indicates if new IEs are being created or if existing ones are being modified or removed.

4.2.1. Adding a new Metadata IE

When a new IE is signaled in a STUN message, a network element should create state for the flow if not already present, and trigger any required processing. If the network element, while processing the metadata attribute, will add significant delay and cause timeouts in the agent state machines, it is recommended that it forwards the STUN message and use the next refresh message to provide the results. When the next STUN message is received, the NE should provide the result of processing this information element only if the locally stored (and acted upon) value is the same as the one in the newly received message. Otherwise a removal or modification has occurred.

The diagram below illustrates the exchange and processing when a new IE is added. Alice sends a STUN request upstream with attribute MD-RESP-UP, MD-AGENT and IE X=A. The network element creates the f(L,R) state where it stores the requested metadata value (m: X=A), the context it was received from (s: MALICE request) and the result of processing (r: x=N). It then updates the response attribute MD-RESP-UP in the STUN request with X=N and forwards it to Bob. Bob reflects back the original metadata requested value and the result.



Upstream Attribute Initial Signaling

Similar processing happens for downstream attributes except that the NE's actions (intercept, flow state creation, etc.) happen when a STUN response is intercepted.

There are many possible transaction types for "X=A". For example:

- Endpoint requests a particular service: "Reserve BW=5Mbps", the endpoint requests a 5Mbps reservation.

- Endpoint requests network notification: "Notify if BW < 5Mbps", the endpoint requires a notification when the queue capacity used for this flow falls below the 5Mbps limit.

- Endpoint request statistics for the flow path: "BW=<>", where <> is the unspecified value for attribute BW, the endpoint requires a response with the current available queue capacity used for this flow.

It is assumed in the rest of this specification that the attribute, information element and/or context unambiguously identify the actions required at network element.

4.2.2. Removing a Metadata IE

Flow state and all its metadata ages out and should be removed when the state has not been refreshed recently by a request or response message. The way to determine the timeout interval is described in [I-D.muthu-behave-consent-freshness].

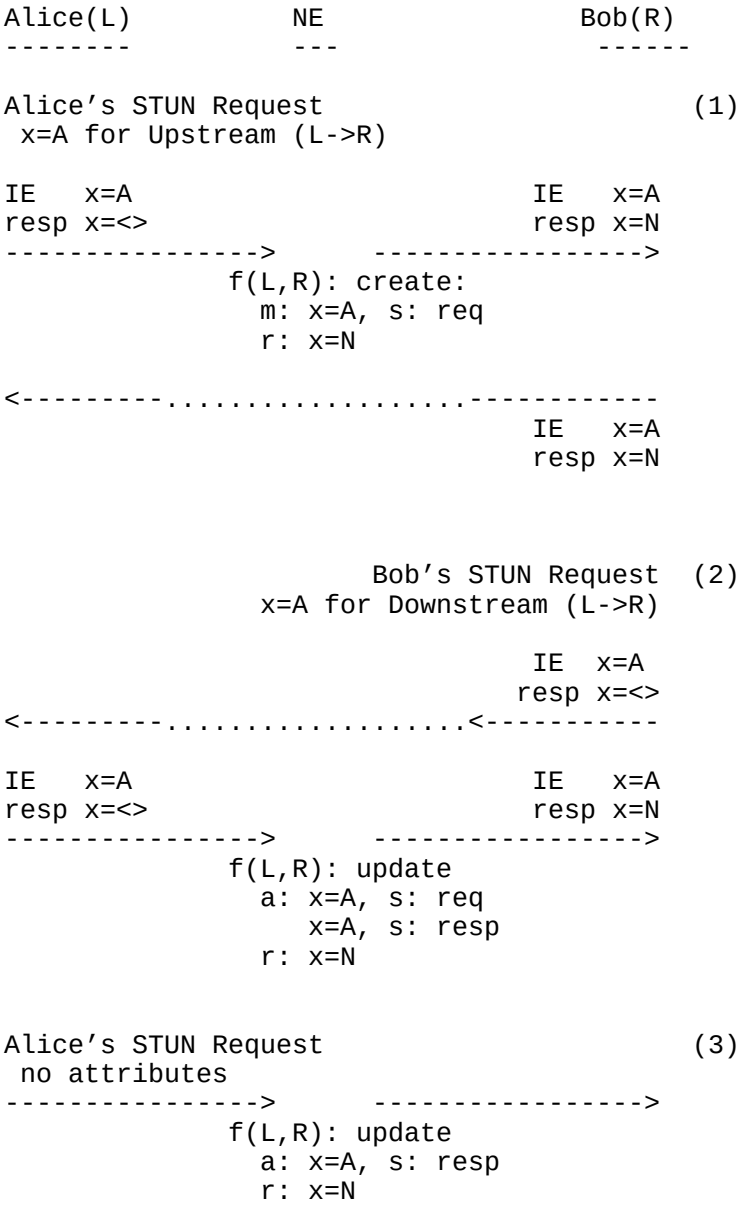
In addition, metadata must be immediately deleted and associated resources released if the IE is not present in any subsequent messages for the flow. An IE should be considered stale and removed if it ceases to appear in STUN requests or responses (section 3.1.2) having the same 5-tuple flow. As illustrated in the diagram below, a NE implementation should keep track of the source and value of the IEs received and detect per source addition, change and removal. More details are provided in the next sections. In the diagram below Bob’s messages do not go through the NE element:

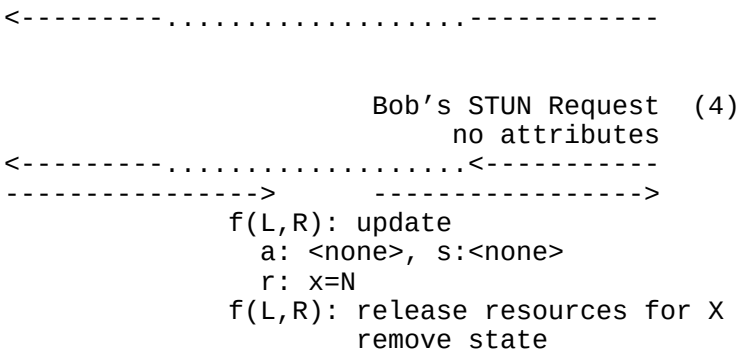
1. Alice signals metadata X=A for the first time. Actions are described in the previous section.

2. Bob signals the same value and equivalent direction for X and in his STUN request, this is copied in the STUN Response from Alice to Bob. When the NE intercepts this L->R response message, it extracts X=A, retrieves the existing information f(L,R) and adds MALICE Response as a new source.

3. Alice sends a new check without any metadata attributes. The NE retrieves the f(L,R) state and removes the MALICE Request from the source list. The flow state is maintained as the NE still sees refreshes for X in the L->R responses to Bob’s checks.

4. Bob sends a new STUN connectivity check without any attributes. The NE retrieves the f(L,R) state and removes the MALICE Response from the source list. Since X has no source, it also removes X from the metadata information element list and releases any resources associated with X. And because the flow state has no more attributes, it also removes the state.



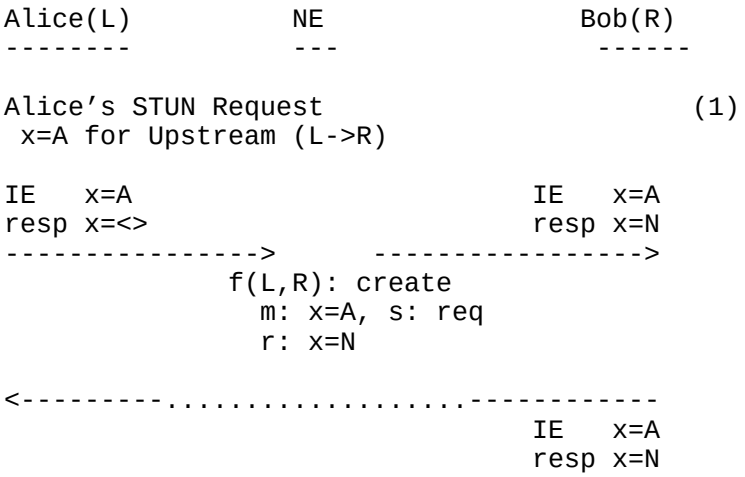


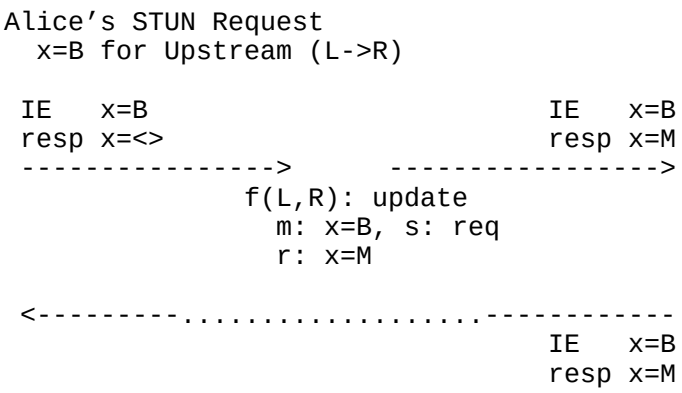
Upstream Attribute Removal

4.2.3. Changing a metadata IE

It is possible for a client to change an IE value. Every request/response message contains an MD-RESP-xx attribute with "not specified" values when sent from the agent. In other words, the agent does not include the result from previous check. When a node detects a change in an attribute value it should trigger the appropriate actions. Like in the case of initial attribute creation, the node should provide the answer in the next refresh message if the answer is not immediately available.

In the diagram below, Alice changes the value of information element X from A to B in the second STUN request which causes the network element to provide a different response.





Upstream Attribute Change

4.2.4. Network Element Response Change

It is possible that the network element result of processing of an IE changes as resource availability changes, e.g. new links are added and removed, new flows come and go, etc. For example, a NE can change the bandwidth available for a flow and may need to update the MD-RESP-xx attribute if the local value is more restrictive (e.g. less bandwidth, lower delay tolerance, etc.) than the one included in the message. Again, it is important for this node to check that the MD-AGENT attribute includes the same attribute and value for which the answer is provided.

4.2.5. Solving Conflicts in Metadata Attribute Values

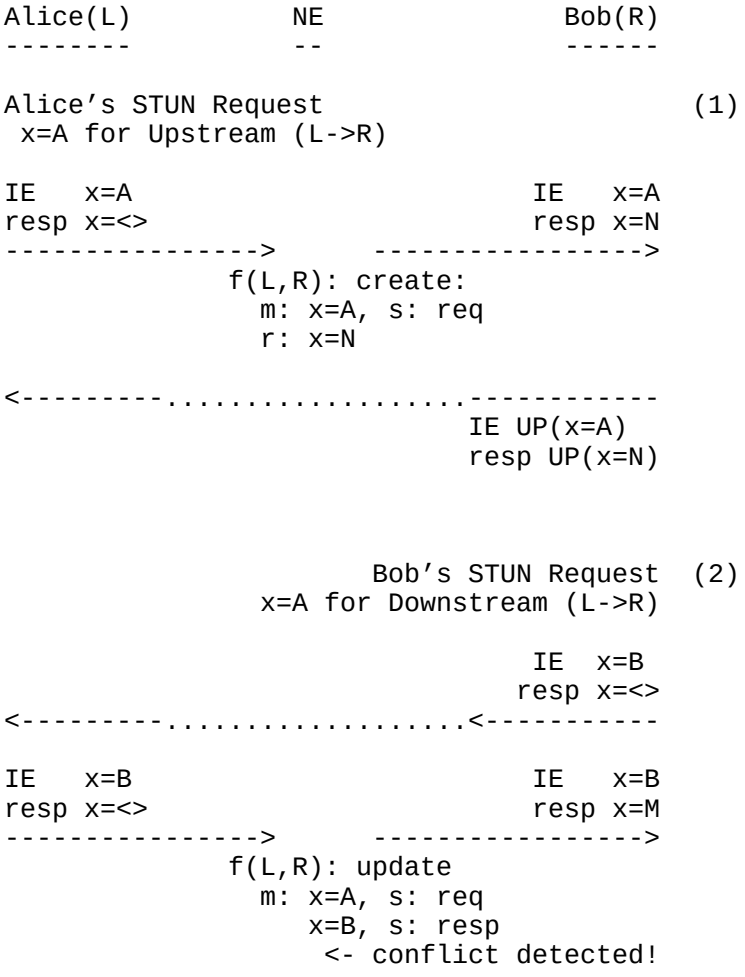
A conflict in a metadata information element occurs when the two agents signal different values for same IE and for the same direction of the flow.

A conflict occurs for an IE X in the upstream direction if the values of X in the L check request are different than in the R check response. When a NE detects an IE conflict it SHOULD keep both values. If the IE is part of binding request, the MALICE node must perform conflict resolution as described in the diagram below and act on the result.

1. Alice sends a request for X with value A for the upstream direction. The NE intercepts the message, creates f(L,R) state and stores X=A remembering this was received in Alice's request. The NE then determines that the response to A should be N, therefore it updates the STUN message and forwards it to Bob.

2. Bob sends a request for X with value B for the upstream direction. The NE intercepts the response for the Bob->Alice request, extracts X=B from the response, looks up f(L,R) flow state, stores (x=B, s:resp) and determines that a conflict has occurred for attribute X since (x=A, s: req) is present in the state. The NE runs the conflict resolution and determines that x=B should be the value used, determines that the result of processing B is M, updates the STUN response and forwards the response to Bob.

3. When the next refresh for X with value A is received from Alice, the NE updates the result to M and forwards the request to Bob. Bob reflects back the result in the response and Alice receives the changed result.



```
                        <- resolution x=B
                   r: x=M


   Alice's STUN Request                          (3)
    x=A for Upstream (L->R)

   IE   x=A                          IE   x=A
   resp x=<>                         resp x=M
   ---------------->      ----------------->
                 f(L,R): refresh:
                   m: x=A, s: req
                      x=B, s: resp
                   r: x=M

   <---------.....................................
                              attr UP(x=A)
                              resp UP(x=M)
```

Upstream Attribute Conflict

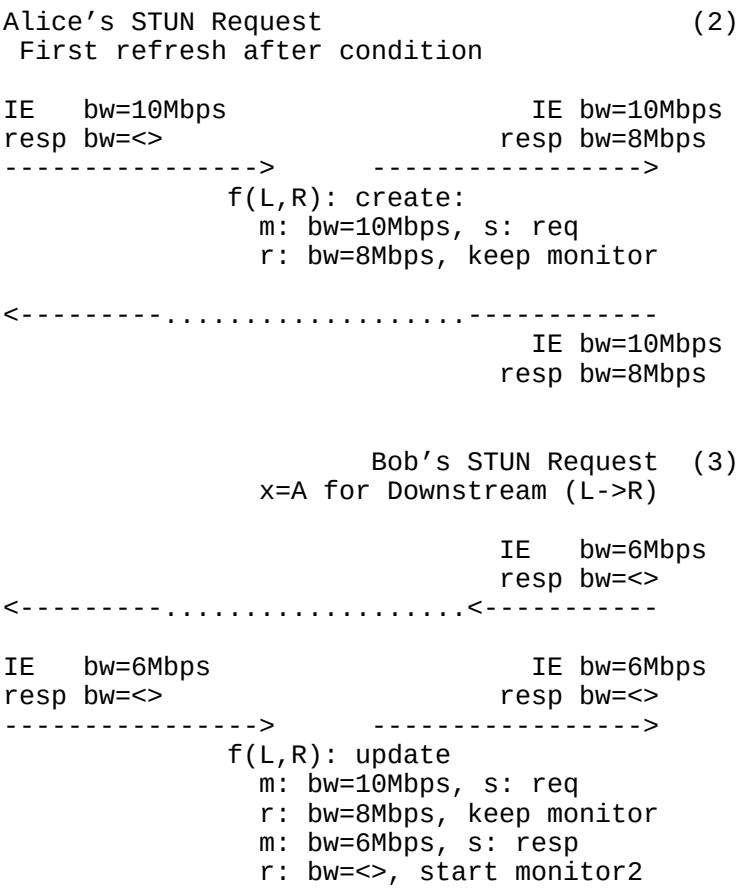
Note that for INFO-ONLY and ADVISORY transactions a conflict resolution cannot occur and, therefore, results should be kept per source. Typical NE resources allocated for these attributes are monitors created to detect conditions or collect network statistics. It is up to the implementation to decide on what can be shared in terms of resources in this case. In the diagram below, for illustration purposes, a second monitor is created for Bob's notification request.

```
   Alice(L)          Mid               Bob(R)
   --------          ---               ------

   Alice's STUN Request                         (1)
    Notif for UP BW < 10Mbps

   IE   bw=10Mbps                   IE  bw=10M
   resp bw=<>                      resp bw=<>
   ---------------->      ----------------->
                 f(L,R): create:
                   m: bw=10M, s: req
                   r: bw=<>, start monitor

   <---------.....................................
                               attr bw=10M
                               resp bw=<>
```



4.2.6. Conflict Resolution

The definition/description of an information element must include a description of how conflict resolution should be done by network elements. Below are a few examples:

- Informational only transactions: the IEs included are signaled in the upstream direction only and they are processed by middleboxes on path with the STUN request. They should never generate conflicts.

- Binding transactions (QoS): the following attributes are currently defined:

* Bandwidth: UP/DOWN Max Bandwidth, UP/DOWN Min Bandwidth

* Service Class: UP/DOWN Delay, Loss and Jitter tolerance - specified as: 0=undefined, 1=very low, 2=low, 3=medium, 4=high

* Priority: UP/DOWN DSCP

For all these attributes the conflicts are resolved by choosing the less strict values (apply a MIN function). For example, assume Alice and Bob request the same service class. If Alice requests 10Mbps UP bandwidth, Bob requests 5Mbps DOWN bandwidth and there are 7Mbps available for the service class specified in the request, the middlebox should allocate 5Mbps and update the result in Alice's check STUN Response. If Alice and Bob request different service classes, the less restrictive is first selected and then the MIN function is applied to the bandwidth values.

o Advisory transactions (Network Analytics): there should not be any conflict resolution applied to these attributes. It is perfectly valid for Alice to request different network analytics than Bob or different thresholds for congestion notifications. As shown in the previous diagram, middleboxes should keep track of the different sources for a given attribute and, in case of network attributes, keep per source results and maybe resources.

4.3. MALICE Server Procedures

When the Malice Server agent receives a STUN Request it follows the same rules described in Section 7.2 of [RFC5245]. In addition, when building the STUN Response the following rules MUST be followed:

o MD-AGENT and MD-RESP-UP attributes are inserted before INTEGRITY

o If the result of the local MALICE check is present, an MD-PEER-CHECK-RES attribute with the result is included before INTEGRITY

o A copy of the MD-RESP-DN attribute received in the STUN Request is included unmodified after INTEGRITY

5. Concluding MALICE Processing

A MALICE Controlling agent is expected to run regular nomination only. This specification also reinforces the recommendation to run a number of checks before nominating a pair. This increases the probability of receiving network element and peer MALICE responses and therefore having more information for the nomination process.

When nominating a pair, the controlling agent may consider the MALICE information received in the last STUN Response and give preference to the pair whose connectivity check indicated favorable network conditions.

6. Subsequent Connectivity Checks

It is possible for a MALICE Client to request a service and include metadata attributes after the nomination process. It is also possible that a successful MALICE check for the nominated (active) pair fails during the media session lifetime. The MALICE Client will have at all times the current status of the MALICE check for the active pair. The actions that the client takes when these change are currently out of the scope of this document. In the absence of support for other specification, these MALICE check status changes are informative only.

7. Security Considerations

7.1. STUN Inspection

Network elements processing STUN packets are open to denial of service attacks from endpoints when there is no previous authorization and indication of which STUN messages should be inspected. The vulnerability and attack vector is similar to those documented for the IP router alert option in [RFC6398].

Flooding a NE with bogus (or simply undesired) STUN messages that contain metadata could impact its operation in undesirable ways. For example, if the NE punts the datagrams containing STUN messages to the slow path, such an attack could consume a significant share of the NE’s slow path and could also lead to packet drops in the slow path (affecting operation of all other applications and protocols operating in the slow path), thereby resulting in a denial of service (DoS) [RFC4732]. Like with other protocols, it is recommended that network elements that implement this functionality use rate limited queues when punting STUN messages. In addition, it is recommended that the implementation enforces limits on the number of states created by the MALICE connectivity checks.

However, the main issue is that the STUN message does not provide a convenient universal mechanism to accurately and reliably distinguish between interesting and unwanted messages. This, in turn, creates a security concern when the STUN metadata attribute is used, because, short of appropriate network element- implementation-specific mechanisms, the NE slow path is at risk of being flooded by unwanted traffic.

One solution to this problem is to include a precursor authorization step where a third-party device authorizes the endpoint and populates the NE with 5-tuple information of the packet carrying the STUN message. [TODO: Reference third party authorization draft]

7.2. Authentication

While endpoints are able to authenticate STUN messages received by a peer endpoint, network elements are unable to authenticate STUN messages. Further, endpoints are not fully trusted by network elements, so network elements need some assurance that what is signaled has been authorized by an application server that defines policies or attributes for a given media flow. Even if an endpoint is well-behaved, the network elements need a means of ensuring STUN messages are not altered during transmission.

8. STUN Extensions

8.1. New Attributes

This specification defines five new attributes, MD-AGENT, MD-REALM, MD-RESP-UP, MD-RESP-DN and MD-PEER-CHECK.

- The MD-AGENT is inserted in the Binding request by the client agent and copied in the Binding response by the server agent. It includes the flow metadata generated by the client agent.

- The MD-RESP-UP is inserted by the client agent in the Binding request and updated by MALICE nodes on upstream path. A MALICE server agent copies this attribute in the response message.

- The MD-PEER-CHECK attribute is inserted by the MALICE server agent in the response message and includes the result of the MALICE check executed by the server agent.

- The MD-RESP-DN is inserted by the client agent in the Binding request, copied by the MALICE server agent in the response and updated by MALICE nodes on downstream path.

In addition, two new sub-TLVs are defined to provide flow prioritization service. This specification allows for easy addition of IEs in the future.

- FLOWDATA Request sub-TLV is included in the MD-AGENT STUN attribute and indicates the desired flow treatment

- o FLOWDATA Response sub-TLV is included in the MD-RESP-* STUN attributes and indicates, when received by the client in the STUN Binding Response, the result of the processing

9. IANA Considerations

This specification registers five new STUN attributes. All attributes include metadata informational elements. Section 10.2 describes a possible STUN specific encoding for these. Another proposal can be found in [I-D.draft-flow-metadata-encoding] and [I-D .draft-flow-metadata-framework]

9.1. STUN Attribute TLV Definitions

This section registers four new STUN attributes per the procedures in [RFC5389].

0x0C02: MD-AGENT
0x0C03: MD-RESP-UP
0x0C04: MD-RESP-DN
0x0C05: MD-PEER-CHECK

9.1.1. MD-AGENT Attribute

Metadata attributes are encoded in sub-TLV format with each sub-TLV corresponding to an information element or metadata. Section 10.3 describes in detail the information elements that can be included in the MD-AGENT attribute. When parsing the STUN request and response, the MD-AGENT STUN attribute Length should be used to identify the location of next STUN attribute.

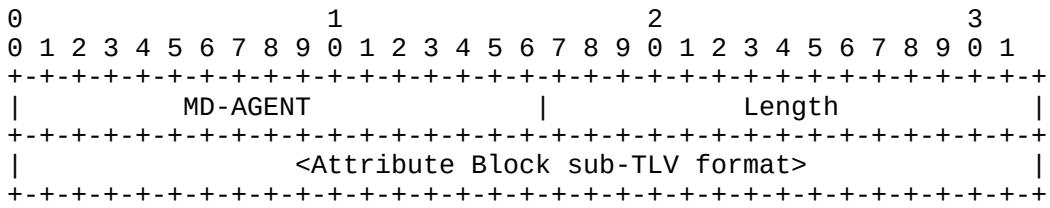


Figure 1: MD-AGENT Attribute

9.1.2. MD-RESP-UP and MD-RESP-DN Attributes

Network Metadata attributes are encoded in sub-TLV format with each sub-TLV corresponding to an information element or metadata.

Section 10.3 describes in detail the network information elements that can be included. When parsing the STUN request and response, the MD-RESP-XX STUN attribute Length should be used to identify the location of next STUN attribute.

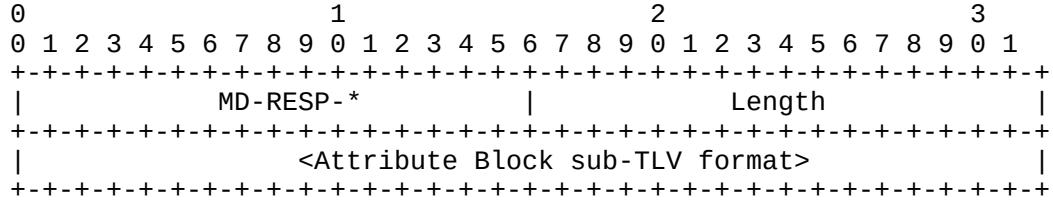


Figure 2: MD-RESP- Attribute

Where MD-RESP-* = {MD-RESP-UP | MD-RESP-DN}

9.1.3. MD-PEER-CHECK Attribute

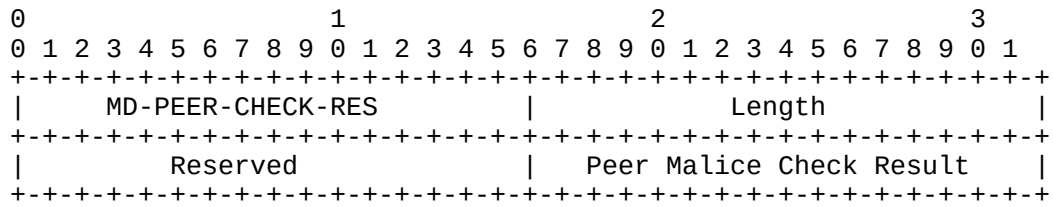


Figure 3: MD-PEER-CHECK Attribute

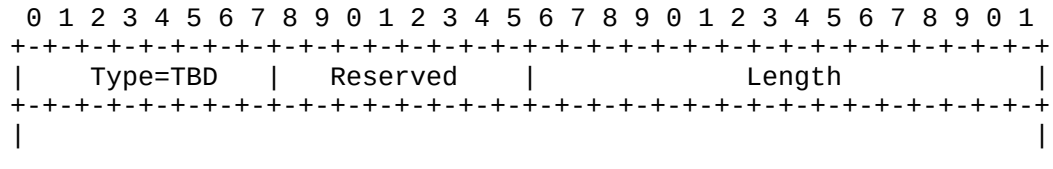
Peer Malice Check Result - "Success" or "Failure".

9.2. Metadata Attributes sub-TLV Definitions

Metadata information elements are encoded in sub-TLV format and included in MD-AGENT and MD-RESP-* STUN attributes described earlier.

9.2.1. FLOWDATA Request

The FLOWDATA IE has the following format.



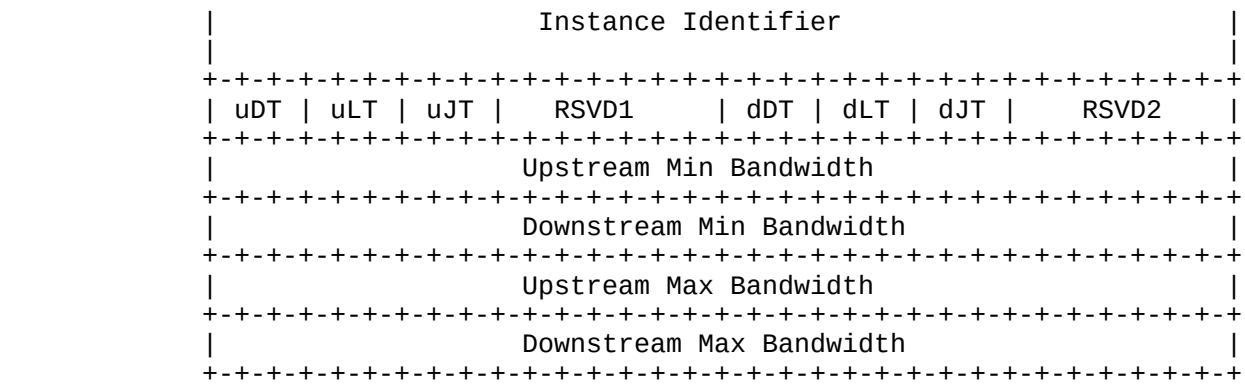


Figure 4: FLOWDATA Request

Type: TBD (optional to process)

Reserved: Must be 0 and ignored by the server.

Length: Option Length is 32 octets.

May appear in: STUN/ICE Binding Request and Response, inside the MD-AGENT STUN attribute

Maximum occurrences: 1

Description of the fields:

Instance Identifier: Instance identifier, see below for description.

uDT: Upstream Delay Tolerance, 0 means no information is available. 1=very low, 2=low, 3=medium, 4=high.

uLT: Upstream Loss Tolerance, 0 means no information is available. 1=very low, 2=low, 3=medium, 4=high.

uJT: Upstream Jitter Tolerance, 0 means no information is available. 1=very low, 2=low, 3=medium, 4=high.

RSVD1: Reserved (7 bits), MUST be ignored on reception and MUST be 0 on transmission

dDT: Downstream Delay Tolerance, 0 means no information is available. 1=very low, 2=low, 3=medium, 4=high.

dLT: Downstream Loss Tolerance, 0 means no information is available. 1=very low, 2=low, 3=medium, 4=high.

dJT: Downstream Jitter Tolerance, 0 means no information available. 1=very low, 2=low, 3=medium, 4=high.

RSVD2: Reserved (7 bits), MUST be ignored on reception and MUST be 0 on transmission.

Upstream Minimum Bandwidth Minimum Upstream bandwidth in bytes per second, 0 means no information is available.

Downstream Minimum Bandwidth: Minimum Downstream bandwidth in bytes per second, 0 means no information is available.

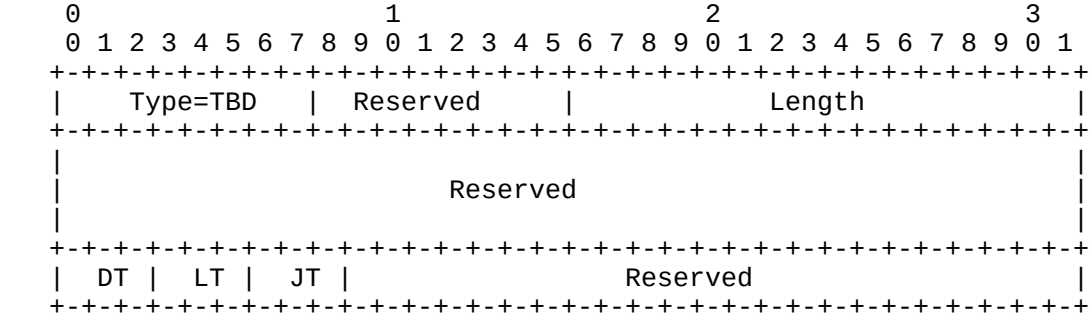
Upstream Maximum Bandwidth: Maximum Upstream bandwidth in bytes per second, 0 means no information is available.

Downstream Maximum Bandwidth: Maximum Downstream bandwidth in bytes per second, 0 means no information is available.

The instance identifier accommodates network traffic where multiple 5-tuples exist for a particular data flow, but the bandwidth flows only over the aggregate of the multiple 5-tuples. One example of this are a phone call which rings on two phones. Only one of those phones will answer first (and send data). FLOWDATA is signaled for both of those phone’s IP addresses and ports, using the same Instance Identifier, indicating to the network that the flow data is being shared with those two different 5-tuples. Another example is TCP video streaming which retrieves short pieces of the movie, often over separate TCP connections for load balancing, which would use the same Instance Identifier for each TCP connection. The way the instance identifier is determined is out of the scope of this document.

9.2.2. FLOWDATA Response

This IE is meant for responses from network to endpoint. It can be included in MD-RESP-UP or MD-RESP-DN, therefore indicating the direction for which the response applies.



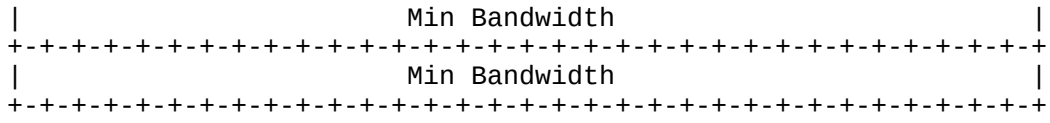


Figure 5: FLOWDATA Response

Type: TBD (optional to process)

Reserved: Must be 0 and ignored by the server.

Length: Option Length is 24 octets.

May appear in: STUN/ICE Binding Request and Response, inside the MD-RESP-UP and/or MD-RESP-DN STUN attributes.

Maximum occurrences: 1

When included in MD-RESP-UP TLV the FLOWDATA Response indicate the response from middleboxes that are on the upstream path. When included in MD-RESP-DN TLV the FLOWDATA Response indicate the response from middleboxes that are on the downsteam path.

Description of the fields:

Reserved: 96 bits, MUST be ignored on reception and MUST be 0 on transmission.

DT: Delay Tolerance, 0 means no information is available.

LT: Loss Tolerance, 0 means no information is available.

JT: Jitter Tolerance, 0 means no information is available.

Reserved: Reserved (7 bits), MUST be ignored on reception and MUST be 0 on transmission

Minimum Bandwidth Minimum bandwidth in bytes per second, 0 means no information is available.

Maximum Bandwidth: Maximum bandwidth in bytes per second, 0 means no information is available.

9.2.3. Usage Example

This section describes how the STUN protocol elements defined above are used to implement flow prioritization.

- Endpoint Metadata Request (REQ-RESP) - Flow Prioritization: Endpoint asks flow prioritization by including in the Binding request non-0 values in the FLOWDATA Request and values initialized to 0 in MD-RESP-UP and MD-RESP-DN TLVs. Upstream MALICE nodes update the MD-RESP-UP with the results. Peer includes in the Binding response the received MD STUN TLVs and the MD-PEER-CHECK-RESP. Downstream MALICE nodes update the MD-RESP-DN TLV. In the example below, the endpoint received the required prioritization for the upstream direction and a lower than requested one for downstream.
  - Binding Request sent by MALICE Client:
    - MD-AGENT (InstID=0, uDT=1, uLT=1, uJT=1, dDT=2, dLT=2, dJT=2, uMinBW=4mbps, uMaxBW=5mbps, uMinBW=5mbps, MaxBW=10mbps)
    - MD-RESP-UP (DT=0, LT=0, JT=0, MinBW=0mbps, MaxBW=0mbps)
    - MD-RESP-DN (DT=0, LT=0, JT=0, MinBW=0mbps, MaxBW=0mbps)
  - Binding Response received by MALICE Client:
    - MD-AGENT (InstID=0, uDT=1, uLT=1, uJT=1, dDT=2, dLT=2, dJT=2, uMinBW=4mbps, uMaxBW=5mbps, uMinBW=5mbps, MaxBW=10mbps)
    - MD-ATTR-UP (DT=1, LT=1, JT=1, MinBW=4mbps, MaxBW=5mbps)
    - MD-ATTR-DN (DT=2, LT=2, JT=2, MinBW=4mbps, MaxBW=5mbps)
    - MD-PEER-CHECK-RES ("Success")

10. Acknowledgements

Authors would like to thank Paul Jones, Sergio Mena de la Cruz and Tirumaleswar Reddy for their comments and review.

11. References

11.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC4732] Handley, M., Rescorla, E., IAB, "Internet Denial-of-Service Considerations", RFC 4732, December 2006.

[RFC5389] Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, "Session Traversal Utilities for NAT (STUN)", RFC 5389, October 2008.

[RFC6398] Le Faucheur, F., "IP Router Alert Considerations and Usage", BCP 168, RFC 6398, October 2011.

[RFC5245] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", RFC 5245, April 2010.

11.2. Informational References

[I-D.ietf-rtcweb-security-arch] Rescorla, E., "RTCWEB Security Architecture", draft-ietf-rtcweb-security-arch-06 (work in progress), January 2013.

[I-D.muthu-behave-consent-freshness] Perumal, M., Wing, D., R, R., and H. Kaplan, "STUN Usage for Consent Freshness", draft-muthu-behave-consent-freshness-03 (work in progress), February 2013.

Authors' Addresses

Reinaldo Penno (editor)
Cisco Systems, Inc.
170 West Tasman Drive
San Jose 95134
USA

Email: repenno@cisco.com

Paal-Erik Martinsen
Cisco Systems, Inc.
Philip Pedersens vei 20
Lysaker, Akershus 1366
Norway

Email: palmarti@cisco.com

Dan Wing
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134
USA

Email: dwing@cisco.com

Anca Zamfir
Cisco Systems, Inc.
EPFL, Quartier de l'Innovation
Ecublens, Vaud 1015
Switzerland

Email: ancaz@cisco.com