MIP6 Working Group
INTERNET DRAFT
Expires: March 2006

K. Kuladinithi
N. A. Fikouras
C. Goerg
ComNets-ikom,
Uni. Bremen
Koltsidas Georgios
Fotini-Niovi Pavlidou
Aristotle University of
Thessaloniki
Oct 2005

Filters for Mobile IPv6 Bindings (NOMADv6)
draft-nomadv6-mobileip-filters-03.txt

Status of this Memo

Copyright Notice

Abstract

Filters for Mobile IPv6 Bindings (NOMADv6) introduces a set of extensions for MIPv6 protocol that allows for intelligent use of multiple points of attachment simultaneously, on a mobile node. It specifies a set of rules (filters) communicated to binding agents using binding updates. In turn, binding agents use this information

to determine whether and where to route flows associated with the mobile node. In this manner, it is possible for a mobile node to

distribute flows or packets of a flow among its available points of attachment or to request that such flow is dropped before traversing the Internet fabric, with or without notification to their source. These extensions mirror a similar extension defined for Mobile IPv4 (NOMADv4) but has been extended to cater to the behavior of IPv6.

Table of Contents

1.Introduction

This document extends Mobile IPv6 protocol, introducing a set of rules (called filters) that are transmitted with binding updates by a mobile node. When receiving the binding update with filters, a binding agent (Mobile IPv6 entities that can maintain bindings, HA, CN, MAP) forwards flows matching filters defined by a mobile node to the point of attachment associated with the respective filter. In this manner it is made possible for mobile nodes to use multiple active points of attachment simultaneously and efficiently.

This draft defines a series of different filter modules that can be used independently or combined to form complex filters. Such filters are relayed to binding agents during binding updates and are included in signaling as mobility options. Binding agents capable of maintaining filters are called filtering agents. All filters contained in a binding update are associated with the point of attachment (care-of-address) indicated in the binding update. In this manner, filtering agents become aware of the relationship between certain flows and specific bindings.

Flows intercepted by, or originating from a Filtering Agent (HA, CN, MAP) will be filtered and individual flows will be forwarded to the are-of address indicated by the respective binding. This enables mobile nodes to distribute flows or to distribute packets of a single flow, among their available points of attachment.

Mobile IPv6 does not provide the facilities for a mobile node to register multiple care-of-addresses for a single home IP address. This functionality is important for the considerations presented in this document. This draft introduces the `N´ bit to the binding update message. This bit, when set, informs the filtering agent to hold multiple simultaneous binding for the given home address of the mobile node and then manipulate the IP traffic based on the filtering rules sent as mobility options. The benefits and goals of using multiple points of attachment simultaneously are explained in [9],

[10] and [11], highlighting the benefits with real life scenarios.

The operation of filtering for Mobile IPv6 is intended to mirror the operation of filtering for Mobile IPv4 [2], with changes necessary to provide a similar behavior. The considerations presented in this document are collectively referred to as the NOMADv6 Extensions.

2 Terminology

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 [1].

This document uses the following terms:

Destination Option
As defined in [3]

Domain A collection of networks sharing a common network administration.

Home link
As defined in [3].

Foreign link
As defined in [3].

Home Agent (HA)
As defined in [3].

Correspondent Node (CN)
As defined in [3].

Mobile Node (MN)
As defined in [3].

Mobility Anchor Point (MAP)
As defined in [4].

Care-Of-Address
As defined in [3].

Mobility Binding
As defined in [2].

Binding Agent (BA)
Any Mobile IP entity (HA,CN,MAP) that can maintain mobility bindings.

Binding Update

Mobile IP signaling with the purpose of establishing or updating a mobility binding.

Binding Acknowledgement
A Binding Acknowledgement is used to acknowledge receipt of a Binding Update, if an acknowledgement was requested in the Binding Update, the binding update was sent to a home agent, or an error occurred.

Filtering Agent (FLA)
Any binding agent that can maintain filters for mobility bindings in its binding cache, such as the HA, CN or MAP.

Filter Module (FLM)
A single filtering criteria that specifies the condition to check for filtering data.

Filter (FL)
A collection of filter modules. Each filter module is interpreted as having an AND relationship with the other filter modules inside the filter. The relationship between filters of a mobile node, is OR.

Filtering Update (FLU)

Mobile IPv6 signaling (binding update) with the purpose of establishing a new mobility binding that contains one or more Filter extensions as mobility options. Each Filtering Update should include the N bit ON on the binding update mobility header.

Filtering Acknowledgement (FLAC)
Mobile IPv6 signaling (binding acknowledgment) for returning the result of a Filtering Update.

Default Filter (DF)
A special Filter applicable for all flows not matching any other Filter. Is either defined by mobile node or automatically allocated from Filtering Agents to the lowest defined Index of 0.

Idle Mobility Binding (IMB)
A mobility binding without Filters.

3.Comparison with Filters for Mobile IPv4 (NOMADv4 vs NOMADv6)

a. In MIPv6, there are no dedicated FAs, GFAs, or RFAs. The roles of these entities have been taken over by the particular routers, which are located along the path which a packet traverses from the HA to the MN or CN to MN. These special routers are called MAPs. Therefore,

in an MIPv6 environment, MN destined packet filtering SHOULD be done by an HA, CN or an MAP.

b. Mobile IPv6 route optimization can be deployed on a global scale between all mobile nodes and correspondent nodes. Therefore, CNs are considered along with Has as filtering agents.

d. MIPv6 lacks support for multiple simultaneous bindings that are available in MIPv4 [6]. The filtering concept described in this draft requires that all filtering agents are able to cater for simultaneous bindings. For this a new N bit is introduced to the binding update mobility header for the support of simultaneous bindings in NOMADv6.

e. Sub types of the Filter extensions are defined on the first byte of the Data field in NOMADv6. NOMADv4 uses standard short and long TLV format as defined in [6] for including sub types.

4 NOMADv6 Protocol Overview

This section provides an overview of how filters for MIPv6 bindings can be realized.

4.1 Protocol Description

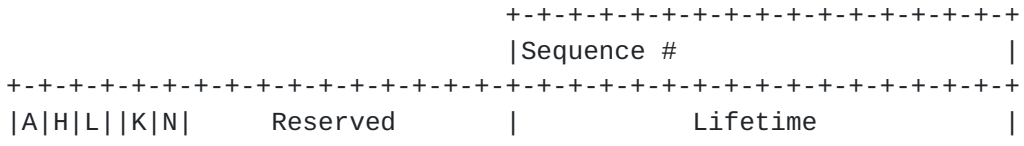
4.1.1 Multiple network interface support and N bit

Filters for Mobile IPv6 is applicable only in the context of a mobile node maintaining multiple points of attachment to one or more

Internet administrative domains. NOMADv6 does not make any assumptions on the number of permanent home IP addresses maintained by a mobile node or the number of home agents related with each of the home IP addresses. In that case, a point of attachment can be associated with one or more home IP addresses and consequently reused in different bindings with different binding agents.

The NOMADv6 reserves a bit in the Message Data field of the binding update message, called the N bit, for the purpose of introducing simultaneous bindings support. Upon receiving a binding update with the N bit set, a filtering agent MUST issue a new binding for the mobile node home IP address indicated in the binding update without affecting any of the existing mobility bindings.

The format of the Message Data field in the binding update message is as follows [3]:



+-+

N When set, the binding agent MUST act based on the functions described in section 4.1.3 and add a new entry to the binding cache without deleting any existing entries for the mobile node´s home address which is specified in the home address destination option.

4.1.2 Sending Filtering Rules

Mobile nodes that wish to associate Filters with an acquired care-of address are required to issue a binding update including a list of Filters that indicate which flows are associated with the registered care-of-address. Such signaling is termed as Filtering Updates. A Filter is consisted of one or more Filter Modules and is terminated by a Filter Control Extension. A Filter Module may contain several predicates. There is an OR relationship between predicates of a Filter Module. Moreover, there is an AND relationship between Filter Modules of the same Filter. Consequently, in order for a flow to match a Filter, it is required to qualify for all of the Filter Modules contained in the Filter.

With the help of the Filter Control Extension, the Filter´s purpose can be defined. It contains the Filter´s Index, and a Weight field. The Index identifies uniquely, a Filter for a given mobile node while the Weight field indicates the relative amount of traffic for which the filter is applicable. If the Weight field is set to zero, then all matching flows will be dropped without notification to their source.

A mobile node may define more than one Filter for a specific mobility binding. The declaration of these Filters may take place during one or more Filtering Updates. In the case of shared Filters, packets of matching flows will get distributed between multiple points of

attachment with respect to the Weight value of each filter. A mobile node may share a Filter between mobility bindings by issuing a Filtering Request from each respective point of attachment. The first one will contain the full Filter (Filter Body + Filter Control Extension) while all subsequent Filtering Requests will contain only a Filter Control Extension indicating the Index number of the Filter to be shared.

Flows that fail to match any of the defined Filters are handled as defined by the Filter with the lowest possible Index, termed as Default Filter. A mobile node may define some of the attributes of the Default Filter such as the associated mobility binding and its Weight field by issuing a Filtering Request. Otherwise, these will be configured by each Filtering Agent (see section 4.1.3).

When a mobile node needs to delete filters, it sends the binding update containing a single Filter Control Extension. The index of the filter to be deleted should be sent in the index field. If a mobile node wishes to delete all filters, index should be set to 255.

All the filtering rules which have to be set in the mobility options of a binding update will be described in section 7.1. The rules by which a mobile node decides on the set of Filters are considered beyond the scope of this document. The extensions presented in this document do not affect in any way the mobile node´s choice on the point of attachment to be used when returning traffic.

4.1.3 Processing at the Filtering Agent

Filtering Updates will be processed by one or more Filtering Agents. A Filtering Agent can be any Mobile IPv6 entity that can maintain mobility bindings with Filters, like a HA, CN or MAP.

Flows that fail to match any of the defined Filters are handled as defined by the Default Filter. If a mobile node fails to promptly define a Default Filter or if the associated mobility binding expires then a new one will automatically be configured by each involved Filtering Agent to the lowest possible Index of 0.

Different Filtering Agents may apply different Default Filter definitions; however it is recommended that the Default Filter be associated with the mobility binding with the longest outstanding lifetime with the Weight field set to 1.

A mobile node may issue Filters corresponding to flows that do not yet exist. When such a flow is initiated it will be handled by the Filtering Agents as indicated by the respective Filter.

A Filtering Acknowledgement contains one or more extensions to the binding acknowledgment indicating the Index of a Filter along with a Code signifying the result of the respective Filtering Update. The Code is used to relay success or the reason of rejection to the mobile node.

Upon receiving a binding update with the N bit not set, a Filtering Agent should replace all existing bindings in its binding cache with the one indicated by the binding update and remove all associated bindings. From that point on the Filtering Agent is required to act as per [3] and ignore the considerations presented in this document. A mobility binding in that state is termed as an Idle Mobility Binding.

4.1.4 Lifetime of a Filter

A Filter remains valid for the lifetime of the corresponding mobility binding. If the lifetime of a binding expires or it is cancelled by the registration of another mobility binding then all associated Filters are deleted from the binding cache.

When renewing mobility binding, a mobile node is not required to include any reference to any requested Filters. A mobile node SHOULD set the N bit on in its Binding Update and then the Filtering Agent SHOULD refresh the lifetime of the binding and all filters, related to the home address sent on the Destination option of the Binding Update.

4.1.6 Filters that split flows between different home addresses.

A mobile node that maintains multiple home IP addresses can share multiple points of attachment between them. That is, having established a binding for a certain home IP address and a specific point of attachment does not restrict the use of the same point of attachment with an other home IP address even when home IP addresses share the same home agent.

A MN with more than one points of attachment, MAY have different home addresses (multi-homed mobile node) for each of those points of attachment. These addresses MAY be registered with different HAs or with the same HA. In this situation, if MN wishes to split its flows coming to one point of attachment (A) to another (B), MN MUST send a Filtering Update via A, including an alternate CoA mobility option with the CoA of the point of attachment B. The HA of the point of attachment A, upon receival of this binding update, MUST tunnel the matching flows to the CoA of the point of attachment B. (Refer Fig. 1)

4.1.7 De-registration when a single PoA is at the home network

When a mobile node is connected to its home network by one of its points of attachment, the mobile node MUST de-register all the other bindings that belong to the same home IP address. In this way, mobile node SHOULD delete all filters associated with the specific binding(s) and revert to operations as defined in [3].

4.2 Model of Operation

Filters for Mobile IPv6 Bindings has two modes of operation that can be seamlessly combined but for the sake of simplicity are covered in this section separately. The first model of operation concerns the

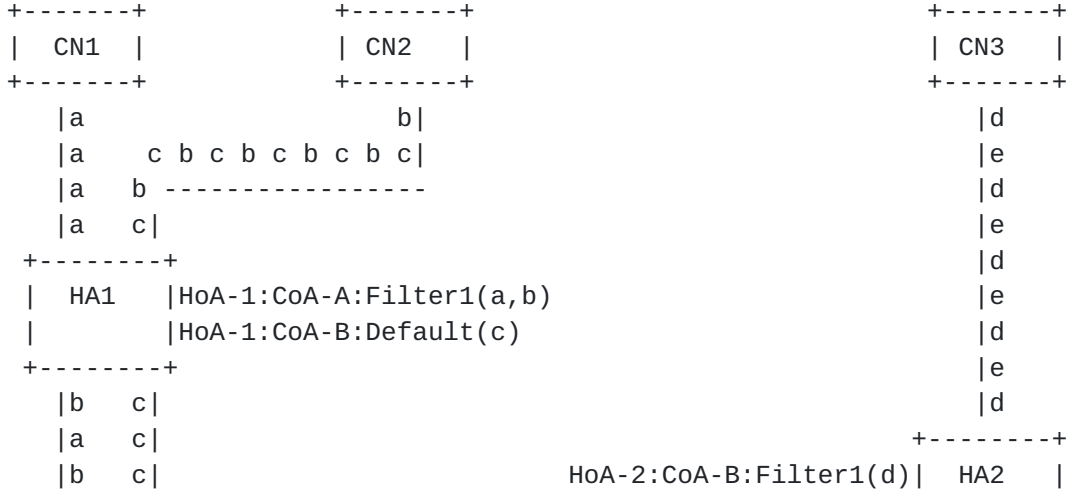
management of whole flows while the second model addresses the distribution of the individual packets of flows between points of attachment.

The distribution of multiple flows is illustrated in figure 1. It shows a mobile node that maintains multiple access interfaces simultaneously. Each interface provides a point of attachment through a foreign network (FN-A, FN-B and FN-C). The extensions presented do not provide any restriction as to how many points of attachment a mobile node may maintain or how many home agents it can be attached to. For example, the mobile node in figure 1 has two separate points of attachment through FN-A and FN-B, communicating with CN-1 and CN-2 via HA-1. In addition, the mobile node maintains another point of attachment through FN-C, corresponding with CN3 via HA2. MN uses one home address (HoA-1) for two interfaces, while the other interface is connected to the HA2 via HoA-2.

In figure 1, the mobile node maintains five communication sessions with correspondent nodes of CN1, CN2 and CN3. Flows associated with CN1 are denoted by 'a' and CN2 are denoted by 'b' & c while the respective flows for CN3 are denoted by d and 'e'.

When MN requires to transfer flows `a´ & `b´ (Filter1) to the interface connected to the FN-A, while receiving all the other flows (Default filter) over FN-B, MN sends a new binding as defined in 4.1.2 with the N bit set.

When MN requires transferring flow `d´ to the interface connected to FN-B, MN sends a binding update with HoA-2 and CoA-C, together with CoA-B in the Alternate care-of-address mobility option and with the required filtering extensions (see section 4.1.6). This causes the addition of a new binding entry (HOA-2:CoA-B:Filter1) at HA2. This will not result in any deletion of existing binding entries (HoA-2:CoA-C will remain). HA2, will now intercept all flows (d & e), but will tunnel flow `d´ through FN-B, while flow `e´ or any other flows continues through FN-C.



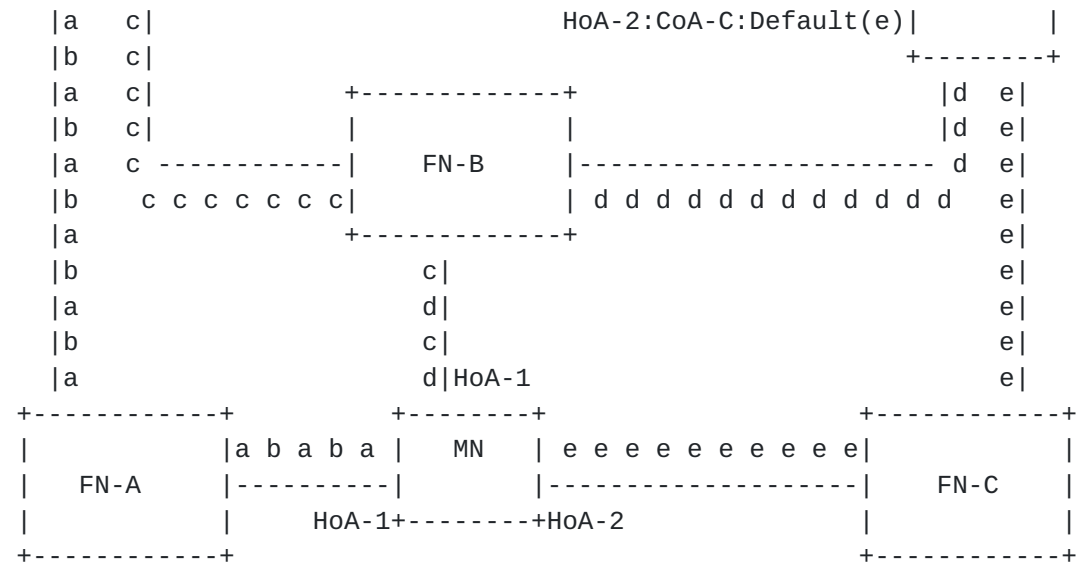
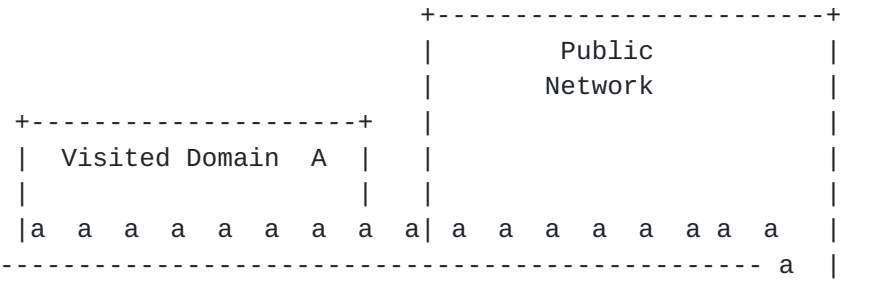


Figure 1: A mobile node with three points of attachment in different foreign networks (CoA-A, CoA-B & CoA-C) with 2 home addresses (HoA-1 & HoA-2). Incoming flows are redirected by the respective filtering agents (HA1, HA2) to different care-of-addresses, based on the filtering rules.

In the example presented in figure 1, the HA1 & HA2 act as the filtering agents. But, any Mobile IPv6 binding agent (HA, MAP, CN) can act as filtering agents. To return traffic, a mobile node may choose any of the available points of attachment.

Figure 2, illustrates the second model of operation. It shows the mobile node that maintains two points of attachment in visited domain A and B, while maintaining one active flow from CN1, denoted with a . In this example, MN maintains two bindings with the CN1 for visiting domain A and B. NOMADv6 extensions are applied to share a Filter (Flow a) over point of attachment A and B. However, distribution of single flow could lead to performance degradation when using standard TCP applications. But, for the applications that could be reorder the out of sequence packets at the receiver, this mechanism performs well.



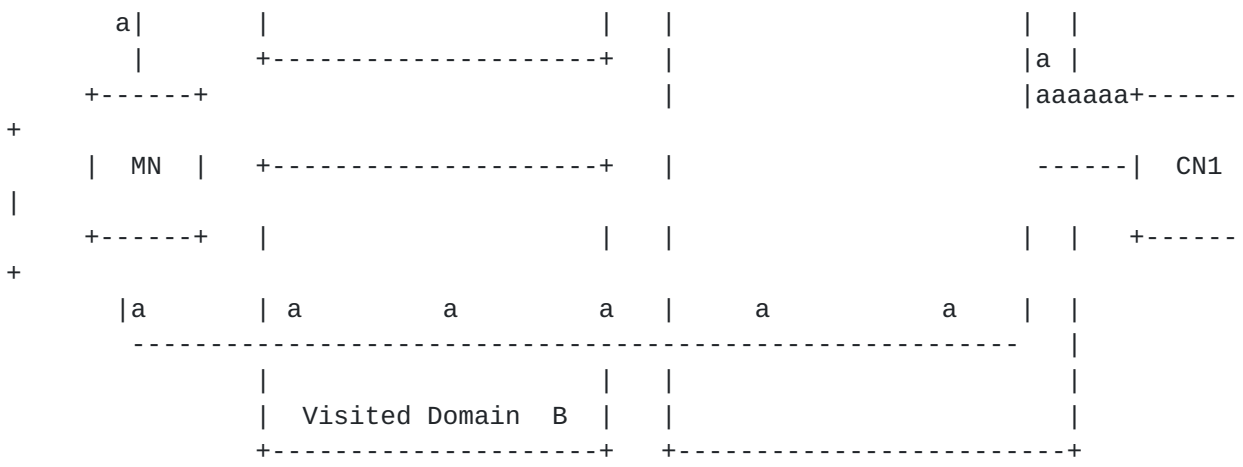


Figure 2: A mobile node with multiple points of attachment in different visited domains. A single incoming flow is distributed by the respective Filtering Agents (HA,CN or MAP) to a different care-of address.

5 Backword compatibility with basic Mobile IPv6

If the binding update does not have the N bit set, the processing of the BU is same as [3]. But if the binding agent has already registered multiple care-of addresses for the same home address, the binding agent MUST overwrite all the bindings for the home address specified in the destination option. Binding updates without the N bit set are considered as idle mobility bindings. In order to preserve backward compatibility with the basic protocol [3], it is stated in section 4.1.3 that a Filtering Agent maintaining only idle Mobility Bindings for a mobile is required to act as per [3] and to ignore the behavior presented in this document.

6 Associating Filters with Bindings

This section gives a detailed description of the steps taken by a mobile node that wishes to associate filters with its bindings. Furthermore, it presents how a filtering agent reacts to the receipt of a binding update containing a list of filters.

6.1 Mobile Node Considerations

A mobile node that acquires a care-of address within a visited domain may issue a binding update containing a list of Filters. All included Filters will be associated with the registered care-of address at all Filtering Agents (HA,CN,MAP). A mobile node that maintains multiple points of attachment may request for simultaneous mobility bindings by setting the N bit in its binding Updates. However, each of the

binding updates must contain its own list of filters. Should the binding update be rejected then the mobile node will receive a Filtering Acknowledgement with a binding acknowledgement Extension indicating the Index of the Filter that was rejected along with the reason for rejection.

It is important for a mobile node to keep a record of the Filters and their corresponding Index numbers per home address.

For the management of Filters eight scenarios are identified. These are presented along with the actions to be undertaken by the mobile node.

6.1.1 Creating a new mobility binding with Filters

In order to create a new mobility binding with associated Filters, the mobile node MUST issue a binding update including one or more full Filter definitions (one or more Filter modules with Filter Control Extension) as mobility options, attached to the binding update mobility header. Each of the Filters MUST be allocated a different Index number.

The destination of the Filtering Update is identified as described in [3].

6.1.2 Replacing a Filter of a mobility binding by Index

In order for a mobile node to replace an existing Filter, it is required to issue a binding update with a full definition of the new Filter. The Filter Control Extension of the Filter must indicate the Index of the Filter to be replaced. The Weight value of the new Filter MAY be different from the Weight of the previous Filter definition.

6.1.3 Adding new Filters to an existing mobility binding

In order for a mobile node to add new Filters to an existing mobility binding, it is required to act as if creating a new mobility binding with Filters. It is necessary for the new Filter to adopt an unallocated Index number otherwise it would be replacing an existing Filter with that Index.

6.1.4 Sharing a Filter between mobility binding

A mobile node may share a Filter between mobility bindings by issuing a binding update from each respective point of attachment. The first one will contain the full Filter (Filter Body + Filter Control Extension) while all subsequent Filtering Requests will contain only a Filter Control Extension indicating the Index number of the Filter to be shared.

6.1.5 Renewing a mobility binding with Filters

Periodically, a mobile node is required to renew its mobility bindings in order to extend their lifetime. Renewing a mobility

binding may occur as described in [3]. The mobile node sets the N bit, when sending a binding update in order to renew all filters allocated for the home address defined in the destination option.

6.1.6 Deleting a defined Filter/s for a mobility binding

In order for a mobile node to delete an existing Filter for a mobility binding, it is required to issue a binding update from any care-of address. The binding update must include a Filter Deletion Extensions indicating the Index of each Filter to be deleted.

6.1.7 Deleting all Filters for a mobility binding

In order for a mobile node to delete all existing Filters for a mobility binding, it is required to issue a binding update from any care-of address. The binding update must include a Filter Deletion Extensions with the Index field set to zero.

6.1.8 Transferring a Filter between mobility bindings

It is required to act as if creating a new mobility binding with Filters and send out a binding update from the point of attachment to which it wants to transfer the Filter to the other. The Filtering Update must attach the Alternate Care-of-Address mobility option and must contain the full Filter. Alternate care-of-address option contains the care-of-address of the point of attachment, which the filter should be transferred. In this way, the transferring of filters are possible irrespective of the same or different home addresses used for each of attachment.

The Weight field of the Filter Control Extension indicates the relative amount of traffic for which a Filter is applicable. If the Weight field is set to zero then all matching flows will be dropped without notification to their source. For any other value of Weight, matching flows will get forwarded to the point of attachment indicated by the corresponding mobility binding. In the case of shared Filters, packets of matching flows will get distributed between multiple points of attachment with respect to the Weight value of each Filter.

6.2 Filtering Agent Considerations

This section contains considerations for Filtering Agents. These are Mobile IPv6 entities that can maintain mobility bindings such as HAs, CNs or MAPs when hierarchical Mobile IPv6 is supported.

Should the Filtering Agent fail to apply any of the Filters then for each such Filter a Filter Acknowledgement Extension must be included in the Filtering Acknowledgement indicating the Index of the rejected Filter along with the reason of rejection. If authentication of the Filtering Update fails, then none of the Filters MUST be applied.

Should the Filtering Agent succeed in applying the Filters, then the Filtering Acknowledgement indicating the index of the success MUST be sent, only if `A´ bit is set on the Binding Update.

When a Filtering Agent intercepts a packet for a mobile node for which it maintains a mobility binding, it is required to identify whether the packet matches any of the Filters associated with the mobility binding. If so, the packet is handled as described by the Weight value of the corresponding Filter. If no matching Filter is found then the packet is handled as indicated by the Default Filter.

When a mobility binding expires or is deregistered by a mobile node then all associated Filters are deleted with it. Whenever a Filtering Agent received a Filtering Update without setting the N bit (i.e. Binding Update), it is required to overwrite all the bindings set for the home address and keep the binding for the new care-of-address, sent. This binding is called the Idle Mobility Binding and it is required to ignore the behavior described in this document and to act as per [3].

7 NOMADv6 Extensions to MIPv6 Binding Messages

In this section, the new Mobile IPv6 extensions required to support the Filters for Mobile IPv6 bindings are specified.

All filtering extensions are sent as mobility options of the binding update or binding acknowledgment mobility header as defined in [3]. The filtering extensions are encoded using a type-length-value (TLV) format in the mobility options.

A complete mobility header, once filter extensions are attached SHOULD be an integer multiple of 8 octets long.

Filter extensions can be categorized into 4 types,

- Filter Module Extensions
- Filter Control Extension
- Filter Deletion Extension
- Filter Acknowledgement Extension

The Filter Module Extensions specify the different filtering rules that the mobile node wishes to inform the Filtering Agent. There are 10 such filter extensions. These extensions are always attached to

the Binding Update mobility header as mobility option/s. To form a valid Filter, at least one of the filter module extensions must be included. The Filter Control Extension must appear once in every Filter following all Filter Modules. Filter control extension may appear more than once in a binding update interleaving with Filter declarations.

Filter Modules of the same type may not appear in a Filter more than once. A Filter Module may include one or more predicates. There is an OR relationship between Filter Module predicates. That is, in order for a flow to match a Filter Module, it is required to qualify for

any of the predicates in it. In addition, there is an AND relationship between Filter Modules of a Filter. As such, in order for a flow to match a Filter, it is required to qualify for all its Filter Modules.

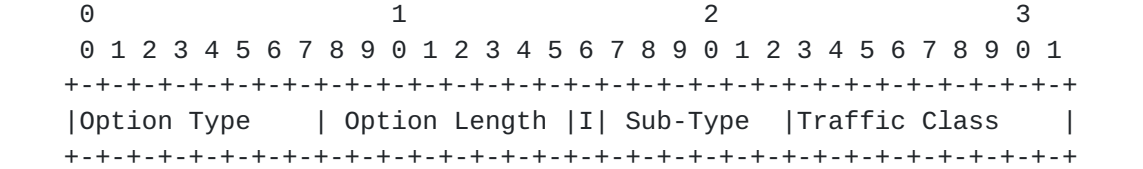
In Filter Modules, the first byte of the data is allocated to define the types of the Filter Modules. The left most bit of the Sub-Type field is used to determine whether the rules included in the Filter Module are positive or negative. In the first case, a flow is required to match exactly the predicates included in the Filter Module while in the second the inverted (NOT) rule is applied.

The Filter Deletion extension is an extension sent to the Filtering Agent by the mobile node to deleted filter/s. This extension is attached to Binding Update mobility header. The Filter Acknowledgement extension is an extension sent to the mobile node by the Filtering Agent to inform of success or any failure of filter accommodation. This extension is attached to Binding Acknowledgement mobility header.

7.1 Filter Module Extensions

7.1.1 Traffic Class Filter Extension.

Specifies the extension required to filter IPv6 packets, based on the value placed on the Traffic Class field of a packet. This has an alignment requirement of 2n. The format is as follows.



Option Type — The type which describes a collection of NOMADv6 extensions (To be defined)

Option Length — N+1, where N is the number of Traffic Class

entries.

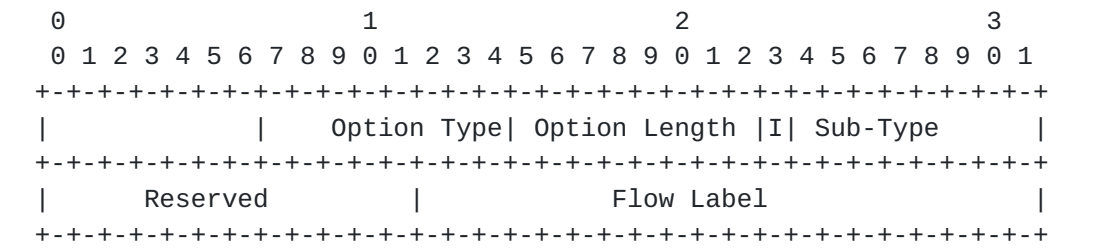
I — Invert. A left most bit of the Sub-Type field is used to invert each predicate of the Filter Module. Due to this bit, two different Sub-Type values are given.

Sub-Type — 0 for given Module, 128 for inverted Module

Traffic Class — Values, related to different classes or priorities of IPv6 packets.[7]

7.1.2 Flow Label Filter Extension

Specifies the extension required to filter IPv6 packets based on the value placed on the Flow Label field of a IPv6 packet. This has an alignment requirement of 4n+1. The format is as follows.



Option Type — The type which describes a collection of NOMADv6 extensions (To be defined)

Option Length — 4N+1, where N is the number of Flow Label entries. Each Flow Label entry is assumed to take 4 bytes (including the Reserved bits)

I — Invert. A left most bit of the Sub-Type field is used to invert each predicate of the Filter Module. Due to this bit, two different Sub-Type values are given.

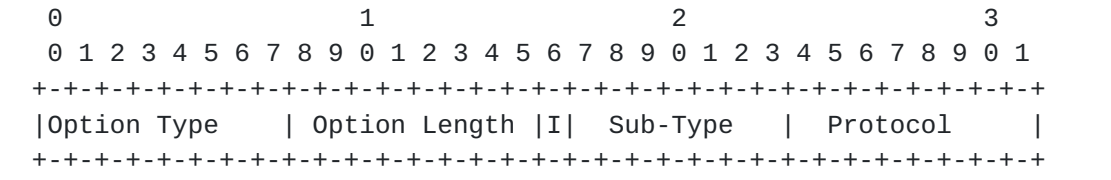
Sub-Type — 1 for given Module, 129 for inverted Module

Flow Label — Any value which is labelled on this field of a IPv6 packet. Refer [7] for what and how flow label is in IPv6.

7.1.3 Protocol Extension

Specifies one or more protocol to be filtered. This has an alignment

requirement of 2n. The format is as follows.



Option Type — The type which describes a collection of NOMADv6 extensions (To be defined)

Option Length — N+1, where N is the number of Protocol entries.

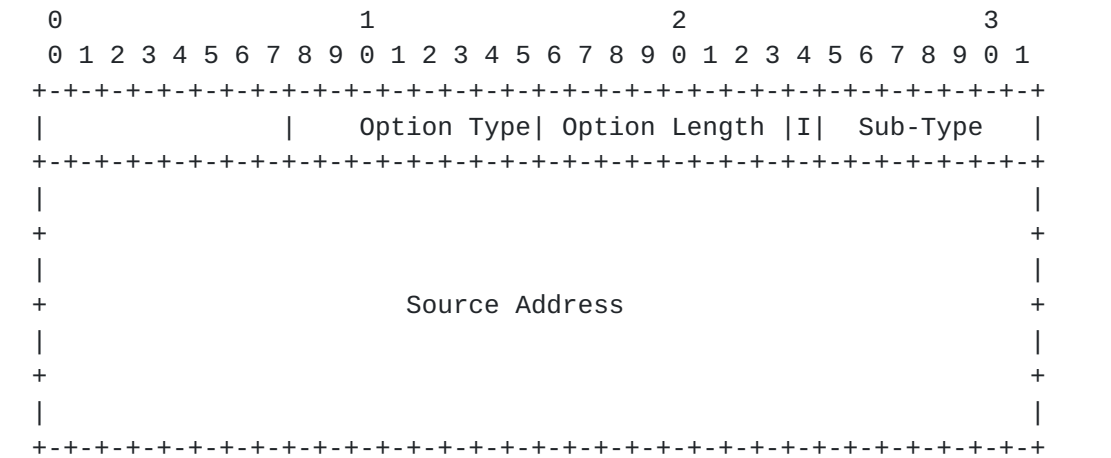
I — Invert. A left most bit of the Sub-Type field is used to invert each predicate of the Filter Module. Due to this bit, two different Sub-Type values are given.

Sub-Type — 2 for given Module, 130 for inverted Module

Protocol — Identifies the next level protocol used in the data portion of the IPv6 datagram. The values for various protocols are specified in [7]

7.1.4 Source Address Extension

Specifies one or more source addresses to be filtered. This has an alignment requirement of 8n+5. The format is as follows.



Option Type — The type which describes a collection of NOMADv6 extensions (To be defined)

Option Length — 16N+1, where N is the number of source addresses.

I — Invert. A left most bit of the Sub-Type field is

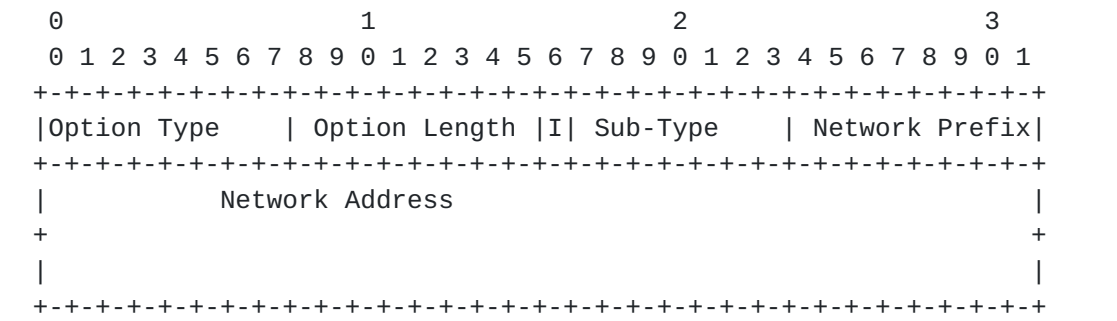
used to invert each predicate of the Filter Module. Due to this bit, two different Sub-Type values are given.

Sub-Type — 3 for given Module, 131 for inverted Module

Source Address — Identifies the source address/es to be filtered.

7.1.5 Source Network Extension

Specifies one or more source network/s to be filtered. This has an alignment requirement of 8n+4. The format is as follows.



Option Type — The type which describes a collection of NOMADv6 extensions (To be defined)

Option Length — 9N+1, where N is number networks.

I — Invert. A left most bit of the Sub-Type field is used to invert each predicate of the Filter Module. Due to this bit, two different Sub-Type values are given.

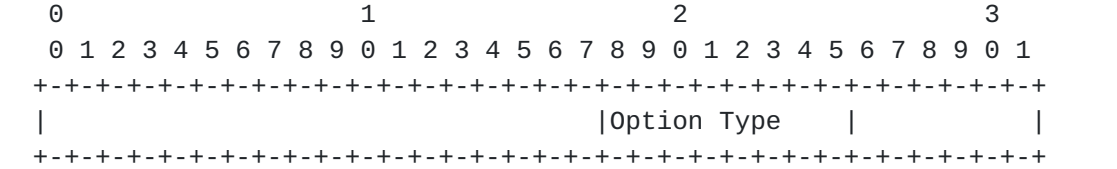
Sub-Type — 4 for given Module, 132 for inverted Module

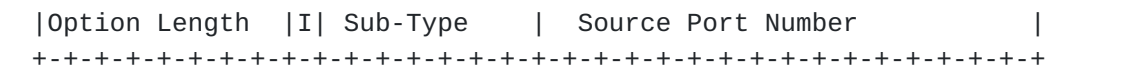
Network Prefix — Identifies the network prefix to be filtered.

Network Address — Identifies the first 64 bits of the Source network address.

7.1.6 Source Port Extension

Specifies one or more source ports to be filtered. This has an alignment requirement of 2n+3. The format is as follows.





Option Type — The type which describes a collection of NOMADv6 extensions (To be defined)

Option Length — 2N+1, where N is number of port entries.

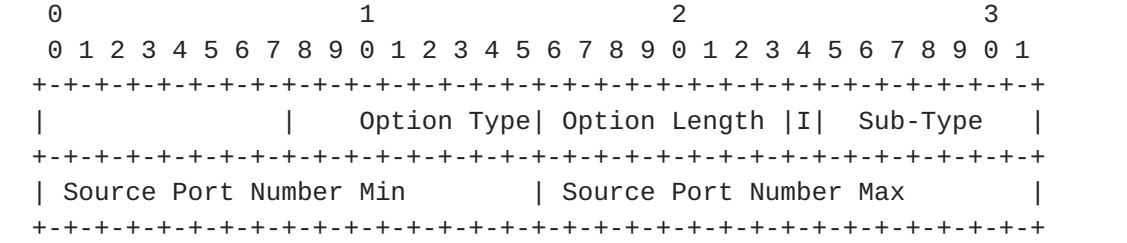
I — Invert. A left most bit of the Sub-Type field is used to invert each predicate of the Filter Module. Due to this bit, two different Sub-Type values are given.

Sub-Type — 5 for given Module, 133 for inverted Module

Source Port — Identifies the Source Port Number/s to be filtered.

7.1.7 Source Port Range Extension

Specifies one or more source ports to be filtered. This has an alignment requirement of 2n+1. The format is as follows.



Option Type — The type which describes a collection of NOMADv6 extensions (To be defined)

Option Length — 4N+1, where N is number of port range entries.

I — Invert. A left most bit of the Sub-Type field is used to invert each predicate of the Filter Module. Due to this bit, two different Sub-Type values are given.

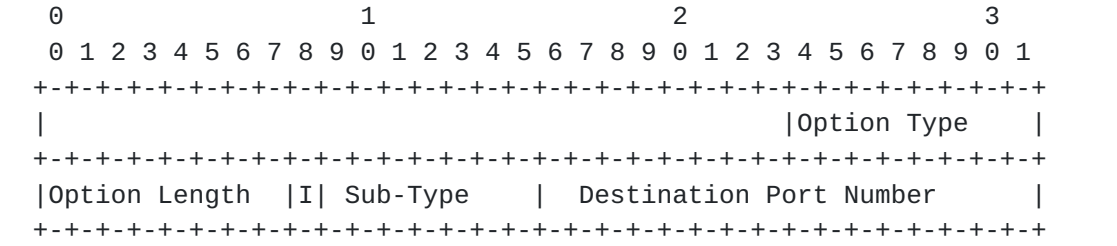
Sub-Type — 6 for given Module, 134 for inverted Module

Port Number Min — Identifies the start point of a range of port numbers.

Port Number Max — Identifies the end point of a range of port numbers.

7.1.8 Destination Port Extension

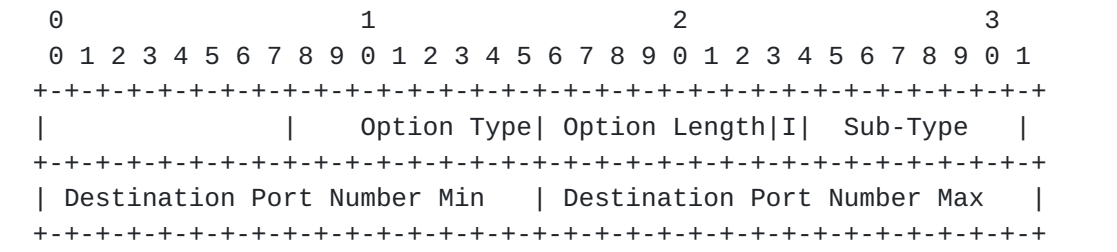
Specifies one or more destination ports to be filtered. This has an alignment requirement of 2n+3. The format is as follows.



Option Type	The type which describes a collection of NOMADv6 extensions (To be defined)
Option Length	2N+1, where N is number of port entries.
I	Invert. A left most bit of the Sub-Type field is used to invert each predicate of the Filter Module. Due to this bit, two different Sub-Type values are given.
Sub-Type	7 for given Module, 135 for inverted Module
Destination Port	Identifies the destination Port Number/s to be filtered.

7.1.9 Destination Port Range Extension

Specifies one or more destination ports to be filtered. This has an alignment requirement of 2n+1. The format is as follows.



Option Type	The type which describes a collection of NOMADv6 extensions (To be defined)

Option Length 4N+1, where N is number of port range entries.

I Invert. A left most bit of the Sub-Type field is used to invert each predicate of the Filter Module. Due to this bit, two different Sub-Type values are given.

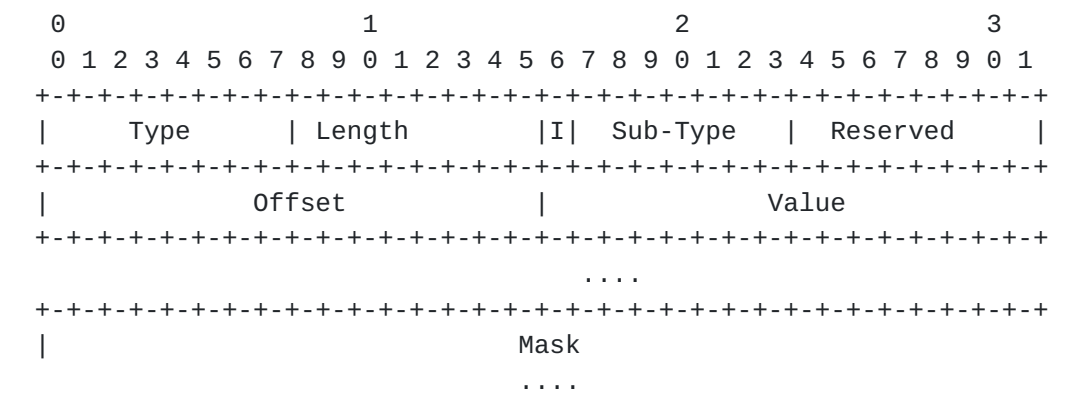
Sub-Type 8 for given Module, 136 for inverted Module

Port Number Min Identifies the start point of a range of port numbers.

Port Number Max Identifies the end point of a range of port numbers.

7.1.10 Free-Form Extension

Specifies the value of an area anywhere within a packet. The alignment requirement is based on the number of bytes on Value field. The format is as follows.



Option Type The type which describes a collection of NOMADv6 extensions (To be defined)

Option Length Is variable, depends on the length of the Value and Mask.

I Invert. A left most bit of the Sub-Type field is used to invert each predicate of the Filter Module. Due to this bit, two different Sub-Type values are given.

Sub-Type 9 for given Module, 137 for inverted Module

Offset Indicates the starting octet location within an IPv6 packet to use to mask with the Mask and check with Value.

Value	Indicates the value to be checked, once masked.
Mask	Indicates the value to use as the mask to mask the octets starting from the offset.

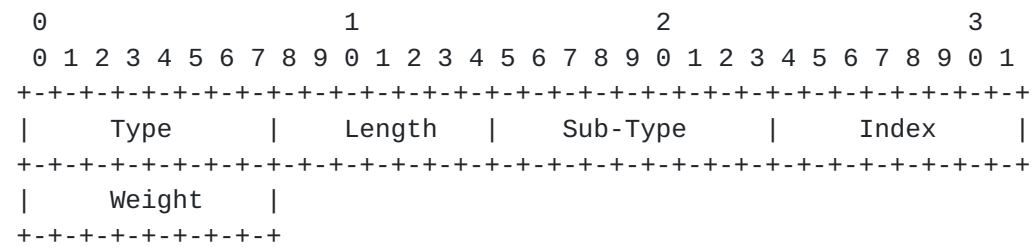
The area indicated by the offset and for a length equivalent to that of Mask is compared against Mask with the bitwise operator AND. The result of this operation is compared against Value. A match would indicate that the packet qualifies the filter.

Value and Mask fields MUST have exactly the same size. However, the length of the Value and Mask may vary with every free-form filter. For the sizes of Value and Mask the following condition holds:

```
Value = Mask = (Length - 4) / 2
```

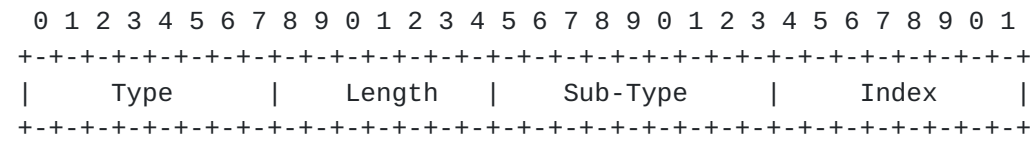
7.2 Filter Control Extension

Specifies a filter´s unique identifier, called the index along with the Filter´s Target.



Option Type	The type which describes a collection of NOMADv6 extensions (To be defined).
Option Length	3
Sub-Type	125
Index	Filter s index number
Weight	Relative amount of traffic for which forwarding Filter is applicable

7.3 Filter Deletion Extension



Type	Length	Sub-Type	Index

Type — The type, which describes a collection of extensions having a common data type. (To Be Defined).

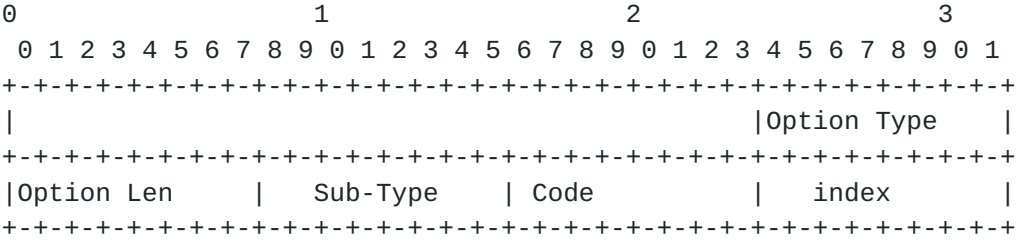
Sub-Type — 126

Length — N, where N is the number of Index entries

Index — A Filter s index number

7.4 Filter Acknowledgement Extension

Specifies the format of an acknowledgement extension which is sent with the binding acknowledgement mobility header to inform the MN about the status of Filters processed at the Filtering Agent. This has an alignment requirement of 2n+3. The format is as follows.



			Option Type
Option Len	Sub-Type	Code	index

Option Type — The type which describes a collection of NOMADv6 extensions (To be defined)

Option Length — 3

Sub-Type — 127.

Index — Filter´s index number

Code — Values to indicate the status of the Filter accommodation

The following section specifies the values to use within the Code field of the Filter Acknowledgement Extension are defined:

Successful Filtering Update Codes:

Code Name	Value
REQUEST ACCEPTED	TBD

Failed Filtering Update Codes:

Error Name	Value
TOO MANY FILTERS	TBD
INVALID FILTER SYNTAX	TBD
UNKNOWN FILTER	TBD
CAN NOT DROP MIP SIG	TBD

The Error Code CAN NOT DROP MIP SIG is used when the mobile node issues a Filtering Update requesting the drop of flows corresponding to Mobile IPv6 signalling such as Router Advertisements, Binding Update, Binding Refresh Request, Binding Acknowledgement or Binding Error.

8. Security considerations

Since the filter extensions defined in this document only concern the messaging between the home agent (or correspondent node with route optimization) and the MN, all security mechanisms that are defined in [3] is considered sufficient to protect the integrity and authenticity of filter extensions that are attached with Binding Update and Binding acknowledgement messages.

References

[1] S. Bradner. Key words for use in RFCs to Indicate Requirement Levels. RFC 2119, IETF, March 1997.
[2] N.A. Fikouras, A. Udugama, K. Kuladinithi, C. Goerg, W. Zirwas. Filters for Mobile IP Bindings (NOMAD).draft-nomad-mobileip-filters-05.txt, IETF, October 2003.
[3] Johnson, D., Perkins, C., and J. Arkko, "Mobility Support in IPv6", RFC 3775, June 2004.
[4] H. Soliman, C. Castelluccia, K. Malki, L. Bellier. Hierarchical MIPv6 Mobility management. Draft-ietf-mobileip-hmipv6-06.txt, IETF, July 2002.
[5] K. Malki, H. Soliman. Simultaneous Binding for Mobile Ipv6 Fast Handoffs. draft-emalki-mobileip-bicasting-v6-06.txt, IETF, July 2005.
[6] C. Perkins, IP Mobility Support for IPv4. RFC 3220, January 2002.
[7] S. Deering, R. Hinden. Internet Protocol Version 6 Specification, RFC 2460, December 1998.
[8] J. Reynolds and J. Postel. Assigned Numbers. Request for Comments 1700, STD 2, IETF, October 1994.
[9] T. Ernst, N. Montavont, R. Wakikawa, C. Ng, E. Paik, K. Kuladinithi and T. Noel, "Goals and Benefits of Multihoming," IETF http://www.ietf.org/internet-drafts/draft-

ernst-generic-goals-and-benefits-01.txt, February 2005
[10] N. Fikouras, K. Kuladinithi, C. Goerg and C. Bormann, "Mobile IPv4 Flow Mobility", draft-nomad-mip4-flow-mobility-pb-00.txt (work in progress), Feb 2004.
[11] N. Montavont, R. Wakikawa, T. Ernst, C. Ng and K. Kuladinithi "Analysis of Multihoming in Mobile IPv6",

http://www.ietf.org/internet-drafts/draft-montavont-mobileip-multihoming-pb-statement-04.txt, June 2005.

A. Changes from Previous Versions

The following updates and changes were made in this version of the Filters for Mobile IPv6 Bindings draft, compared to earlier versions.

A.1. Updates from version 00

Removed the Target field from the Filter Control Extension

Introduced the Weight field in the Filter Control Extension.

Introduced the Filter Deletion Extension

Introduced shared Filters based on the Index field.

Extended the section 4.2 to explain the distribution of packets of a flow.

A.2. Updates from version 01

Clarified what happens if one interface is attached to the home network (section 4.1.7)

Added references to the problem statement drafts of multi-homing goals and benefits in the introduction

Added the security section

A.3. Updates from version 02

Added copyright statement as defined in RFC 3667

Added new reference of 11 and updated references of 5, 3 & 9

Remove the following statement from the section 4.1.6
If the filtering agent is a CN instead of a HA, then packets will be delivered to the CoA of the point of attachment B using a Type 2 Routing Header as stated in [3] . This does not go with return routerbility procedure that is defined in [3]

Authors' Addresses

Koojana Kuladinithi
Department of Communication Networks (ComNets)
Center for Information and Communication Technology (ikom)
University of Bremen
D-28219 Bremen, Germany
Tel: +49-421-218-8264
Email: koo@comnets.uni-bremen.de

Niko A. Fikouras
Aristotle University of Thessaloniki
Dept. of Electrical and Computer Engineering
Telecommunications Division
Panepistimioupolis
P.O.Box 54124
Thessaloniki
Greece
Tel.: +302310256491
Fax.: +302310270724
Email: niko@ieee.org

Carmelita Goerg
Department of Communication Networks (ComNets)
Center for Information and Communication Technology (ikom)
University of Bremen
28219, Bremen, Germany
Tel: +49-421-218-2277
Email: cg@comnets.uni-bremen.de

Koltsidas Georgios
Aristotle University of Thessaloniki
Dept. of Electrical and Computer Engineering
Telecommunications Division
Panepistimioupolis
P.O.Box 54124
Thessaloniki
Greece
Tel.: +302310994192
Email: fractgkb@auth.gr

Fotini-Niovi Pavlidou
Aristotle University of Thessaloniki
Dept. of Electrical and Computer Engineering
Telecommunications Division
Panepistimioupolis
P.O.Box 54124

Thessaloniki
Greece
Tel.: +302310996285
Email: niovi@auth.gr

Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Disclaimer of Validity

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Copyright Statement

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.