Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: February 3, 2008

K. Mitsuya
Keio University
K. Tasaka
KDDI R&D Lab
R. Wakikawa
Keio University
R. Kuntz
University of Tokyo

August 2, 2007

A Policy Data Set for Flow Distribution
draft-mitsuya-monami6-flow-distribution-policy-04.txt

Status of this Memo

Copyright Notice

Abstract

The multiple care-of addresses registration protocol [1] allows a mobile node to maintain, at the same time, multiple virtual paths with its home agent or correspondent nodes. This document presents a

policy data set for flow distribution to be used with the multiple care-of addresses registration protocol. This policy data set defines policies, in an OS-independant manner, for a mobile node and its home agent or correspondent node, from the point of view of one of these nodes. This data set is aimed to be processed by this node and the output is a set of filter rules to be used and exchanged with its correspondents.

Table of Contents

1. Introduction

A mobile node (mobile host or router) has several network accesses to the Internet and switches between or simultaneously uses these accesses. Traffic from and to the mobile node are distributed to these accesses based on user's and network's operator policies.

The multiple care-of addresses registration protocol (MCoA) [1] provides an extension to Mobile IPv6 [2] to maintain multiple virtual paths between a mobile node and its home agent or correspondant nodes. An unique identifier named BID (Binding Unique Identification number) is assigned on each path to distinguish each of them. A binding is identified by a pair of a home address and a BID, so it is possible for a mobile node to register multiple CoAs on its peers.

The MCoA protocol only defines a way to maintain multiple paths between two nodes. However, both node have to use filtering rules to know how to distribute the traffic among these multiple paths. As the filtering decision is taken on multiple nodes (the end points of the multiple paths), it is important that the overall rules are consistent with the user's or operator's will.

We first present in this draft a policy data set that aims at defining in an OS-independent manner a way to describe filtering rules. We then explain how this data set can be used by a node to decide the filtering rules information for itself and all its peers, for to the traffic coming from or to this node.

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 [3].

The key words "Policy", "Filter Rules", "Filter", "Filtering Peer" in this document are to be interpreted as described in [4]. In addition, this document defines the following terms:

Target node: A set of filter rules can be associated to several nodes. The target node refers to the node on which the associated rules MUST be installed: a Filtering Peer or the mobile node itself.

Condition: A condition is a set of characteristics of available access networks, associated to a set of target nodes. If the condition matches, the set of filter rules for each target nodes is selected.

Policy Data Set: A policy data set is a set of conditions, target nodes, and filter rules from the point of view of the node where the conditions are tested.

3. Flow Distribution with MCoA

3.1. Scenario

The overview of our targeted flow distribution scenario is shown in Figure 1. Multiple virtual paths are maintained between two nodes (eg. a mobile node and its home agent) thanks to the multiple care-of addresses registration protocol: TUN1, TUN2, TUN3 and TUN4. Each node has its own IP filtering framework (for example IPFilter on BSD or Netfilter on Linux), described as "IPF" in the figure. Two bi-directional flows (Flow-A and Flow-B) between the Mobile node and a correspondent in the Internet are represented respectively with "***" and "~~~". Other available but not used virtual paths are represented with "===".

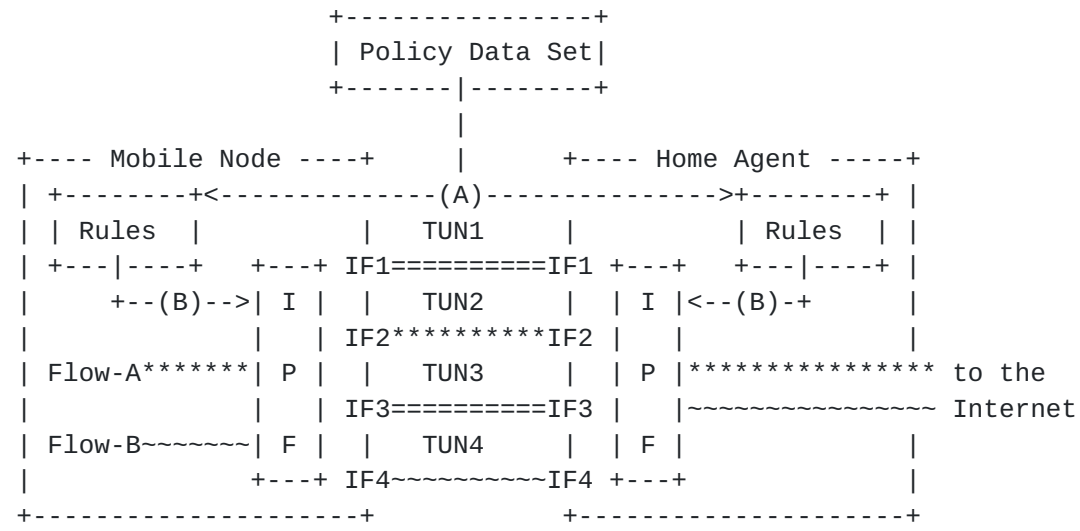

Figure 1: Flow Distribution Architecture Overview

Flows can be distributed among the available paths if proper policies are installed on the system with the IP filtering framework. The box referred to as "Policy Data Set" is a policy encoded as defined in this document Section 4. Such data set can be shipped with the product or received by using a secured transport protocol (step A) (such exchange protocol not being covered by this draft). This policy data set is then processed by the system on the host (step B) according to its available conditions. The output is a set of filter

rules for each target hosts: the rules for the local hosts can be translated and installed with the OS-specific IP filtering framework, and rules for the remote hosts can be sent using a policy enforcement mechanism, for example [5].

3.2. Use Case

User-oriented policy: When a user wants to distribute traffic amoung multiple paths, user installs a policy data set to a mobile node and sends filter rules to target nodes such as a home agent and correspondent nodes.

Operator-oriented policy: When a network operator such as the information and communication company needs to consider about management of home agent and traffic control to avoid traffic congestion, a network operator applies filter rules to target nodes such as mobile nodes. This is more effective when traffic concentrates on the base station (e.g. in case of any disaster).

If a user or an operator doesn't want to change its own policy data set and filter rules, they set deciding authority of their nodes (target nodes). Therefore, nodes wishing to send filter rules to target nodes check their deciding authority before sending policy data set and filter rules. If they want to change writing authority of target nodes, they negotiate policy data set with users or operators who uses and manages target nodes.

3.3. Architecture Overview

We assume that an IP filtering framework implemented in the operating system (such as IPFilter for BSD or Netfilter for Linux) is used to distribute the traffic among the multiple available paths maintained by MCoA. Such framework is usually tightly integrated to the system, thus no generic tool nor syntax exists to describe and install filtering rules on different operating systems. We first aim at defining a generic (ie OS-independent) grammar to define a policy data set that could be exchanged and understood by heterogeneous hosts. Such policy data set would describe filter rules based on user's or network operator's policy. We also aim at defining a framework that processes this policy data set.

We also assume that this policy data set can originally be stored either on a mobile node or its home agent (eg. when the policy data set is shipped with the product), but also on an authorized third-part node that may not have any Mobile IPv6 functionnalities. This policy data set can then be dynamically sent to the node willing to install filtering policies (as this draft focuses on the definition of the policy data set itself and its processing, such exchange

protocol will not be defined here). We thus separate binding registrations and flow distribution policy exchange, in opposition to the proposed protocols [6], [7], [8] and [9]. This allows to have a very flexible mechanism, where, for example, mobile nodes could get their policy data set from a policy database.

A policy data set includes filter rules for several hosts, classified in a set of conditions from the point of view of one node. This node processes its policy data set as follow:

- It firsts look in the list of conditions to see which one matches with its current state. In the MCoA protocol, the BID identifies in an unique manner the multiple paths between a mobile node and its peers. Therefore, this document defines the conditions as a list of the available BIDs on the node that processes the policy data set.

- Senders of conditions or filter rules check deciding authority of target nodes by using exchanging protocol before sending policy data set and filter rules.

- Each condition being associated to a list of target hosts and rules, when a condition match we obtain a list of rules to apply on several target hosts.

- If one of the target is the local host itself, it can translate the associated filter rules for its own IP filtering framework, and install them. Filter rules associated to other targets can be sent using a filter rules exchange mechanism (in the case of a Mobile Node, it could be for example [5]).

- Each time the condition changes on the host (for example, one BID is not available anymore), the host processes again the policy data set in order to select and install the most suitable filter rules for him and its peers.

4. Policy Data Set

This section defines the format used to describe a policy data set. It respects the ABNF (Augmented BNF) defined in [10] and is defined as below.

A policy data set can include policies for a set of several hosts. Each host is identified by its permanent IPv6 address (for a Mobile Node, it would be its Home Address). Each defined node has a set of available condifions, that refers to the characteristics of its available access networks. Here, the BID is used as such

characteristics. For each set of conditions, a set of rules can be defined for several target hosts. Each target host is identified with its permanent IPv6 address, or refered as "local" (that refers to the host that processes the policy data set) or "any" (any hosts the local host is binded with).

A set of rules is then defined for each host. Each rule associates some selectors (for example the source and destination address, the source and destination ports, the protocol number, etc.) with an action (the output path to choose, or drop the packets) and a lifetime.

```
   policy-set      = 1*(idaddr conditions-sets ";")
   idaddr          = ADDR
   conditions-sets = 1*(conditions target-set)
   conditions      = condition *("," condition)
   condition       = NUMBER
   target-set      = 1*(target rules)
   target          = "local" / "any" / ADDR
   rules           = 1*(flow action [lifetime])
   flow            = ["proto" protocol] [srcaddr] [dstaddr] [match]
   protocol        = [no] icmpv6 / [no] tcp / [no] udp / [no] NUMBER
   icmpv6          = "icmpv6" icmptype
   icmptype        = [no] type ["/" code]
   type            = NUMBER
   code            = NUMBER
   tcp             = "tcp" [srcport] [dstport] [tcpflags]
   udp             = "udp" [srcport] [dstport]
   srcport         = "sport" [no] ports
   dstport         = "dport" [no] ports
   ports           =  port / port ":" port / ":" port / port ":"
   port            = QSTRING / NUMBER
   tcpflags        = "flags" flags ["/" flags]
   flags           =  FLAG / flags "," FLAG
   srcaddr         = "from" [no] addr
   dstaddr         = "to" [no] addr
   addr            = "any" / host [mask]
   host            = ADDR
   mask            = "/" NUMBER / "/" HEXNUM
   match           = "match" 1*(matchitem)
   matchitem       = "hoplimit" [no] hoplimit / "tclass" [no] tclass
                      / "ip6h" [no] ip6headers
   hoplimit        = NUMBER / ":" NUMBER / NUMBER ":"
   tclass          = NUMBER / HEXNUM
   ip6headers      = ip6header *("," ip6header)
   ip6header       = NUMBER
   action          = "bid" NUMBER / "drop"
   lifetime        = "lft" NUMBER
   no              =  "!" / "not" / "no"

   addr1           = 1*4HEXDIG ":" *(1*4HEXDIG":") 1*(":" 1*4HEXDIG)
   addr2           = 1*4HEXDIG *6(":" 1*4HEXDIG) "::"
   addr3           = 7*7(1*4HEXDIG ":") 1*4HEXDIG
   ADDR            = addr1 / addr2 / addr3 / "::" / "::1"
   QSTRING         = ALPHA *(ALPHA / DIGIT / "-")
   NUMBER          = 1*DIGIT
   HEXNUM          = "0x" 1*HEXDIG
   FLAG            = "F" / "S" / "R" / "P" / "A" / "U"
```

For example, a Mobile Node has the 2001:db8::1000 Home Address and is

registered to its Home Agent whose address is 2001:db8::2000. One policy data set defines the policies for both the MN and its HA, from the MN point of view: the first field ("2001:db8::1000") defines the destination of the policy data set (here, the MN).

```
 2001:db8::1000
     11,800
     local
         proto tcp sport 80 to any bid 800
         from 2001:db8::1000 to any bid 11
     2001:db8::2000
         proto tcp dport 80 to any bid 800
         from any to 2001:db8::1000 bid 11
     11
     local
         proto tcp sport 80 to any drop
         from 2001:db8::1000 to any bid 11
      2001:db8::200
         proto tcp dport 80 to any drop
         from any to 2001:db8::1000 bid 11
```

This mobile node can register two Care-of Addresses whose BIDs are 11 and 800. When both CoAs are available (ie. when conditions "11,800" matches), the policies are defined as follow:

For the MN ("local") the http traffic ("proto tcp sport 80 to any") is sent via the path binded to the BID 800 ("bid 800") and all other traffic ("from 2001:db8::1000 to any") is sent via the path binded to the BID 11 ("bid 11").

For the HA ("2001:db8::2000"), symetric policies are defined ("proto tcp dport 80 to any bid 800" and "from any to 2001:db8::1000 bid 11").

If only the CoA whose BID is 11 is available (condition "11" matches), the policies are defined as follow:

For the MN ("local") the http traffic ("proto tcp sport 80 to any") is dropped ("drop") and all other traffic ("from 2001:db8::1000 to any") is sent via the path binded to the BID 11 ("bid 11").

For the HA ("2001:db8::2000"), symetric policies are defined ("proto tcp dport 80 to any drop" and "from any to 2001:db8::1000 bid 11").

The host 2001:db8::1000 may have received this policy data set dynamically using a secured transport protocol (such as SFTP, HTTPS,

etc.), or it may have been shipped with it. The host processes this data set according to its available BIDs. It then installs the "local" rules on its own system, and send the rules specific to its home agent using a policy exchange mechanism (e.g [5]).

5. Changes from Previous Revisions

Version 04 change:

- Added the Use-case section
- Update the Architecture Overview section

Version 03 change:

- Clarified our architecture
- Removed the example of the dynamic policy exchange
- Improved the Policy Data Set
- Added the new author!

Version 02 change:

- Changed from the xml-based format to a BNF format.

Version 01 change:

- Clarified the meaning of "Direction".
- Added how to specify a range of address or port.
- Mentioned that any nodes can also be Initiator/Responder.

6. Acknowledgment

The authors would like to thank Manabu Tsukada for his comments. The authors would also like to thank Shigeyuki Akiba, Masatoshi Suzuki and Hiroki Horiuchi for their support and assistance.

7. References

[1] Wakikawa, R., "Multiple Care-of Addresses Registration", draft-ietf-monami6-multiplecoa-01 (work in progress),

October 2006.

[2] Johnson, D., Perkins, C., and J. Arkko, "Mobility Support in IPv6", RFC 3775, June 2004.

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

[4] Larsson, C., "A Filter Rule Mechanism for Multi-access Mobile IPv6", draft-larsson-monami6-filter-rules-01 (work in progress), October 2006.

[5] Soliman, H., "Flow Bindings in Mobile IPv6", draft-soliman-monami6-flow-binding-03 (work in progress), October 2006.

[6] Montavont, N., "Home Agent Filtering for Mobile IPv6", draft-montavont-mobileip-ha-filtering-v6-00 (work in progress), December 2003.

[7] Soliman, H., "Flow Movement in Mobile IPv6", draft-soliman-mobileip-flow-move-03 (work in progress), June 2003.

[8] Kuladinithi, K., "Filters for Mobile IPv6 Bindings", draft-nomadv6-mobileip-filters-03 (work in progress), October 2005.

[9] Wakikawa, R., "Multiple Network Interfaces Support by Policy-Based Routing on Mobile IPv6", Proceedings the International Conference on Wireless Networks (ICWN), July 2002.

[10] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 4234, October 2005.

Authors' Addresses

Koshiro Mitsuya
Keio University
5322 Endo
Fujisawa, Kanagawa 252-8520
Japan

Phone: +81 466 49 1100
Email: mitsuya@sfc.wide.ad.jp
URI: http://www.sfc.wide.ad.jp/~mitsuya/

Kazuyuki Tasaka
KDDI R&D Laboratories Inc.
2-1-15 Ohara
Fujimino, Saitama 356-8502
Japan

Phone: +81 49 278 7574
Email: ka-tasaka@kddilabs.jp

Ryuji Wakikawa
Keio University
5322 Endo
Fujisawa, Kanagawa 252-8520
Japan

Phone: +81 466 49 1100
Email: ryuji@sfc.wide.ad.jp
URI: http://www.wakikawa.org/

Romain Kuntz
The University of Tokyo
Japan

Phone: +81 445 80 1600
Email: kuntz@sfc.wide.ad.jp
URI: http://www.sfc.wide.ad.jp/~kuntz/

Full Copyright Statement

Intellectual Property

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).