Network Working Group
Internet-Draft
Expires: April 16, 2005

M. Bagnulo
UC3M
J. Arkko
Ericsson

October 16, 2004

Functional decomposition of the M6 protocol draft-bagnulo-multi6dt-functional-dec-00

Status of this Memo

This document is an Internet-Draft and is subject to all provisions of section 3 of RFC 3667. By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she become aware will be disclosed, in accordance with RFC 3668.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on April 16, 2005.

Copyright Notice

Copyright (C) The Internet Society (2004).

Abstract

In this note we will present a functional decomposition of the M6 protocol i.e. the protocol for preserving established communications in multihomed environments. We will do so by describing a protocol walkthrough, presenting which functions have to be performed at each stage and the messages required to accomplish them. The functional decomposition presented in this draft is based on the general

functional analysis of multihoming approaches presented in [3].

Table of Contents

1. Introduction

In this note we will present a functional decomposition of the M6 protocol i.e. the protocol for preserving established communications in multihomed environments. We will do so by describing a protocol walkthrough, presenting which functions have to be performed at each stage and the messages required to accomplish them. The functional decomposition presented in this draft is based on the general functional analysis of multihoming approaches presented in [3].

We will first present some possible procedures for the initial contact and session state establishment. Next, we will consider the management of the available locator set. Then, we will consider the re-homing of an ongoing communication and finally we will deal with session state removal.

This note is agnostic to the security mechanism used in protecting the control messages related to multihoming. However, when it is deemed necessary, a security analysis that attempts to understand the security required for the message exchange will be included.

2. Initial contact

2.1 Initial contact

During the initial contact, the minimum information that has to be exchanged by the two communicating nodes is: the identifiers that have to be presented to the upper layers, and a reachable locator for each node. In the case that the the identifier presented to the upper layers is a valid reachable locator, then only this address that will perform both roles has to be exchanged. If this is the case, no special M6 features are required. This means that as long as the identifier and the locator used are the same IPv6 address, there is no need to perform any special M6 exchange for the initial contact and regular IPv6 can be used. However, if an identifier that is different from the locator used for exchanging the initial packet is used, then the M6 protocol has to be used even to perform the initial contact between the two nodes, starting by the capabilities detection procedure described in section 2.1.2.

2.2 Failure during startup

In the case that the locator used for initial contact is unreachable, there are two possible approaches that can be followed:

One approach is to simply retry the initial contact using a different locator. This means that the initial contact procedure is started all over again, using a different locator. This approach is likely to be the one required when the M6-capable host is establishing communications with legacy hosts that don't support the M6 protocol. In this case, the address included in the initial packet is both the locator and the identifier and if it is not reachable, an alternative address has to be used. Such retrying would by default occur at the application layer, which is aware of the multiple addresses. It might be possible to optimize this should the transport protocol be made aware of the multiple addresses.

Another approach is to change locator while preserving the identifier. This approach is not supported by legacy IPv6 nodes, so the M6 protocol has to be used in this approach. However, this scheme would allow to recover from failures on locators used for initial contact in a upper layer transparent fashion. However, if this is the case, the full M6 protocol has to be used for initial contact, starting by the capabilities detection procedure described next.

3. Capabilities detection and M6 host-pair context establishment

When a M6 capable host desires to obtain the enhanced features provided by the M6 protocol in the communications established with a given peer node, it has to first determine if M6 capabilities are available in the peer node and second establish a M6 host-pair context, as defined in [5].In this section we will analyze the different mechanisms to perform both actions.

3.1 Capabilities detection

This section presents a number of possible approaches for the capability detection, and analyzes their properties.

3.1.1 Node Configuration

A simplistic approach would be to require the M6 capability from peers, if a node supports M6 and has been configured to use it. However, this would severely limit the ability of the node to communicate until the feature is widely supported.

3.1.2 DNS Configuration

A better configuration-based approach would be to add some information to the DNS to tell the peers whether the target node supports M6 or not. For instance, a new RR record could be used. This way each node could dynamically decide whether it can run M6 with the peer or not. This could often be known before actually attempting to communicate.

This approach requires that every node that wishes to use M6 must have a DNS entry. In addition, the ability to use M6 and the information stored to the DNS may not always be synchronized. For instance, changing the operating system might remove M6 capability from a particular user's machine, leading to a need for updating DNS. This synchronization problem could be avoided by the use of Dynamic DNS updates -- with the implied requirements for setting up a security association between the DNS servers and client machines.

Similarly, the manually updated DNS approach requires support for Dynamic DNS [2] where RFC 3041 [1] or other dynamically changing addresses are used.

Finally, all DNS-based approaches suffer from an an administrative split between the actual nodes and the ones storing data about them; establishing Dynamic DNS or manual updates may be hard for a private subscriber of a large operator, for instance.

On the other hand, a DNS-based mechanism may work well if the chosen Multi6 protocol is based on the use of DNS, such as in NOID [4].

3.1.3 Host-Based Dynamic Discovery

A host-based mechanism discovers the M6-capability directly between the communicating nodes. Two variants of this mechanism exist:

Independent

This mechanism adds a message pair for the discovery. If the peer responds to the message that the initiator sends, then both nodes know that they support M6.

Integrated

This mechanism uses the rest of the M6 signaling, doing both actual M6 work and capability detection at the same time.

In the interest of reducing number of initial communications latency, the second approach would be preferrable. We also argue that a M6 protocol MUST do an initial (or at least an early) signaling exchange in any case. This is because the nodes need to discover alternate locators PRIOR to a multihoming event disabling the current ones. For instance, lets assume hosts A and B communicate over two separate links without going through the Internet. Lets further assume the nodes use plain IPv6 at the beginning without M6, and use one of the links and its prefix P for communication, with addresses P::A and P::B. If this link goes down, the obvious multihoming solution would be to switch to Q::A and Q::B, the other link and its prefix. However, neither side is aware that the other node is available under the Q prefix, so communications can not continue.

On the other hand, the integrated approach makes the initial packets larger which is a disadvantage when the peer does not support multihoming. However, we do not expect the M6 part of the initial packets to be large or contain many addresses on the average, so this seems like a good engineering tradeoff.

3.1.4 Timing

Capability detection needs to occurr prior to or at the same time as Multi6 state is created between peers. If the capabilities are stored in DNS, a convenient time to look this information is at the time a DNS query is made (even if it may not always lead to an actual communication attempt later). If host-based discovery is used, the

best time to perform it is in the initial Multi6 exchange packets.

The capabilities of the peer must be remembered while the M6 state exists; the state persistence is discussed in Section 5 of [4]. The capabilities should preferrably be cached even beyond this, in order to avoid discovering the capabilities and/or locators of peers when they are contacted again within a small time frame.

Negative caching should be used to remember the peers which do not support multihoming. Depending on the type of the chosen capability detection mechanisms, this state is indexed either by an IP address or by both IP addresses and identifiers. The latter becomes necessary if DNS configuration indicates an identifier and Multi6 cabability, but the node refuses to communicate using M6.

3.2 M6 host-pair context establishment

Once that the the M6 protocol support is confirmed, the ULP identifiers associated to the M6 host-pair context need to be defined.

With respect to locator exchange, it is clear that at least one locator (the one included in the source address field of the packet) is exchanged in the initial contact. The question would be if additional locators are needed to be exchanged during the M6 host-pair context establishment phase. Several considerations should be taken into account at this point: On one hand, the capability of recovering from an outage may depend on knowing alternative locators of the other node. It is clear that a node is aware of its own locators, so a possible approach would be that if a failure is detected, the node simply tries to communicate using an alternative source locator. Since both nodes behave this way, a failure in any single locator used in the communication can be recovered. However, in the case that both locators used in the communication fail simultaneously, the approach of trying different source addresses will not be enough to restore the communication. This is basically because retrying with a different source address assumes that at least on of the original locators is working. So, in order to be able to provide fault tolerance support in the situations when both locators fail simultaneously, it is required that at least one of the nodes is aware of multiple locators of its correspondent node. On the other hand, exchanging alternative locator information imposes an additional overhead in the communication, which is only useful if an outage occurs (which is supposed not to be so frequent).

In conclusion, adding additional locators in the M6 host-pair context establishment exchange provides enhanced fault tolerance but it imposes an additional cost. A reasonable approach would be that the

M6 host-pair context establishment exchange should support the exchange of additional locators, but should not mandate it. In addition, the M6 protocol should support the exchange additional locators during the lifetime of the session.

Besides the identifiers and locators associated to the M6 session, it may be required to negotiate Context Tags that allow to identify data packets that belong to that M6 host-pair context. The need of such Context Tags depends on the demultiplexing mechanism used, as described in [5]

3.2.1 Security Information

In addition, it seems that security related information needs to be exchanged during the M6 host-pair context establishment exchange.

Including a sort of cookie/key/hash chain anchor in the exchange, limits the potential attackers to those present in the path during this initial exchange. It also implies that in order to complete the exchange, the other node must be receiving the reply packets. In addition, such key would be useful to secure future exchanges. So, it seems a good option to exchange a shared secret during the M6 host-pair context establishment exchange.

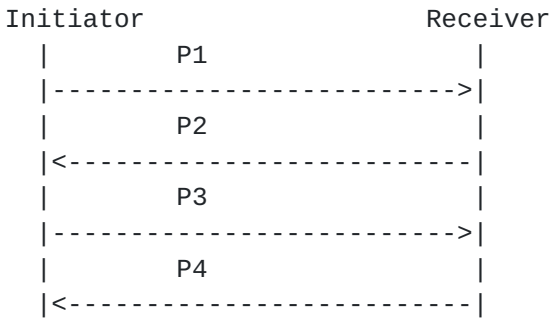
The exchange of additional security information may be required to provide protection against future attacks, depending on the security scheme used.

3.2.2 DoS protection.

Depending on the mechanism used to provide protection against future attacks, the M6 host-pair context establishment exchange may more or less susceptible to DoS attacks. If the security mechanism used requires a considerable amount of processing, it may be used to launch a DoS attack consuming the processing power of the receiver. In addition, after the exchange is completed, a state is created in each node, storing information about identifiers, multiple locators, keys and so on. Such state requires memory, so it can be used by a malicious node to generate a DoS attack based in memory consumption. In order to provide some protection from these attacks, the receiver node should not create any state until the initiating node has done so. This can be achieved by using a 4 way exchange, where the receiver does not create any state until the third packet and the initiator has to prove that it has some state created after receiving the second packet. This can be done by the initiator by showing some information received in the second packet and that the receiver can regenerate without requiring some specific information (like hashing a key and the initiator address). The result is not a complete

protection from such attacks, since the resulting state in the receiver is long lived, and the attacker can discard its state after finishing the 4 way handshake. However, after the 4 way handshake, the receiver will know a valid locator of the attacker, which can be used to track the attacker.

3.2.3 Resulting M6 host-pair establishement exchange



P1 is essentially a request to initiate an exchange. It will also be used to detect the M6 capability of the receiver.

P2 will provide the information needed by the initiator to prove that he has some state about this communication. So, P2 will contain something like a Hash of a secret key of the receiver (common to all initiators) and the initiator's address. The receiver will receive P1 and generate P2 without creating any state specific to this communication. It would be possible to also include the alternative locators of the receiver in P2. The question about this if this couldn't be used as an amplifier to launch other DoS attacks to third parties.

P3 will contain the Hash obtained in P2. In addition, it will contain the identifier used for this communication and it may contain alternative locator information. In addition, information to validate the locator set may be included. Finally, the key/cookie/hash chain anchor related information is also included.

P4 serves as an acknowledgment of the information received in P3 and it also includes information about alternative locators, identifier and key/cookie/hash chain anchor related information that hasn't been exchanged yet.

4. Locator set management

The management of the locator set involves adding new locators and removing existing locators.

The reasons for adding a new locator are pretty straightforward: there is an additional locator that the holder node wants to add to the available set. The reasons for that may be that a new locator is available in the node (e.g. a mobile node) or simply that the node wants to obtain enhanced fault tolerance by adding additional locators to the available set.

The reasons for deleting an existing locator are essentially local. Nodes have local information about the status of their own locators. In case that one of the locators becomes unavailable for some reason (e.g. it is deprecated through Router Advertisement) it would make sense to inform the other nodes that this locator should no longer be used.

There are two possible approaches to the addition and removal of locators: atomic and differential approaches. Atomic approaches essentially send the complete locators set each time that a variation in the locator set occurs. In this case, there is only one message exchange defined i.e. a message that informs about the new locator set and an acknowledgment message. Differential messages send the differences between the existing locator set and the new one. In this case, a message for adding a new locator and another message for deleting locators have to be defined. Both messages can be acknowledged. The atomic approach imposes additional overhead, since all the locator set has to be exchanged each time while the differential approach requires re-synchronization of both ends through changes i.e. that both ends have the same idea about what the current locator set is.

4.1 Security of the exchange for adding locators

The additional locators conveyed through this mechanism should belong to the node that performed the initial exchange. Security information must be included in this messages to prove that. One possibility is to include information about the key/cookie/hash chain defined in the initial exchange. This is not enough to prevent future attacks. In order to provide future attack protection, alternative schemes like HBA or CGAs has to be used.

4.2 Security of the exchange for deleting locators

The security required for the message for removing a locator may be achieved using the key/cookie/hash chain information created during

the initial contact exchange.

5. Re-homing procedure

The re-homing procedure occurs when a new locator pair is to be used for the communication. The reason for a re-homing is essentially that the current locator pair is no longer working. The re-homing procedure involves detecting that the locator pair currently in use is no longer working, exploring alternative locator pairs and re-homing the communication to the reachable locator pair.

In order to verify that a locator pair is working, a Reachability Test exchange is needed. This can be used to check if the locator pair that is being used is working properly or to explore if potential locator pairs are working. In addition, in the last case, the Reachability Test is also a mechanism to prevent thrid-party flooding attacks.

The Reachability Test exchange includes the following packets:

Reachability Test (RT) packet: including the random nonce and maybe information related to the initial key/cookie/hash chain

Reachability Test Reply (RTR) packet: include the nonce of the RT packet and maybe information related to the initial key/cookie/hash chain

In the case that a bidirectional path is available, the pair of locators contained in the RT and RTR packets will be the same. However, if only two disjoints unidirectional paths are available, the locators contained in the RT will differ from the ones included in the RTR. Additional discussion on this topic can be found in [6].

5.1 Security

In any case, it is required to know if there is a node replying at the other end. The messages should include a nonce in order to match the replies with original messages. In addition, the nonce should be randomly selected, imposing the reception of the initial message to be able to properly generate the reply. Finally, including information related to the key/cookie/hash chain defined in the initial exchange would guarantee that only the node involved in the initial exchange can participate in the reachability exchange (depending on the particular mechanism, this may be more or less expensive) In the case that the mechanism is used to prevent third-party flooding attacks additional security measures are needed, since any M6 capable host would reply to a Reachability Test request, precluding its usage as flooding attack prevention. So, in order to use this mechanism to prevent flooding, additional checks are needed. One option would be to include information related to the

key/cookie/hash chain defined in the initial exchange as described above. Another option would be to require that nodes only reply Reachability Test requests coming from nodes that they are already communicating with.

6. Removal of M6 session state

There are essentially two approaches for discarding an existing state about locators, keys and identifiers of a correspondent node: a coordinated approach and an unilateral approach.

In the unilateral approach, each node discards the information about the other node without coordination with the other node based on some local timers and heuristics. No packet exchange is required for this. In this case, it would be possible that one of the nodes has discarded the state while the other node still hasn't. In this case, an error message may be required to inform about the situation.

In the coordinated approach, there is a closing exchange that is performed in order to coordinate the process, in order to make sure that both nodes discard the state related to the previous communication. This would require a pair of messages Close and Close-ACK.

6.1 Security

The Close message has to be secured using the key/cookie/hash chain information created during the initial contact exchange.

7. Security considerations

The security requirements of each message exchange considered in this note are detailed in the same section where the message exchange is analyzed.

8. Contributors

This document was originally produced of a MULTI6 design team consisting of (in alphabetical order): Jari Arkko, Marcelo Bagnulo Braun, Iljitsch van Beijnum, Geoff Huston, Erik Nordmark, Margaret Wasserman, and Jukka Ylitalo.

9. References

9.1 Normative References

[1] Narten, T. and R. Draves, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 3041, January 2001.

[2] Wellington, B., "Secure Domain Name System (DNS) Dynamic Update", RFC 3007, November 2000.

9.2 Informative References

[3] Huston, G., "Architectural Approaches to Multi-Homing for IPv6", draft-huston-multi6-architectures-01 (work in progress), June 2004.

[4] Nordmark, E., "Multihoming without IP Identifiers", draft-nordmark-multi6-noid-02 (work in progress), July 2004.

[5] Nordmark, E. and M. Bagnulo, "Multihoming L3 Shim Approach", draft-nordmark-multi6dt-shim-00.txt (work in progress), October 2004.

[6] Arkko, J., "Failure Detection and Locator Selection in Multi6", draft-multi6dt-failure-detection-00 (work in progress), October 2004.

Authors' Addresses

Marcelo Bagnulo
Universidad Carlos III de Madrid
Av. Universidad 30
Leganes, Madrid 28911
SPAIN

Phone: 34 91 6249500
EMail: marcelo@it.uc3m.es
URI: http://www.it.uc3m.es

Jari Arkko
Ericsson
Jorvas 02420
Finland

EMail: jari.arkko@ericsson.com

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

Copyright (C) The Internet Society (2004). This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

Acknowledgment

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