

Nortel Networks
Multi-link Multi-node PPP Bundle Discovery Protocol

Status of this Memo

This memo provides information for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (1999). All Rights Reserved.

Abstract

This document specifies a standard way for Multi-link PPP to operate across multiple nodes. Both the mechanism by which the Bundle Head is discovered and the PPP fragment encapsulation are specified.

Acknowledgements

I would like to thank Joe Frazier for filling in some of the details and reviewing this document.

1. Introduction

Multi-link PPP [MP] allows a dial-in user to open multiple PPP connections to a given host. In general, this is done on an on-demand basis. That is, a secondary link, or multiple secondary links, are established when the data load on the primary link, and any previously established secondary links, nears capacity. As the load decreases, the secondary link(s) may be disconnected.

Many dial-in hosts which support multi-link PPP dial the same phone number for all links. This implies that there exists a rotary at the Point Of Presence (POP) which routes incoming calls to a bank of modems. These may be physically independent modems connected to Remote Access Server (RAS) and a rotary of analog phone lines, or a RAS with internal modems connected to analog lines or a T1/E1 or T3/E3 channel. In any case, a given RAS can only handle just so many simultaneous connections. A typical POP may need to support hundreds of connections, but no RAS today can handle that many. This creates a problem when a user's primary PPP connection is established to one

RAS in a POP and a secondary connection is established to another. This may occur because the first RAS has no available modems, or because incoming calls are assigned to ports in a round-robin fashion, for example, and the second call is simply assigned to another RAS.

The solution to this problem is to provide a mechanism by which a RAS can determine if a Multi-link PPP connection is a primary or secondary and, if a secondary, where the Bundle Head (the process within a RAS which reassembles the PPP fragments transmitted over the primary and secondary links) resides. If the Bundle Head resides on a different RAS, a protocol must be used to transfer the PPP fragments to the RAS containing the Bundle Head so that the PPP frame can be reassembled.

Section 2 of this document specifies the Discovery Mechanism. Section 3 specifies the Transfer Protocol. Section 4 specifies the configuration parameters needed for the Discovery Protocol.

2. Bundle Head Discovery Mechanism

When a user dials into a RAS and negotiates Multi-link PPP (MP) during the Link Control Protocol (LCP) phase, the RAS must determine which one of the following three cases exists:

- 1- This is the primary (first) link of the MP connection. In this case, the RAS should create the Bundle Head.
- 2- This is a secondary link of the MP connection and the Bundle Head resides on this RAS. In this case, the RAS should add the link to the Bundle (standard MP).
- 3- This is a secondary link of the MP connection and the Bundle Head resides on a different RAS. In this case, the RAS should establish a path (see section 3) to the RAS that has the Bundle Head, and use that path to transfer MP fragments.

In operation, a RAS will make the determination for case 2 first (because it is the easiest and requires no communication with other RASes. If the Bundle Head is not local, the Discovery Protocol is used to determine where the Bundle Head is, if it exists at all.

2.1 Packet Format

See "IANA Considerations" (section 6) for UDP port number assignment.

A Discovery Message has the following format:

```

+-----+-----+-----+-----+-----+
| type | length | random ID | hash | endpoint ID |
+-----+-----+-----+-----+-----+

```

where:

type - 2 octets

Message type: 1-query, 2-response.

length - 2 octets

The length (in octets) of the endpoint ID.

Random ID - 4 octets

A random identifier generated by the RAS used to resolve contention. See "Contention Handling" (section 2.4) for the use of this field.

hash - 2 octets

The unsigned sum (modulo 2^{16}) of the unsigned octets of the Endpoint ID. A value of zero indicates that no hash has been generated. See "Endpoint Identifier Matching" (section 2.2) for the use of this field.

endpoint ID - variable length

The endpoint identifier of the connection. From the discovery protocol's point of view, this is an opaque value. However, to ensure multi-vendor interoperability, the format of this field must be defined. The descriptions of, and legal values for, the fields in the endpoint ID are defined in [MP].

remote EPD class	remote EPD length	remote EPD data	local EPD class	local EPD length	local EPD data	user name length	user name data

Notes:

EPD = EndPoint Discriminator.
 remote = dial-in host.
 local = RAS.
 class and length fields are 1-octet in length.
 data fields are of variable (including zero) length.

The MP protocol requires that the RASes all have the same Local EPD. For MMP, this implies that a RAS may not use its IP or Ethernet address as an EPD. This also implies that all RASes on a rotary must have the same EPD. RASes on different rotaries may share different EPDs. The Local EPD is included in the endpoint identifier to ensure that RASes on different rotaries, but sharing a common Ethernet, will not join a particular discovery if the Remote EPDs just happen to be the same.

Except for unicast Response Messages, all messages are sent to the multicast address specified in "IANA Considerations". If a system cannot send multicast messages, the limited broadcast address (255.255.255.255) should be used.

2.2 Endpoint Identifier Matching

Comparing Endpoint IDs can be time consuming. First, the classes of the EPDs must be determined, then the values compared. These comparisons might be fast arithmetic compares or slow octet-wise compares of 20-octet long values. To improve performance, because the protocol is time-driven, the hash field may be used for a fast comparison.

When a Bundle Head is created, the hash is created and stored along with the Endpoint ID. When a Query or Response Message is generated, the hash is created and stored in the message. When a RAS receives a message, it can do a quick comparison of the hash in the message to the hashes in its tables. If a hash does not match, the Endpoint ID cannot match. However, if a hash does match, the Endpoint IDs must be properly compared to verify the match.

Obviously, there is a cost associated with creating the hashes, but they are created only once per message and once for each Bundle Head creation. However, the comparisons occur multiple times in multiple RASes for each new secondary connection. Therefore, there is a net savings in processing.

2.3 Protocol Operation

Throughout this section, configurable variables are specified by their names (e.g., ROBUSTNESS refers to the number of transmits).

The Discovery Protocol begins by multicasting ROBUSTNESS Query Messages at QUERY_INTERVAL intervals. If no Response Message for that Request is received within QUERY_INTERVAL of the last broadcast (a total time of ROBUSTNESS * QUERY_INTERVAL), the RAS assumes that this is the primary link and begins to build the Bundle Head. It then sends a multicast Response Message (in case another link comes up after the time-out but before the Bundle Head is built). If a Response Message is received (i.e., a Bundle Head exists on another RAS), no additional Query Messages are sent and the RAS establishes a path to the RAS containing the Bundle Head.

If a RAS receives a Query Message for an MP connection for which it has the Bundle Head, it sends a unicast Response Message to the querier. Note that no repetition of the Response Message is necessary because, if it is lost, the querier's next query message will trigger a new Response Message.

2.4 Contention Handling

If, while sending Query Messages, a Query Message for the same MP connection is received, it indicates that the Dial-in Node has brought up multiple links simultaneously. The resolution to this contention is to elect the bundle head. To do this, each RAS waits until all Query Messages are sent (ROBUSTNESS * QUERY_INTERVAL). At that time, the RAS with the lowest Random ID becomes the Bundle Head. If two or more RASes have the same Random ID, the RAS with the lowest IP address becomes the Bundle Head. That RAS then sends TWO Response Messages, with a QUERY_INTERVAL interval, and indicates to the MP process that a Bundle Head should be formed. When the other RAS(es) receive the Response Message, they cease broadcasting (if they haven't already sent ROBUSTNESS Query Messages), stop listening for additional Response Messages, and indicate to their respective MP processes where the Bundle Head resides.

Note that a RAS generates a Random ID for each connection and uses that value for all Query and Response messages associated with that connection. The same Random ID must not be reused until it can be

guaranteed that another RAS will not mistake the message for an old message from a previous connection. For this reason, it is recommended that the Random ID be either monotonically increasing or a clock value (either time since boot or time of day).

2.5 MP Operation

MP must use the following algorithm to ensure that there are no windows of vulnerability during which multiple Bundle Heads might be created for the same MP connection.

When an MP link is negotiated, MP first checks to see if it already has the Bundle Head for this connection (i.e., is this a secondary link). If it does, it should attach to it and not initiate a discovery. As an optimization, if MP does not have a Bundle Head for this connection, but does have an existing secondary link for it, MP should attach to the known Bundle Head without initiating discovery.

If MP knows of no Bundle Head for this connection, it should initiate a discovery. If the discovery should locate a Bundle Head, it should attach to the indicated bundle head. If no Bundle Head is found, MP should create a Bundle Head.

When a RAS determines that it is to become the Bundle Head for a connection, it should establish the Bundle Head as quickly as possible so Query Messages for that connection from other RASes will be recognized. If a RAS indicates that it will become the Bundle Head, but delays establishment of it, other RASes may time out on their discovery and begin to establish additional Bundle Heads of their own.

3. Transfer Protocol

The Layer 2 Tunneling Protocol (L2TP) [L2TP] will be used to transfer PPP fragments from a RAS containing a secondary link to the RAS containing the Bundle Head. By specifying the use of an existing protocol, it is neither necessary to create nor implement a new protocol.

4. Configuration

There are two required configuration switches and one conditional configuration switch. None of the switches are optional.

4.1 Robustness - required

This switch sets the number of transmits (repetitions) for Query Messages. It may be set between 1 and 15. The default is 3. Be aware that lower settings may create windows of vulnerability. Higher settings may cause MP timeouts, but may be needed on very lossy or congested networks.

4.2 Query Interval - required

This switch sets the interval between Query Messages and the interval between multicast Response Messages. It should be calibrated in deciseconds (1/10 second) and may be set between 1 and 15. The default is 1. Be aware that higher settings may cause MP timeouts, but may be needed on very slow systems/networks.

4.3 TTL - conditional

This switch sets the IP Time-To-Live (TTL) of all Discovery packets. For systems which are using the limited broadcast address, this switch should not be implemented and the TTL should be set to 1. The default value should be 1.

5. Security Considerations

No security is designed into the Discovery Mechanism. When not forwarding multicast packets (or when using the limited broadcast address), the discovery packets are restricted to a single LAN. If the LAN is physically secure, there is no need for software security. If the multicast packets are forwarded, but the range is limited to a small, physically secure network (e.g., a POP), there is still no need for software security. If the discovery packets are allowed to cross an internet (and this is NOT recommended for timing reasons), authentication of RASes may be done with IPSEC. For increased security on a LAN, or in a POP, IPSEC may still be used.

L2TP security is discussed in [L2TP].

6. IANA Considerations

UDP port number: 581
Multicast address: 224.0.1.69

7. References

- [MP] Sklower, K., Lloyd, B., McGregor, G., Carr, D. and T. Coradetti, "The PPP Multilink Protocol (MP)", RFC 1990, August 1996.
- [L2TP] Townsley, W., Valencia, A., Rubens, A., Pall, G., Zorn, G. and B. Palter, "Layer Two Tunneling Protocol "L2TP"", RFC 2661, August 1999.

Author's Address

Gary Scott Malkin
Nortel Networks
11 Elizabeth Drive
Chelmsford, MA 01824-4111

Phone: +1 (978) 250-3635
Email: gmalkin@nortelnetworks.com

Full Copyright Statement

Copyright (C) The Internet Society (1999). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS 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.

Acknowledgement

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

