Network Working Group
Request for Comments: 5356
Category: Experimental

T. Dreibholz
University of Duisburg-Essen
M. Tuexen
Muenster Univ. of Applied Sciences

September 2008

Reliable Server Pooling Policies

Status of This Memo

Abstract

This document describes server pool policies for Reliable Server Pooling (RSerPool) including considerations for implementing them at Endpoint Handlespace Redundancy Protocol (ENRP) servers and pool users.

Table of Contents

1. Introduction

The protocols defined in [RFC5353], [RFC5352], and [RFC5354] support a variety of server policies. Some of the policies use dynamic load information of the pool elements and others do not. Therefore, we classify them as adaptive and non-adaptive. The selection of the pool element is performed by two different entities, the ENRP server and the pool user. Some of the consequences for policies that are not stateless are described in [ICN2005] and [LCN2005].

Therefore, this document describes not only packet formats but also gives a detailed description of the procedures to be followed at the ENRP servers and the pool users to implement each server policy.

2. Conventions

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 [RFC2119].

3. Terminology and Definitions

3.1. Load

The term load is a value specifying how much a pool element's resources are currently utilized. 0x00000000 states that the pool element is not utilized (0%); 0xffffffff states that it is fully utilized (100%). Defining what utilization means is application-dependent and out of the scope of RSerPool. However, it is required that all pool elements of the same pool using load information have the same definition of load.

For example, load may define the current amount of users out of a maximum on an FTP server, the CPU usage of a database server, or the memory utilization of a compute service.

3.2. Weight

Weight defines a pool element's service capacity relative to other pool elements of the same pool. Theoretically, there is no upper limit for weight values (although limited by datatype size). Defining what value weights compare is application-dependent and out of the scope of RSerPool. However, it is required that all pool elements of the same pool using weight information have the same definition of weight.

A weight of 0 denotes that the pool element is not capable of providing any service; a higher weight denotes that the pool element is capable of providing better service than a pool element having a lower weight.

For example, weight may define a compute service's computation capacity. That is, a pool element of weight 100 will complete a work package in half the time compared to a pool element of weight 50.

4. Non-Adaptive Policies

4.1. Round Robin Policy

4.1.1. Description

The Round Robin (RR) policy is a very simple and efficient policy that requires state. This policy is denoted as the default policy and MUST be supported by all RSerPool components.

4.1.2. ENRP Server Considerations

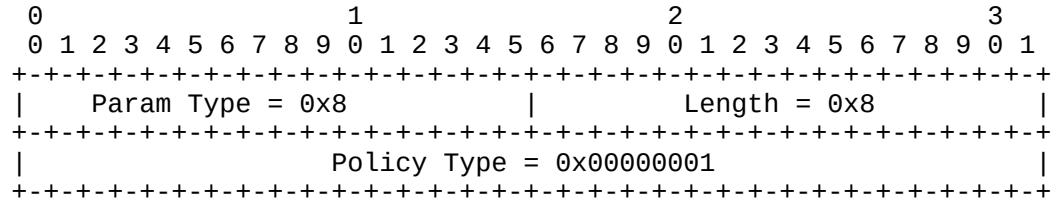
The ENRP server SHOULD hold the pool elements of each server pool in a circular list and SHOULD store a pointer to one of the elements, called the head. On reception of a handle resolution request, the ENRP server SHOULD return the pool elements from the circular list, starting with head. Then the head SHOULD be advanced by one element.

Using this algorithm ensures that not all lists presented to the pool users start with the same element.

4.1.3. Pool User Considerations

A pool user SHOULD use the list of pool elements returned by the ENRP server in a round robin fashion, starting with the first. If all elements of the list have been used, it should start from the beginning again until the information is out of date.

4.1.4. Pool Member Selection Policy Parameter



4.2. Weighted Round Robin Policy

4.2.1. Description

The Weighted Round Robin (WRR) policy is a generalization of the RR policy. If all weights are 1, then WRR is just RR.

4.2.2. ENRP Server Considerations

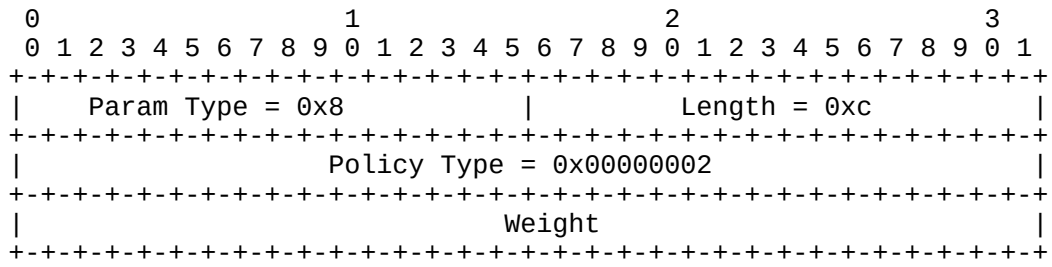
The ENRP server SHOULD follow the same rules as RR but initialize and modify the circular list differently. The ENRP server puts each pool element, possibly, multiple times into the list such that:

- The ratio of the number of occurrences of a pool element to the list length is the same as the ratio of the weight of that pool element to the sum of weights.
- The multiple entries of each pool element should be as evenly distributed as possible in the circular list.

4.2.3. Pool User Considerations

The pool user SHOULD follow the same rules as RR.

4.2.4. Pool Member Selection Policy Parameter



- Weight (32 bits, unsigned integer): Weight constant for the WRR process.

4.3. Random Policy

4.3.1. Description

The Random (RAND) policy is a very simple stateless policy.

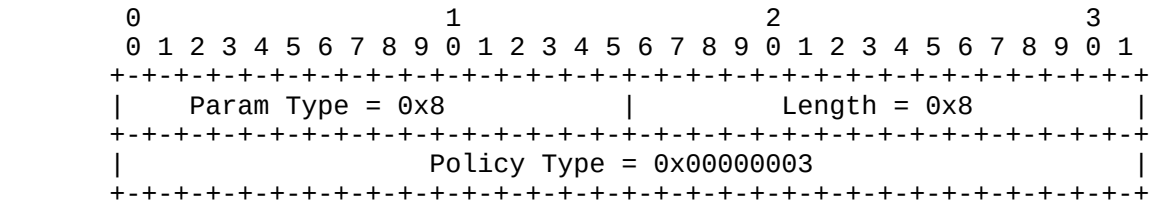
4.3.2. ENRP Server Considerations

The ENRP server selects, at most, the requested number of pool elements from the list of pool elements. Each element MUST NOT be reported more than once to the pool user.

4.3.3. Pool User Considerations

Each time the pool user must select one pool element, it does this by randomly selecting one element from the list of pool elements received from the ENRP server.

4.3.4. Pool Member Selection Policy Parameter



4.4. Weighted Random Policy

4.4.1. Description

The Weighted Random (WRAND) policy is a generalization of the RAND policy, adding a weight for each pool element entry. RAND is equal to WRAND having all weights set to 1.

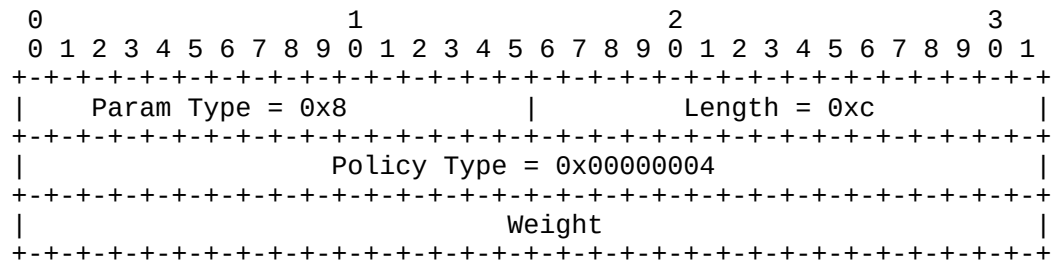
4.4.2. ENRP Server Considerations

The ENRP server SHOULD select, at most, the requested number of pool elements randomly from the list of pool elements. Each element MUST NOT be reported more than once to the pool user. The probability of selecting a pool element should be the ratio of the weight of that pool element to the sum of weights.

4.4.3. Pool User Considerations

Each time the pool user must select one pool element, it does this by randomly selecting one element from the list of pool elements received from the ENRP server.

4.4.4. Pool Member Selection Policy Parameter



- o Weight (32 bits, unsigned integer): Weight constant for the WRAND process.

4.5. Priority Policy

4.5.1. Description

The Priority (PRIO) policy can be used to select always a pool element with the highest priority.

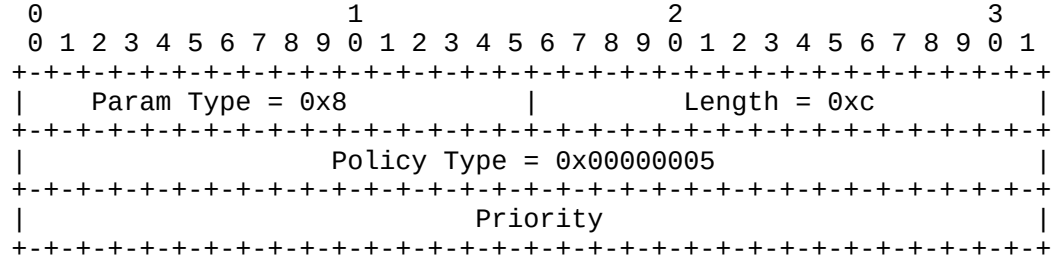
4.5.2. ENRP Server Considerations

The ENRP server MUST select the pool elements with the highest priorities. They MUST be reported in decreasing order. If multiple pool elements have the same priority, they may be listed in any order.

4.5.3. Pool Element Considerations

The pool user MUST select the active pool element with the highest priority.

4.5.4. Pool Member Selection Policy Parameter



- Priority (32 bits, unsigned integer): Larger numbers mean higher priorities.

5. Adaptive Policies

5.1. Least Used Policy

5.1.1. Description

The Least Used (LU) policy uses load information provided by the pool elements to select the lowest-loaded pool elements within the pool.

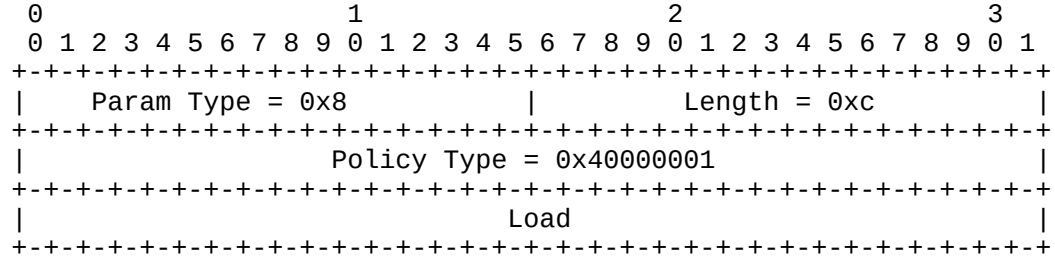
5.1.2. ENRP Server Considerations

The ENRP server SHOULD select, at most, the requested number of pool elements. Their load values SHOULD be the lowest possible ones within the pool. Each element MUST NOT be reported more than once to the pool user. If there is a choice of equal-loaded pool elements, round robin selection SHOULD be made among these elements. The returned list of pool elements MUST be sorted in ascending order by load value.

5.1.3. Pool User Considerations

The pool user should try to use the pool elements returned from the list in the order returned by the ENRP server. A subsequent call for handle resolution may result in the same list. Therefore, it is RECOMMENDED for a pool user to request multiple entries in order to have a sufficient amount of feasible backup entries available.

5.1.4. Pool Member Selection Policy Parameter



- Load (32 bits, unsigned integer): Current load of the pool element.

5.2. Least Used with Degradation Policy

5.2.1. Description

The Least Used with Degradation (LUD) policy extends the LU policy by a load degradation value describing the pool element's load increment when a new service association is accepted.

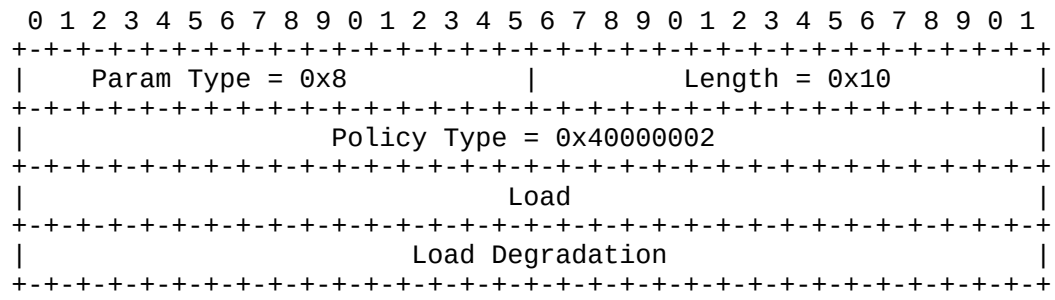
5.2.2. ENRP Server Considerations

For every pool element entry, a degradation counter MUST be stored. When a pool element entry is added or updated by registration or re-registration, this counter MUST be set to 0. When an entry is selected for being returned to a pool user, the internal degradation counter MUST be incremented by 1. The selection of pool element entries is handled like for LU, except that the selected pool element entries SHOULD have the lowest possible sum of load value + degradation counter * load degradation value.

5.2.3. Pool User Considerations

See LU policy.

5.2.4. Pool Member Selection Policy Parameter



- Load (32 bits, unsigned integer): Current load of the pool element.
- Load Degradation (32 bits, unsigned integer): Load Degradation constant of the pool element.

5.3. Priority Least Used Policy

5.3.1. Description

The Priority Least Used (PLU) policy uses load information provided by the pool elements to select the lowest-loaded pool elements within the pool under the assumption that a new application request is accepted by the pool elements. Therefore, the pool elements also have to specify load degradation information.

Example: Pool elements A and B are loaded by 50%, but the load of A will increase due to a new application request only by 10% while B will be fully loaded. PLU allows the specification of this load degradation in the policy information; the selection is made on the lowest sum of load and degradation value. That is, A will be selected (50+10=60) instead of B (50+50=100).

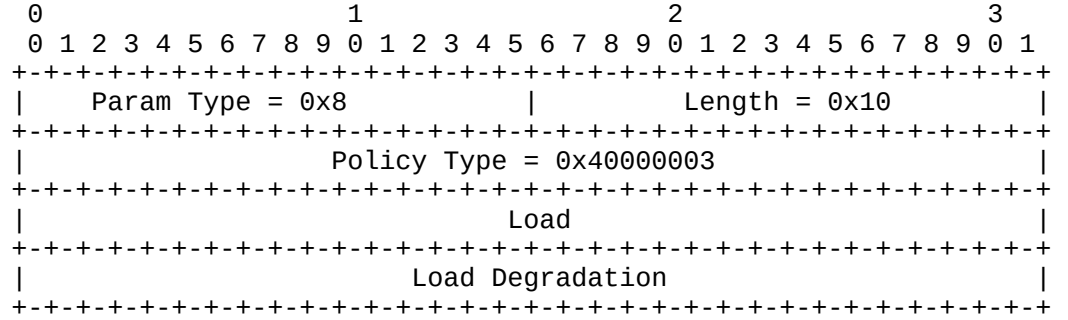
5.3.2. ENRP Server Considerations

The ENRP server SHOULD select, at most, the requested number of pool elements. Their sums of load + degradation SHOULD be the lowest possible ones within the pool. Each element MUST NOT be reported more than once to the pool user. If there is a choice of equal-valued pool element entries, round robin SHOULD be made among these elements. The returned list of pool elements MUST be sorted ascending by the sum of load and degradation value.

5.3.3. Pool User Considerations

The pool user should try to use the pool elements returned from the list in the order returned by the ENRP server. A subsequent call for handle resolution may result in the same list. Therefore, it is RECOMMENDED for a pool user to request multiple entries in order to have a sufficient amount of feasible backup entries available.

5.3.4. Pool Member Selection Policy Parameter



- Load (32 bits, unsigned integer): Current load of the pool element.

- Load Degradation (32 bits, unsigned integer): Load Degradation constant of the pool element.

5.4. Randomized Least Used Policy

5.4.1. Description

The Randomized Least Used (RLU) policy combines LU and WRAND. That is, the pool element entries are selected randomly. The probability for a pool element entry A, utilized with load_A, to be selected is (0xFFFFFFFF - load_A) / (sum(0xFFFFFFFF-load_x)), i.e., this PE's unload part related to the whole pool unload rate.

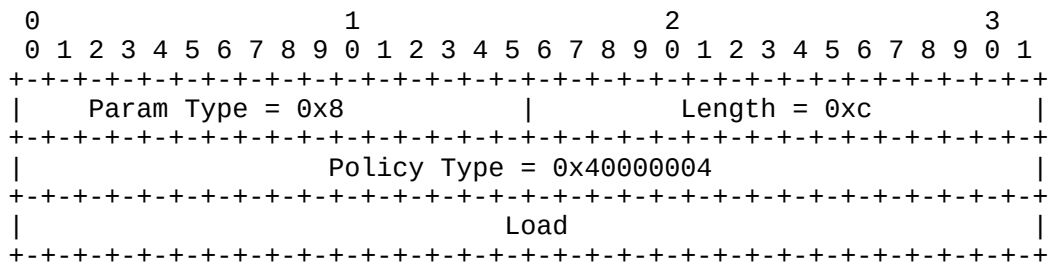
5.4.2. ENRP Server Considerations

The ENRP server SHOULD behave like WRAND, having every PE's weight set to (0xffffffff -- load value provided by the pool element).

5.4.3. Pool User Considerations

See WRAND policy.

5.4.4. Pool Member Selection Policy Parameter



- Load (32 bits, unsigned integer): Current load of the pool element.

6. Security Considerations

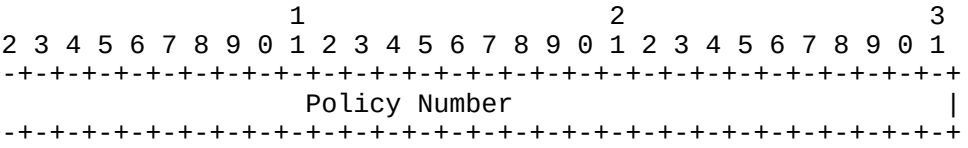
The security threats regarding RSerPool have been analyzed in RSerPool threats [RFC5355]. The server policy descriptions in this document do not add any other threats.

7. IANA Considerations

This document (RFC 5356) is the reference for all registrations described in this section. All registrations have been listed on the RSerPool Parameters page.

7.1. A New Table for RSerPool Policy Types

RSerPool policy types that are 4-byte values are maintained by IANA. The format of the policy type value is defined as follows:



- X: If set to 1, the policy is user defined and not standardized. All standards policies reserved by the IETF use X=0.
- A: If set to 1, the policy is adaptive. Otherwise, it is non-adaptive.
- Policy Number: The actual number of the policy.

Nine initial policy types have been assigned and are maintained in a new table, "RSerPool Policy Types":

Value	Policy	Reference
0x00000000	(reserved, invalid value)	RFC 5356
0x00000001	Round Robin	RFC 5356
0x00000002	Weighted Round Robin	RFC 5356
0x00000003	Random	RFC 5356
0x00000004	Weighted Random	RFC 5356
0x00000005	Priority	RFC 5356
0x00000006	(reserved by IETF)	RFC 5356
...		
0x3fffffff	(reserved by IETF)	RFC 5356
0x40000000	(reserved, invalid value)	RFC 5356
0x40000001	Least Used	RFC 5356
0x40000002	Least Used with Degradation	RFC 5356
0x40000003	Priority Least Used	RFC 5356
0x40000004	Randomized Least Used	RFC 5356
0x40000005	(reserved by IETF)	RFC 5356
...		
0x7fffffff	(reserved by IETF)	RFC 5356
0x80000000	(private use, non-standard policy)	RFC 5356
...		
0xffffffff	(private use, non-standard policy)	RFC 5356

Requests to register an RSerPool policy type in this table should be sent to IANA. The number must be unique and use the appropriate upper bits. The "Specification Required" policy of [RFC5226] MUST be applied.

The policy type space from 0x80000000 to 0xffffffff is designated for private use.

8. Reference Implementation

A reference implementation of RSerPool and the policies described in this document is available at [RSerPoolPage] and described in [Dre2006].

9. References

9.1. Normative References

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

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[RFC5354] Stewart, R., Xie, Q., Stillman, M., and M. Tuexen, "Aggregate Server Access Protocol (ASAP) and Endpoint Handlespace Redundancy Protocol (ENRP) Parameters", RFC 5354, September 2008.

[RFC5352] Stewart, R., Xie, Q., Stillman, M., and M. Tuexen, "Aggregate Server Access Protocol (ASAP)", RFC 5352, September 2008.

[RFC5353] Xie, Q., Stewart, R., Stillman, M., Tuexen, M., and A. Silverton, "Endpoint Handlespace Redundancy Protocol (ENRP)", RFC 5353, September 2008.

[RFC5355] Stillman, M., Ed., Gopal, R., Guttman, E., Holdrege, M., and S. Sengodan, "Threats Introduced by Reliable Server Pooling (RSerPool) and Requirements for Security in Response to Threats", RFC 5355, September 2008.

9.2. Informative References

[RSerPoolPage] Dreibholz, T., "Thomas Dreibholz's RSerPool Page", <http://tdrwww.iem.uni-due.de/dreibholz/rserpool/>.

[Dre2006] Dreibholz, T., "Reliable Server Pooling -- Evaluation, Optimization and Extension of a Novel IETF Architecture", Ph.D. Thesis University of Duisburg-Essen, Faculty of Economics, Institute for Computer Science and Business Information Systems, March 2007, <http://duepublico.uni-duisburg-essen.de/servlets/DerivateServlet/Derivate-16326/Dre2006-final.pdf>.

[LCN2005] Dreibholz, T. and E. Rathgeb, "On the Performance of Reliable Server Pooling Systems", Proceedings of the 30th IEEE Local Computer Networks Conference, November 2005.

[ICN2005] Dreibholz, T., Rathgeb, E., and M. Tuexen, "Load Distribution Performance of the Reliable Server Pooling Framework", Proceedings of the 4th IEEE International Conference on Networking, April 2005.

Authors' Addresses

Thomas Dreibholz
University of Duisburg-Essen, Institute for Experimental Mathematics
Ellernstrasse 29
45326 Essen, Nordrhein-Westfalen
Germany

Phone: +49-201-1837637
Fax: +49-201-1837673
EMail: dreibh@iem.uni-due.de
URI: http://www.iem.uni-due.de/~dreibh/

Michael Tuexen
Muenster University of Applied Sciences
Stegerwaldstrasse 39
48565 Steinfurt
Germany

Phone: +49-2551-962550
Fax: +49-2551-962563
EMail: tuexen@fh-muenster.de

Full Copyright Statement

Intellectual Property