

GROBJ BOF  
Internet-Draft  
Intended status: Standards Track  
Expires: April 23, 2010

B. Carpenter, Ed.  
Univ. of Auckland  
M. Boucadair  
France Telecom  
J. Halpern  
Ericsson  
S. Jiang  
Huawei Technologies Co., Ltd  
K. Moore  
Network Heretics  
October 20, 2009

**A Generic Referral Object for Internet Entities**  
**draft-carpenter-behave-referral-object-01**

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

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 23, 2010.

Copyright Notice

Copyright (c) 2009 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents in effect on the date of publication of this document (<http://trustee.ietf.org/license-info>). Please review these documents carefully, as they describe your rights

and restrictions with respect to this document.

## Abstract

The purpose of a referral is to enable a given entity in a multiparty application to pass information to another party. This memo specifies a Generic Referral Object (GRO) to be used in the context of referrals. The proposed object is compact and is application-independent. Both IPv4 and IPv6 schemes are supported, as well as upper layer identifiers. Additional information to characterise an enclosed reference is also described. To allow proper interpretation of referrals, a new notion of scope identifiers is introduced.

## Table of Contents

<a href="#">1.</a>	<a href="#">Introduction and Motivation</a>	<a href="#">3</a>
<a href="#">1.1.</a>	<a href="#">Terminology</a>	<a href="#">6</a>
<a href="#">1.2.</a>	<a href="#">Normative Notation</a>	<a href="#">6</a>
<a href="#">2.</a>	<a href="#">Summary of Requirements</a>	<a href="#">6</a>
<a href="#">3.</a>	<a href="#">Referral Semantics and Scope Identifiers</a>	<a href="#">7</a>
<a href="#">4.</a>	<a href="#">Generic Referral Object Format</a>	<a href="#">11</a>
<a href="#">4.1.</a>	<a href="#">End of Qualifiers (EQQ)</a>	<a href="#">12</a>
<a href="#">4.2.</a>	<a href="#">IPv4 and IPv6 Addresses (references)</a>	<a href="#">12</a>
<a href="#">4.3.</a>	<a href="#">FQDN (reference)</a>	<a href="#">13</a>
<a href="#">4.4.</a>	<a href="#">HIT (reference)</a>	<a href="#">13</a>
<a href="#">4.5.</a>	<a href="#">HI (reference)</a>	<a href="#">13</a>
<a href="#">4.6.</a>	<a href="#">IPv4 and IPv6 Masks (qualifiers)</a>	<a href="#">13</a>
<a href="#">4.7.</a>	<a href="#">Ref_lifetime (qualifier)</a>	<a href="#">13</a>
<a href="#">4.8.</a>	<a href="#">Ref_source (qualifier)</a>	<a href="#">14</a>
<a href="#">4.9.</a>	<a href="#">Ref_scope (qualifier)</a>	<a href="#">14</a>
<a href="#">4.10.</a>	<a href="#">ScopeID (qualifier)</a>	<a href="#">15</a>
<a href="#">4.11.</a>	<a href="#">Port_number (qualifier)</a>	<a href="#">15</a>
<a href="#">4.12.</a>	<a href="#">Port_range_contig (qualifier)</a>	<a href="#">16</a>
<a href="#">4.13.</a>	<a href="#">Transport_protocol (qualifier)</a>	<a href="#">16</a>
<a href="#">4.14.</a>	<a href="#">Port_source (qualifier)</a>	<a href="#">16</a>
<a href="#">4.15.</a>	<a href="#">Extensibility</a>	<a href="#">16</a>
<a href="#">5.</a>	<a href="#">Security Considerations</a>	<a href="#">17</a>
<a href="#">6.</a>	<a href="#">IANA Considerations</a>	<a href="#">18</a>
<a href="#">7.</a>	<a href="#">Acknowledgements</a>	<a href="#">18</a>
<a href="#">8.</a>	<a href="#">Change log</a>	<a href="#">19</a>
<a href="#">9.</a>	<a href="#">References</a>	<a href="#">19</a>
<a href="#">9.1.</a>	<a href="#">Normative References</a>	<a href="#">19</a>
<a href="#">9.2.</a>	<a href="#">Informative References</a>	<a href="#">19</a>
<a href="#">Appendix A.</a>	<a href="#">Example Use Cases</a>	<a href="#">21</a>
<a href="#">Authors' Addresses</a>		<a href="#">22</a>



## **1. Introduction and Motivation**

A frequently occurring situation is that one entity A connected to the Internet (or to some private network using the Internet protocol suite) needs to inform another entity B how to reach either A itself or some third-party entity C. This is known as a referral.

In the original design of the Internet, IP addresses were global, unique, and quasi-permanent. Also any differentiation beyond that provided by an IP address was done by protocol and port numbers. Referrals were therefore performed simply by passing an IP address and possibly protocol and port numbers. In fact simple referrals (the first case above, sometimes called first-party referrals) were never needed since B could simply use A's address. Third-party referrals were trivial: A would tell B about C's address. Thus, it became common practice to pass raw addresses between entities. A classical example is the FTP PORT command [[RFC0959](#)].

Unfortunately, this simple approach to referrals often fails in today's Internet. As has been known for some time [[RFC2101](#)], addresses no longer all have global scope, often have limited reachability, and may have limited lifetime. It is no longer reasonable to assume that a host with a fixed location has a fixed address, or even a stable address.

We also encounter multi-interfaced hosts whose reachability is bound to a particular (logical/physical) interface. Furthermore, in the context of IPv4 address exhaustion, several solutions have emerged to share a single public IPv4 address between several customers simultaneously. Consequently, an IPv4 address often no longer identifies a single customer/user/host. Other information (e.g., port range) is required to identify unambiguously a given customer/user/host.

Both addresses and port numbers may be different on either side of a NAT or some other middlebox [[RFC3234](#)], and firewalls may block them. It is no longer reasonable to assume that an address for host H, which allows a given peer to reach that host in one location, also works from a different location - even if H is reachable from the second location. Also, the Internet now has two co-existing address formats for IPv4 and IPv6. Sending an out-of-scope or expired address, or one of the wrong format, as a referral will fail.

In some cases, this problem may be readily solved by passing a Fully Qualified Domain Name (FQDN) instead of an IP address. Indeed, that is an architecturally preferred solution [[RFC1958](#)]. However, it is not sufficient in many cases of dynamic referrals. Experience shows that an application cannot use a domain name in order to reliably



find useable address(es) of an arbitrary peer. Domain names work fairly well to find the addresses of public servers, as in web servers or SMTP servers, because operators of such servers take pains to make sure that their domain names work. But DNS records are not as reliably maintained for arbitrary hosts such as might need to be contacted in peer-to-peer applications, or for servers within corporate networks. Many small networks do not even maintain DNS entries for their hosts, and for some networks that do list local hosts in DNS, the listings may well be unusable from a remote location, because of two-faced DNS, or because the A record contains a private address. These cases may even be intentional as part of a security ring-fence, where `w3.example.com` only resolves within the corporate boundary, and/or resolves to IP addresses which are only reachable within the corporate administrative boundaries. In such contexts, incoming connections are usually filtered by the corporate firewall.

Furthermore, an FQDN may not be sufficient to establish successful communications involving heterogeneous peers (i.e., IPv4 and IPv6) since A and AAAA records may not be consistently provisioned. There are known cases where a server has one name that produces an A record (e.g., `www.example.com`) and another name that produces an AAAA record (e.g., `ipv6.example.com`).

In such cases, an IP address either cannot be derived from an FQDN, or if so derived, cannot be accessed from an arbitrary location in the Internet.

A related problem is that an application does not have a reliable way of knowing its own domain name - or to be more precise, a way of knowing a domain name that will allow the application to be reached from another location.

There are wider systemic problems with the DNS as a reliable way to find a useable address, which are somewhat out of scope here, but can be summarised:

- o In large networks, it is now quite common that the DNS administrator is out of touch with the applications user or administrator, and as a result, that the DNS is out of sync with reality.
- o DNS was never designed to accommodate mobile or roaming hosts, whose locator may change rapidly.
- o DNS has never been satisfactorily adapted to isolated, transiently-connected, or ad hoc networks.
- o It is no longer reasonable to assume that all addresses associated with a DNS name are bound to a single host. One result is that the DNS name might suffice for an initial connection, but a specific address is needed to rebind to the same peer, say, to



recover from a broken connection.

- o It is no longer reasonable to assume that a DNS query will return all usable addresses for a host.

For all the above reasons, the problem of address referrals cannot be solved simply by recommending the use of FQDNs instead. The guideline in [\[RFC1958\]](#) is in fact too simple for today's network. Something more elaborate than an IP address or an FQDN appears to be needed in the general case of application referrals.

The first motivation for this draft is the observation that unless the parties involved have reached an understanding about the scope, lifetime, and format of the elements in a referral through some other means, that information must be passed with the referral. This is required so that the receiving entity can determine whether or not the referral is useful. The referral therefore needs to consist of a fully-fledged data structure.

When a referral fails, good design suggests that the receiving entity should attempt to correct the situation. For example, if communication fails to be established using an IP address, it would often be appropriate to attempt a DNS lookup, despite the difficulties mentioned above. The second motivation for this draft is that it may be helpful to the entity receiving a referral to also receive information about the source of the referral, such as an FQDN, if that is known to the sender of the referral. The receiving entity can then attempt to recover a valid address (and possibly port number) for the referred entity.

The third motivation is to allow a referral to contain alternatives to an IP address or an FQDN, when any such alternatives exist.

We observe that we have outlined two separate requirements above: the need to define address scope more precisely, and the need for a generic referral object. Below, the two requirements are made more precise and a GRO format is defined.

It should be noted that partial or application-specific solutions to this problem abound. A non-normative Appendix gives examples, in the form of use cases. The objective of this specification is to define a generic and extensible solution, to allow more robust application design. It is an open question whether existing applications will benefit from retro-fitting GROs, or whether they will mainly be of use for new applications.





### **1.1. Terminology**

This document makes use of the following terms:

- o "Generic Referral Object (GRO)": the data object defined by this specification.
- o "Entity": we use this rather than "application" to describe any software component embedded in a host, not just a specific application, that sends, receives or makes use of GROs. Also, in case of dynamic load sharing or failover, an entity might even migrate between hosts.
- o "Referral": the act of one entity informing another entity how to contact a specific entity.
- o "Reference": the actual data (name, address, identifier, locator, pointer, etc.) that is the basis of a referral.
- o "Scope Identifier (ScopeID)": an identifier for the scope of reachability of a reference.
- o "Qualifier": a data item that gives additional information about a Reference.
- o "Referring entity": the entity that sends a referral.
- o "Receiving entity": the entity that receives a referral.
- o "Referenced entity": the entity described in a GRO.

### **1.2. Normative Notation**

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](#)].

## **2. Summary of Requirements**

A GRO should be self-describing; that is to say, even if it is forwarded several times across the network, the ultimate receiving entity should be able to extract and interpret all the information inserted by the original referring entity.

A GRO should be compact (i.e., binary encoded) and designed for efficient processing.

The GRO format must not be specific to a given IP version and must not be application-specific.

A GRO should contain information that the referring entity can provide about the scope, lifetime, format and source of the referral, encoded in a universal format.

The GRO format should be extensible with well-defined backwards compatibility.



A damaged GRO would be useless. However, to maintain efficiency, intrinsic error detection or correction for GROs should not be mandatory. Therefore, GROs SHOULD be sent over a channel supporting error detection or correction.

A forged GRO would be at least as dangerous as a forged IP address. However, to maintain efficiency, intrinsic cryptographic authentication of GROs should not be mandatory. The use of an authenticated channel to transmit GROs is RECOMMENDED.

An intercepted GRO would be at least as revelatory as an intercepted IP address. However, to maintain efficiency, intrinsic encryption of GROs should not be mandatory. The use of an encrypted channel to transmit GROs is RECOMMENDED.

### **3. Referral Semantics and Scope Identifiers**

The principal purpose of a referral is to enable one entity in a multi-party application to pass information to another party involved in the same application. This specification makes no assumptions about whether the entities are acting as clients, servers, peers, super-nodes, relays, proxies, etc., as far as the application is concerned. Neither does it take a position as to how the various entities become aware of the need to send a referral; this depends entirely on the structure of the application.

It is the responsibility of the referring entity to construct a GRO on the basis of information in its possession. It is the responsibility of the receiving entity to interpret and check this information. Due to the fluidity of connectivity in today's Internet, the referring entity cannot guarantee that the referenced entity can be reached. This can only be checked by the receiving entity. In the event of a reachability problem, information in the GRO may assist the receiving entity to find an alternative path.

Since the most fundamental quantity likely to be conveyed in a GRO is an IP address, (and possibly a port number) its scope is a key question. Address scope is not a simple concept, as shown by the discussion in [[RFC4007](#)] and the practical difficulties caused by [[RFC3484](#)]. Even the concept of link-local scope is complicated by the existence of multi-link subnets [[RFC4903](#)]. For the purpose of referrals, it seems that previous formalisations of the concept of scope are inadequate. Assuming that a GRO is trustworthy, one question that a receiving entity must answer is: "can the address in this GRO be reached from here?" That question is not answered by knowing only the scope (in the sense of [[RFC4007](#)]) as defined at the location of the referring entity. For that reason, scope is



represented in a new way in GROs. Firstly, the scope is qualified (to the best of the referring entity's knowledge) as follows:

- o Null. The address is known not to be applicable outside the referring host (e.g., a loopback address). This option is provided mainly for completeness. There is no value in such a GRO, and for privacy reasons it should not be communicated anyway.
- o Link. Apart from the standard Ethernet-like view of link locality, this scope would also apply to point-to-point links and to fragments of a multi-link subnet. Although on-link referrals should be trivial, this case is included to allow for uniform design of applications utilising GROs, so that link-local does not become a special case.
- o Limited. The address has applicability beyond the link, but it is known not to have global applicability. Examples include IPv4 private addresses [[RFC1918](#)] and IPv6 Unique Local Addresses (ULAs)[[RFC4193](#)]. Other cases include addresses on subnets which the referring entity knows to be obstructed by firewalls, network address translators, or other barriers to transparency [[RFC2775](#)]. A typical case is the set of subnets sharing a single set of border routers connecting them to the Internet.
- o Global. The address has applicability beyond the link, and is believed to have global applicability within its address family.

However, particularly in the case of limited scope, this is insufficient for the receiving entity to decide whether the address is applicable in the receiving entity's context. The scopes above are described as if they were a set of concentric circles, but reality is more complex, and limited scopes might overlap each other in an arbitrary way, for example when multiple VPNs are formed. A case in point is a VPN constructed between two independent sites, over which only those two sites' ULAs are routed. This would allow a complex pattern of overlapping scopes. For example, hosts in site A might potentially have addresses in three different scopes (global, Site\_A\_only, ULA\_A+B). Similarly, site B might also recognize three scopes (global, Site\_B\_only, ULA\_A+B). Which hosts can send packets to each other will depend on the combination of addresses and scopes available. For example, a host which only has an address in scope Site\_A\_only cannot send a packet to a host which only has an address in scope ULA\_A+B. Hosts in scope ULA\_A+B can send packets between sites A and B over the VPN. This can readily be encoded in routing configurations, but application software is generally unaware of it.

Thus, a referring entity may or may not be aware that the receiving entity and the referenced entity are within a link scope or limited scope that does not contain the referring entity. Therefore, a GRO may also include a scope identifier (ScopeID), which is an arbitrary label for a region of the network within which certain link or limited scope addresses are applicable.



We cannot assume that the referring entity knows the scopes that are accessible to the receiving entity. For this reason, all available information SHOULD be included in a GRO.

[[ Discussion invited: Should we also define optional preference information for each reference in a GRO, or alternatively require that references be listed in order of preference? Here is some tentative text, not followed up in the strawman GRO format below. ]]

A preference order (or reachability trust level?) MAY be associated with enclosed objects but the receiving entity is not obliged to follow that order since this may induce conflicts with local policies (sometime on per interface basis).

There needs to be a high level of assurance that ScopeIDs are unique, or at least that a GRO will never be forwarded outside a region in which ScopeIDs are unique. Also, all referring and receiving entities need to be aware of the ScopeID(s) that apply to them. However, it is clearly undesirable to create a new global registration scheme for ScopeIDs.

The delimiter of a limited scope will in many cases be the device (firewall or NAT) that obstructs transparency. A tempting solution would be to use some unique identifier of that device as the unique ScopeID. Unfortunately, this cannot be an IP address of the device, since in the case of nested NATs, all its addresses may be ambiguous. Neither can we rely on such a device having its own FQDN, or on that FQDN being known to all entities within the scope concerned. Finally, some limited scopes may not be hidden behind a single such device; for example, a limited scope might consist of a company's network and selected VPN connections to subsets of several business partners' networks. Alternatively, multiple limited scopes might be hidden behind the same device. Device addresses are therefore not suitable as ScopeIDs.

Therefore, a limited scope can best be defined as whatever set of referring and receiving entities have been configured (statically or dynamically) to accept a given ScopeID in some unambiguous namespace (see [Section 4.10](#)).

Methods for configuring, advertising and discovering ScopeIDs are not defined in this document. However, in their absence, it is extremely hard for receiving entities to interpret and use information about limited scopes. To the extent possible, all entities involved in referrals should determine what scope is shared between the referred entity and the receiving entity, by any means. Those means are not covered in this document, but may include use of external services, agreement on scope identifiers, or direct negotiation.





- o If shared scope (or set of scopes) is determined, a referral should ideally only include information useful in that scope or set of scopes.
- o If shared scope is uncertain, a referral should include all information that might be useful, taking privacy considerations into account.

In general, the referring entity cannot know the scope in which the GRO will be interpreted. For example, the initial receiving entity may itself be behind a NAT, unknown to the referring entity, or the receiving entity may send the GRO onwards to another host in yet another scope. In practice, we have to leave the receiver to decide whether certain information is useful or not. In the case of a ScopeID in particular, the referring entity is not required to know which ScopeIDs apply to the receiving entity.

Discovery or negotiation of ScopeIDs between referring, referenced and receiving entities is certainly a possibility, but may be expensive, and is not assumed by this specification.

A referring entity may obtain the address and port number for the referenced entity in various ways, and knowledge about this may help the receiving entity when combined with scope information. For example, if the receiving entity is aware that the address has been translated, and that it has global scope, it may choose to use it without further checks. If it is not marked as translated, and has limited scope, the receiving entity may then verify whether it has a suitable ScopeID.

To enable such logic, a GRO may describe the source of an address or port number. How knowledge of this source is obtained is outside the scope of the present specification, but ICE [[I-D.ietf-mmusic-ice](#)] is an example method. It is also out of scope here to describe exactly how the receiving entity uses the information; for example GRO semantics do not include or imply preferences or priorities when multiple addresses are provided. The receiving entity may choose to use a predefined policy, apply general logic as sketched in the previous paragraph, or follow application-specific logic, all based on the data provided in a GRO.

Obviously, a GRO is no use unless it contains at least one item that can be used to find a path to the referred entry. One option would be to make the presence of at least one IP address mandatory. However, there are alternatives, the most obvious one being an FQDN. Any form of identifier-locator separation, with HIP [[RFC5201](#)] as an example, may also offer an alternative. Therefore, we do not require a GRO to include an IP address, even though its inclusion is a very likely case.



#### 4. Generic Referral Object Format

[[ Note: This section is a strawman approach to make the ideas more concrete. It is entirely open for discussion. ]]

A GRO is composed of a sequence of binary-encoded type-length-value fields (TLVs) transmitted in network byte order. The TLV format is as follows:

```

      0               1               2               3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+..
|  GRTYPE    |Q|  GRLength    |          GRValue          ...
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+..

```

A GRO MUST include at least one reference that allows a receiving entity to attempt to establish a path to the referred entity. A typical case is an IPv4\_address or IPv6\_address TLV. Multiple references may be present, and their order is not significant.

Apart from this, all TLVs are OPTIONAL.

[[ Discusssion invited: At the moment, there is no total length field or end flag for the whole GRO, assuming that GROs will be sent in some kind of container. Opinions among the authors vary about whether this is OK. ]]

GRTYPE: Specifies the type of the current TLV. GRTYPE is encoded in 7 bits. The initially specified types are, in decimal:

- 0: EOQ.
- 1: IPv4\_address.
- 2: IPv6\_address.
- 3: FQDN.
- 4: HIT.
- 5: HI.
- 65: IPv4\_mask.
- 66: IPv6\_mask.
- 67: Ref\_lifetime.
- 68: Ref\_source.
- 69: Ref\_scope.
- 70: ScopeID.
- 71: Port\_number.
- 72: Port\_range\_contig.
- 73: Transport\_protocol.



74: Port\_source.  
127: reserved.

A receiving entity MUST silently ignore any TLV with an unknown or reserved GRTYPE.

Each TLV is classified semantically as a reference or as a qualifier. A qualifier provides extra information about a reference or another qualifier..

[[ Discusssion invited: Do we need a syntactic method of distinguishing references from qualifiers? Since unknown TLVs are always discarded, why would that be needed? ]]

Q bit: If this bit is set to 1, the current TLV is followed by one or more TLVs that qualify it.

GRLength: The length in bytes of the GRValue field. Thus, the total length of the TLV is GRLength+2 bytes.

GRValue: The content and encoding depend on GRTYPE. Any padding required to fill an integral number of bytes MUST consist of a sequence of zero bits at the end of the content.

#### **4.1. End of Qualifiers (EOQ)**

This TLV follows the last TLV that qualifies a TLV whose Q bit is set to 1. Its GRLength must be set to 0.

The Q bit and EOQ MAY be used recursively, so that qualifiers may themselves be qualified if that proves to be useful.

Example (GT=GRTYPE):

```

GT Q|GT Q|GT Q|GT Q|GT Q|GT Q|GT Q|GT Q
A 1|xx 0|xx 0|B 1|xx 0|xx 0|EOQ 0|EOQ 0
      <-----> Qualifiers of B
      <-----> Qualifiers of A

```

The Q bit MUST NOT be set in an EOQ TLV.

#### **4.2. IPv4 and IPv6 Addresses (references)**

IPv4 and IPv6 addresses are encoded in their normal binary form, with GRLength being 4 and 16 respectively.

When multiple addresses are provided, their order does not imply an order of preference. The receiving entity SHOULD apply a local



policy and mechanism to choose between alternative addresses, using other information included in the GRO appropriately. This document does not describe such policies and mechanisms, which could be application specific.

#### **[4.3.](#) FQDN (reference)**

The Fully Qualified Domain Name of the referenced entity in ASCII format according to [[RFC1035](#)].

The GRLength is variable (maximum 63).

[[ Discussion invited: Is there also value in a generic URI item? See section on Extensibility below for a related discussion point. ]]

#### **[4.4.](#) HIT (reference)**

The Host Identity Tag of the referenced entity [[RFC5201](#)].

The GRLength must be set to 16.

#### **[4.5.](#) HI (reference)**

The Host Identifier of the referenced entity [[RFC5201](#)].

The GRLength is variable.

[[ Discussion invited: Is this necessary in order to run the HIP base exchange? The HI is a large object to include in a GRO. Also, do we need a more precise definition of what the HI is (see [section 5.2.8 of RFC5201](#))? ]]

#### **[4.6.](#) IPv4 and IPv6 Masks (qualifiers)**

IPv4 and IPv6 masks are encoded in their normal binary form, with GRLength being 4 and 16 respectively.

#### **[4.7.](#) Ref\_lifetime (qualifier)**

Remaining lifetime in seconds of the reference that it qualifies, encoded as a 32 bit binary number in the format of an IPv6 Valid Lifetime [[RFC4861](#)]. GRLength must be set to 4.

If the lifetime is absent, or if it indicates an infinite lifetime [[RFC4861](#)], the receiving entity MUST assign a lifetime of one day to the corresponding reference.

The receiving entity MUST count down a received lifetime





appropriately. If the GRO is forwarded to an additional receiving entity, the lifetime MUST be updated appropriately.

[[ Discussion invited: would it be better to specify an expiry timestamp? ]]

[[ Discussion invited: is the default of one day reasonable? ]]

#### **4.8. Ref\_source (qualifier)**

This is a single byte indicating the source of the reference that it qualifies. GRLength must be set to 1. The following values may be used:

- 0: source was static configuration
- 1: source was DNS lookup
- 2: source was DHCP or DHCPv6
- 3: source was SLAAC
- 4: relayed address (e.g. from TURN [[I-D.ietf-behave-turn](#)] or SOCKS)
- 5: translated address. ("server reflexive" in ICE [[I-D.ietf-mmusic-ice](#)] terminology.)
- 6: source was DNS64 synthesis.

A receiving entity MUST silently ignore unknown values.

#### **4.9. Ref\_scope (qualifier)**

This is a single byte indicating the scope of the reference that it qualifies. GRLength must be set to 1. The scopes are explained in [Section 3](#). The currently defined values are as follows:

- 0: Null
- 1: Link
- 2: Limited
- 7: Global. Note that some unused values precede this value, in case of future changes.

A receiving entity MUST silently ignore unknown values.

References qualified with the Null value SHOULD NOT be sent and MUST be silently ignored by a receiving entity.

When a receiving entity receives a reference qualified with a Link or Limited Ref\_scope, and without a ScopeID, it should take locally defined steps to check whether the reference is in fact within a reachable scope.



#### **4.10. ScopeID (qualifier)**

A ScopeID, if present, is a label for the scope of the reference that it qualifies.

When a receiving entity receives a reference qualified with a Ref\_scope and a ScopeID, it should verify the ScopeID against a list of ScopeIDs known to be reachable and if not, take other locally defined steps to check whether the reference is in fact within a reachable scope.

ScopeIDs should be reasonably certain to be unique, yet require no new system for central administration.

[ [ Discussion invited: It isn't clear to the authors that a single syntax for ScopeID is sufficient. Should we allow for subtypes, so that (e.g.) ULA format and FQDN format would both be possible? Should we consider a URI format? The following proposal is tentative. ] ]

The proposed method is that each organisation that needs to define a ScopeID will first generate a ULA prefix as defined in [[RFC4193](#)], and then form a specific IPv6 address using that ULA prefix. It is RECOMMENDED to form an address using a valid universal EUI-64 interface identifier according to [[RFC4291](#)], and this EUI-64 identifier MAY be the same one as used in the [RFC4193](#) procedure.

The GRLength must be set to 16. The GRValue is the ScopeID in the format of an IPv6 address, although it will be treated entirely as an opaque binary value in the GRO referring and receiving entities.

#### **4.11. Port\_number (qualifier)**

The inbound TCP/UDP/SCTP/DCCP port number associated with the reference that it qualifies. The port number may be bound to a specific transport protocol.

The GRLength must be set to 2.

The GRValue is a 16-bit port number.

This TLV MAY be qualified by Transport\_protocol or Port\_source TLVs. An IP address may be qualified by zero, one or several Port\_number TLVs.



#### **4.12. Port\_range\_contig (qualifier)**

A contiguous range of TCP/UDP/SCTP/DCCP port numbers associated with the reference that it qualifies. The port range may be bound to a specific transport protocol (see Transport\_protocol item).

The GRLength is 4.

The GRValue is two 16-bit port numbers, defining the lower and upper bounds of the port range.

This TLV MAY be qualified by Transport\_protocol or Port\_source TLVs. An IP address may be qualified by zero, one or several Port\_range\_contig TLVs.

Defining this TLV is motivated by the need to define a compact object instead of listing all port numbers that are part of a port range [[I-D.boucadair-port-range](#)], [[I-D.ymbk-aplusp](#)]. Use cases for this qualifier still need to be defined.

#### **4.13. Transport\_protocol (qualifier)**

This is a single byte indicating the IPv4 protocol number or IPv6 Next Header value used with the reference or Port\_number or Port\_number\_contig that it qualifies. GRLength must be set to 1.

A receiving entity MUST silently ignore unknown values.

#### **4.14. Port\_source (qualifier)**

This is a single byte indicating the source of the Port\_number that it qualifies. GRLength must be set to 1. Accepted values are:

- 0: direct (i.e. known to be the original port number used by the referenced entity)
- 4: relayed port (e.g. from TURN [[I-D.ietf-behave-turn](#)] or SOCKS)
- 5: translated port. ("server reflexive" in ICE [[I-D.ietf-mmusic-ice](#)] terminology.)

[[ Discussion invited: Is this distinction useful? ]]

The assigned values were chosen to align with those for Ref\_source. A receiving entity MUST silently ignore unknown values.

#### **4.15. Extensibility**

Additional GRTypes may be assigned in the range up to 126 by IANA action as defined in [Section 6](#). The documentation of a new GRType must specify its name, define its GRLength, and describe the contents



and meaning of its GRValue, including whether it is a reference or a qualifier.

This extensibility is not intended to allow a GRO to grow enough to contain every possible kind of application-layer identifier that could ever be used in a referral, because then it would be too hard to write a generic "please connect me to the peer at this GRO" function. Thus, additional GRTypes SHOULD NOT be assigned except for generic purposes that will apply to multiple applications. Similarly, additional sub-types for Address\_source, Address\_scope, Transport\_protocol, and Port\_source SHOULD NOT be assigned except for generic purposes.

[[ Discussion invited: Sheng Jiang suggested that there should be a generic 'Application-specific ID' GRType, for example in URI format. A problem with this is that it might end up as a catch-all like a DNS TXT record, and threaten interoperability as a result. ]]

The reserved GRType value 127 is intended to be used to define an extended range of GRTypes in the highly unlikely event that this becomes necessary.

## 5. Security Considerations

It should be noted that GROs cannot function as a way to nullify the effect of a firewall or any other security mechanism. If the receiving entity chooses a particular reference in the GRO and attempts to send packets to the corresponding IP address, whether they are delivered or not will depend on the existing security mechanisms, whatever they may be.

Nevertheless, if a site security policy requires it, certain references MAY be excluded from GROs sent to certain destinations. This would require a security policy mechanism to be added to the process of generating GROs and is not further specified here.

Forged or intercepted GROs would enable a wide variety of attacks. Although not fundamentally different from attacks based on forged or observed IP addresses or FQDNs, no doubt GROs would allow such attacks to be more ingenious, simply because they provide more information than an address or FQDN alone. As noted in [Section 2](#), GROs SHOULD be transmitted through authenticated and encrypted channels. Since this is a requirement of the channel and not of the GRO, and the channel used depends on a specific use case, it is not further elaborated here.

Conceivably, an extension to the GRO format could be defined which





would allow it to carry security information (for example, some sort of handle or nonce to authorise firewall penetration). Any such extension MUST be adequately encrypted, in such a way that only a receiving entity in possession of a given key can decrypt it. This is necessary even if the GRO is transmitted through a secure channel.

GROs are variable length objects with no defined maximum length. It is possible that a malicious GRO could be constructed, with harmful code masquerading as legitimate or unknown GRType items. All implementations of receiving entities MUST guard against buffer overflows, as well as obeying the rules about ignoring unknown values in [Section 4](#).

Unknown TLVs in GROs are to be ignored by the receiving entity. However, GROs may be forwarded to additional receiving entities, in which case the unknown TLVs will be forwarded too. A receiving entity MAY remove unknown TLVs before forwarding a GRO, as a precaution against malicious use.

GROs raise potential privacy issues, which are not explored in this document. For example, in the SIP context, mechanisms such as [\[RFC3323\]](#) and [\[I-D.ietf-sip-ua-privacy\]](#) are available to hide information that might identify end-points. Usage scenarios for GROs MUST ensure that they do not unintentionally defeat privacy solutions.

## **6. IANA Considerations**

IANA is requested to established a Generic Referral Object (GRO) registry, containing sub-registries for GRType, Ref\_source, Ref\_scope, and Port\_source. The range and initial assignments are defined in [Section 4](#).

New values in this registry are to be assigned according to the Specification Required policy defined in [\[RFC5226\]](#), which implies review by a Designated Expert according to [Section 4.15](#).

[[ Discussion invited: It has been suggested to define a (small) registry for Global ScopeIDs, instead of assuming that "global" for IPv4 and IPv6 is unambiguous. ]]

## **7. Acknowledgements**

This document originated from a Thai Lunch BOF (a variant of a Bar BOF) at IETF74. Scott Brim contributed substantially to the first version. Valuable comments and contributions were made by Dan Wing,



Andrew Sullivan, ...

This document was produced using the xml2rfc tool [[RFC2629](#)].

## 8. Change log

[draft-carpenter-referral-object-00](#): original version, 2009-05-11

[draft-carpenter-referral-object-01](#): updated after discussion in BEHAVE WG at IETF75 and on GROBJ BOF mailing list, 2009-10-20

## 9. References

### 9.1. Normative References

- [RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, [RFC 1035](#), November 1987.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", [RFC 4193](#), October 2005.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", [RFC 4291](#), February 2006.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", [RFC 4861](#), September 2007.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", [BCP 26](#), [RFC 5226](#), May 2008.

### 9.2. Informative References

- [I-D.boucadair-port-range]  
Boucadair, M., Levis, P., Bajko, G., and T. Savolainen, "IPv4 Connectivity Access in the Context of IPv4 Address Exhaustion: Port Range based IP Architecture", [draft-boucadair-port-range-02](#) (work in progress), July 2009.
- [I-D.boucadair-sipping-ipv6-atypes]  
Boucadair, M., Noisette, Y., and A. Allen, "The atypes



media feature tag for Session Initiation Protocol (SIP)",  
[draft-boucadair-sipping-ipv6-atypes-02](#) (work in progress),  
July 2009.

[I-D.ietf-behave-turn]

Rosenberg, J., Mahy, R., and P. Matthews, "Traversal Using  
Relays around NAT (TURN): Relay Extensions to Session  
Traversal Utilities for NAT (STUN)",  
[draft-ietf-behave-turn-16](#) (work in progress), July 2009.

[I-D.ietf-mmusic-ice]

Rosenberg, J., "Interactive Connectivity Establishment  
(ICE): A Protocol for Network Address Translator (NAT)  
Traversal for Offer/Answer Protocols",  
[draft-ietf-mmusic-ice-19](#) (work in progress), October 2007.

[I-D.ietf-sip-ua-privacy]

Munakata, M., Schubert, S., and T. Ohba, "UA-Driven  
Privacy Mechanism for SIP", [draft-ietf-sip-ua-privacy-08](#)  
(work in progress), May 2009.

[I-D.ymbk-aplusp]

Bush, R., "The A+P Approach to the IPv4 Address Shortage",  
[draft-ymbk-aplusp-04](#) (work in progress), July 2009.

[RFC0959] Postel, J. and J. Reynolds, "File Transfer Protocol",  
STD 9, [RFC 959](#), October 1985.

[RFC1918] Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G., and  
E. Lear, "Address Allocation for Private Internets",  
[BCP 5](#), [RFC 1918](#), February 1996.

[RFC1958] Carpenter, B., "Architectural Principles of the Internet",  
[RFC 1958](#), June 1996.

[RFC2101] Carpenter, B., Crowcroft, J., and Y. Rekhter, "IPv4  
Address Behaviour Today", [RFC 2101](#), February 1997.

[RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", [RFC 2629](#),  
June 1999.

[RFC2775] Carpenter, B., "Internet Transparency", [RFC 2775](#),  
February 2000.

[RFC3234] Carpenter, B. and S. Brim, "Middleboxes: Taxonomy and  
Issues", [RFC 3234](#), February 2002.

[RFC3323] Peterson, J., "A Privacy Mechanism for the Session



Initiation Protocol (SIP)", [RFC 3323](#), November 2002.

- [RFC3484] Draves, R., "Default Address Selection for Internet Protocol version 6 (IPv6)", [RFC 3484](#), February 2003.
- [RFC4007] Deering, S., Haberman, B., Jinmei, T., Nordmark, E., and B. Zill, "IPv6 Scoped Address Architecture", [RFC 4007](#), March 2005.
- [RFC4091] Camarillo, G. and J. Rosenberg, "The Alternative Network Address Types (ANAT) Semantics for the Session Description Protocol (SDP) Grouping Framework", [RFC 4091](#), June 2005.
- [RFC4092] Camarillo, G. and J. Rosenberg, "Usage of the Session Description Protocol (SDP) Alternative Network Address Types (ANAT) Semantics in the Session Initiation Protocol (SIP)", [RFC 4092](#), June 2005.
- [RFC4903] Thaler, D., "Multi-Link Subnet Issues", [RFC 4903](#), June 2007.
- [RFC5201] Moskowitz, R., Nikander, P., Jokela, P., and T. Henderson, "Host Identity Protocol", [RFC 5201](#), April 2008.

## **Appendix A. Example Use Cases**

[[ This appendix is incomplete and preliminary. ]]

Referrals may be used to add an entity to a multi-party conversation, or they may be used (in applications such as telephony) as the first step of transferring one end of a conversation from the referring entity to the receiving entity. [[ Say more? ]]

TBD: FTP/PORT, HTTP referrals... text needed

BitTorrent is a distributed file sharing infrastructure. It is based on P2P techniques for exchanging files between connected users. Three parties are involved in a BitTorrent architecture: (1) The server into which, has been uploaded the torrent file. (2) The Tracker which maintains a list of clients which have the file or some portions of that file. (3) Entities which are downloading and/or uploading portions of the file. In order to download a given file, a BitTorrent client needs to obtain the corresponding torrent file (i.e. a file which includes the meta-data information of the file to be shared: the file name, its length, a hash and the URL of the tracker.). Then, it connects to the tracker to retrieve a list of leechers (clients which are currently downloading the file but do not





yet detain all the portions of the file) and seeders (clients which detain all the portions of the file and are uploading them to other requesting clients). The client connects to those machines and downloads the available portions of the requested file.

In a GRO for Skype purposes, if the address fails, you'd have to fall back to the Skype ID instead of an FQDN. This is a case where allowing an application-specific ID might be valuable. Another case would be Lotus Domino databases - if both IP address and FQDN fail to find the relevant server, the server name and the database name could be used as fallback identifiers.

In SIP environments, a SIP Proxy Server intervenes in the placement of SIP sessions between two UAs. Particularly, the SDP part of relayed SIP messages includes required information for establishment of RTP sessions (particularly IP address and port number). A media description may be unidirectional or symmetric. ICE and ANAT allow listing several network types and addresses in the same SDP offer.

ANAT: ANAT [[RFC4091](#)], [[RFC4092](#)] is a procedure used by Dual Stack UAs to provide both IPv4 and IPv6 addresses in the context of a single logical media stream. This helps interworking as, whatever the distant UA version is (IPv4/IPv6-only or Dual-Stack) provided that this latter is able to understand at least one of the offers. ANAT semantic does not allow to characterize the IP address(es) it carries. For instance, no indication if the UA is behind a translator or not is supported by ANAT (or even ICE). ICE deprecates ANAT attribute.

Atypes [[I-D.boucadair-sipping-ipv6-atypes](#)]: atypes is a SIP media feature tag which indicates the IP address type capabilities of the UA (User Agent) and can aid the routing process and ease the invocation of required functions (e.g. SIP-ALG, NAT64, NAT46) when heterogeneous (i.e. IPv4 and IPv6) parties are involved in a given SIP session. Atypes can be used jointly with GRO (also with ICE and ANAT) to optimise the media path as experienced between involved parties (especially when Dual-stacks UAs are involved).



Authors' Addresses

Brian Carpenter (editor)  
Department of Computer Science  
University of Auckland  
PB 92019  
Auckland, 1142  
New Zealand

Email: [brian.e.carpenter@gmail.com](mailto:brian.e.carpenter@gmail.com)

Mohamed Boucadair  
France Telecom  
3, Avenue Francois Chateaux  
Rennes 35000  
France

Email: [mohamed.boucadair@orange-ftgroup.com](mailto:mohamed.boucadair@orange-ftgroup.com)

Joel M. Halpern  
Ericsson  
P. O. Box 6049  
Leesburg, VA 20178  
US

Email: [jhalpern@redback.com](mailto:jhalpern@redback.com)

Sheng Jiang  
Huawei Technologies Co., Ltd  
KuiKe Building, No.9 Xinxu Rd.,  
Shang-Di Information Industry Base, Hai-Dian District, Beijing  
P.R. China

Email: [shengjiang@huawei.com](mailto:shengjiang@huawei.com)

Keith Moore  
Network Heretics

Email: [moore@network-heretics.com](mailto:moore@network-heretics.com)

