

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: May 5, 2016

B. Williams  
Akamai  
T. Reddy  
Cisco  
November 2, 2015

Peer-specific Redirection for Traversal Using Relays around NAT (TURN)  
draft-williams-peer-redirect-04

## Abstract

This specification describes a peer-specific redirection method that allows the TURN server to redirect a client for the purpose of improving communication with a specific peer without negatively affecting communication with other peers.

## Status of This Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on May 5, 2016.

## Copyright Notice

Copyright (c) 2015 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 (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

<a href="#">1.</a>	<a href="#">Introduction</a>	<a href="#">2</a>
<a href="#">1.1.</a>	<a href="#">Redirection for Performance</a>	<a href="#">3</a>
<a href="#">1.2.</a>	<a href="#">Redirection for Load Balancing</a>	<a href="#">4</a>
<a href="#">2.</a>	<a href="#">Terminology</a>	<a href="#">4</a>
<a href="#">3.</a>	<a href="#">Peer-specific Server Redirect Mechanism</a>	<a href="#">4</a>
<a href="#">3.1.</a>	<a href="#">Forming an Allocate Request</a>	<a href="#">5</a>
<a href="#">3.2.</a>	<a href="#">Receiving an Allocate Request</a>	<a href="#">5</a>
<a href="#">3.3.</a>	<a href="#">Forming a CreatePermission or ChannelBind Request</a>	<a href="#">5</a>
<a href="#">3.4.</a>	<a href="#">Receiving a CreatePermission or ChannelBind Request</a>	<a href="#">6</a>
<a href="#">3.5.</a>	<a href="#">Forming a Redirect Indication</a>	<a href="#">7</a>
<a href="#">3.6.</a>	<a href="#">Receiving a Redirect Indication</a>	<a href="#">8</a>
<a href="#">4.</a>	<a href="#">ICE Interactions</a>	<a href="#">8</a>
<a href="#">5.</a>	<a href="#">Security Considerations</a>	<a href="#">9</a>
<a href="#">5.1.</a>	<a href="#">Permission Flood</a>	<a href="#">9</a>
<a href="#">5.2.</a>	<a href="#">Unsolicited or Invalid Redirect Indication</a>	<a href="#">10</a>
<a href="#">5.3.</a>	<a href="#">Replayed Redirect Indication</a>	<a href="#">10</a>
<a href="#">6.</a>	<a href="#">IANA Considerations</a>	<a href="#">11</a>
<a href="#">7.</a>	<a href="#">Acknowledgements</a>	<a href="#">12</a>
<a href="#">8.</a>	<a href="#">References</a>	<a href="#">12</a>
<a href="#">8.1.</a>	<a href="#">Normative References</a>	<a href="#">12</a>
<a href="#">8.2.</a>	<a href="#">Informative References</a>	<a href="#">12</a>
<a href="#">Appendix A.</a>	<a href="#">Change History</a>	<a href="#">12</a>
<a href="#">A.1.</a>	<a href="#">Changes from version 03 to 04</a>	<a href="#">13</a>
<a href="#">A.2.</a>	<a href="#">Changes from version 02 to 03</a>	<a href="#">13</a>
<a href="#">A.3.</a>	<a href="#">Changes from version 01 to 02</a>	<a href="#">13</a>
<a href="#">A.4.</a>	<a href="#">Changes from version 00 to 01</a>	<a href="#">13</a>
	<a href="#">Authors' Addresses</a>	<a href="#">13</a>

[1.](#) Introduction

A Traversal Using Relay around NAT (TURN) [[RFC5766](#)] service provider may provide multiple candidate TURN servers for use by a host, but it might not be possible to determine which candidate TURN server will provide the best performance until both peers have been identified. This could be true for a variety of reasons, including:

- o Using the selected relay for a specific peer results in a sub-optimal end-to-end Internet path.
- o Load conditions on the selected relay have changed since the allocation was established such that it cannot support the new

data flow.

At the same time, the above conditions might apply to one peer but not another, such that it would be best to selectively use the existing relay allocation for peers that will receive reasonable

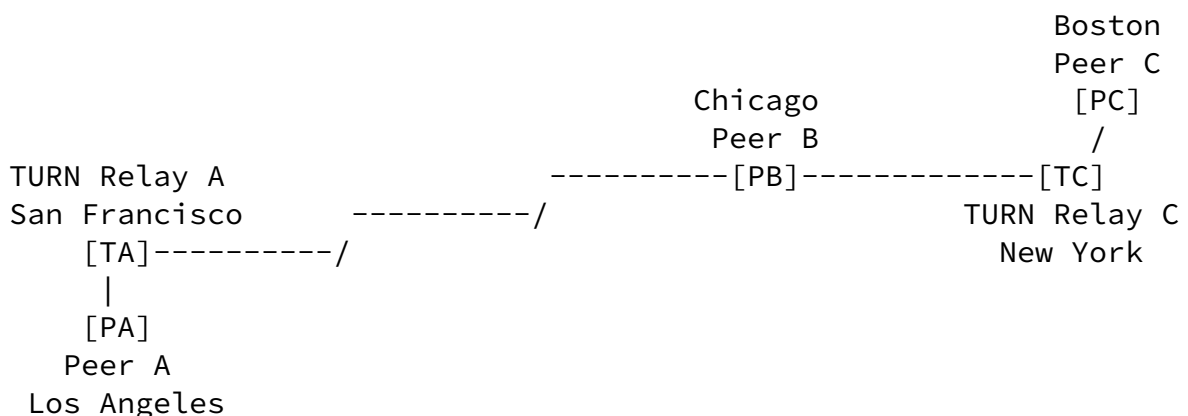
performance and redirect data flows for other peers to an alternate server. These scenarios are discussed in greater detail below.

The Session Traversal Utilities for NAT (STUN) protocol [[I-D.ietf-tram-stunbis](#)] defines an ALTERNATE-SERVER mechanism with which a server can redirect a client to another server by replying to a request message with an error response with error code 300 (Try Alternate). The TURN protocol describes error code 300 as one of the possible error codes for an Allocate error response.

This specification describes an additional use of the ALTERNATE-SERVER STUN attribute for TURN that allows the TURN server to redirect a client for the purpose of improving communication with a specific peer without negatively affecting communication with other peers.

### [1.1.](#) Redirection for Performance

Consider the following example:



When Peer B wishes to communicate with either Peer A or Peer C, it performs a DNS lookup and discovers TURN Relay C, the nearest of the candidate TURN servers. Peer B then sends a TURN Allocate request to TURN Relay C to determine the reflexive and relay candidates to

offer. After the reflexive candidate has been chosen, Peer B sends a ChannelBind request to TURN Relay C to establish a channel for communication with the peer. If Peer C is the remote peer, the existing allocation will perform reasonably well, but if Peer A is the remote peer, the latency for relayed packets will be nearly twice as long as if TURN Relay A had been selected as the relay candidate. The problem is worse if Peer B wishes to communicate with both Peer A and Peer C, since there is no single relay candidate that would provide optimum performance for both peers.

If TURN Relay C and TURN Relay A are part of a common TURN service, it would be possible for TURN Relay C to determine that TURN Relay A

will provide optimal service for communication between Peer B and Peer A. This allows the TURN service to redirect just the data channel between Peer A and Peer B to TURN relay A, thus providing optimal performance for both relay channels.

The above example describes the problem in terms of physical geography instead of network geography in order to help clarify the discussion. However, readers should note that the problem of selecting a relay server to achieve optimal end-to-end routing is much more complicated than the above description suggests, requiring a detailed real-time view of network connectivity characteristics and the peering relationships between autonomous systems. A naive approach based solely on the physical location of the hosts involved is just as likely to produce negative results as positive ones.

That said, a relay service provider with a broadly distributed system for actively monitoring network performance across the relevant parts of the Internet could make use of the resulting data set to select the optimal relay for each peer pair.

## [1.2.](#) Redirection for Load Balancing

At the point when a relay allocation is first established, it can be difficult to determine how much aggregate concurrent load could eventually be associated with that allocation. The initiating peer could attempt to use that allocation for any number of peer-to-peer data flows over an extended period of time, during which time load conditions on the relay could change substantially, such that quality of service for already established flows would degrade if the relay

were to accept additional flows.

Under these conditions, a TURN service provider with multiple relay hosts and distributed capacity could improve service quality by redirecting data flows to a different host that has more available capacity.

## [2.](#) Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [[RFC2119](#)].

## [3.](#) Peer-specific Server Redirect Mechanism

This specification describes a new STUN indication type, Redirect, which is used by a TURN server to notify a TURN client when better service could be available through an alternate TURN server. The

Redirect indication contains an ALTERNATE-SERVER attribute to provide the address for the alternate TURN server.

This specification also defines two new comprehension-optional STUN attributes: CHECK-ALTERNATE and XOR-OTHER-ADDRESS. The CHECK-ALTERNATE attribute is used by the client to request that the server perform peer-specific redirection. The XOR-OTHER-ADDRESS is used by the client to provide an alternate peer address for location identification in the event that the XOR-PEER-ADDRESS attribute in the CreatePermission or ChannelBind request is not expected to reliably serve this purpose.

### [3.1.](#) Forming an Allocate Request

When forming an Allocate request, a TURN client includes a CHECK-ALTERNATE STUN attribute to signal to the TURN server that peer-specific redirection is both supported and desired.

When forming a CHECK-ALTERNATE attribute, the STUN Type is TBD-CA. To maintain backward compatibility, this type is in the comprehension-optional range, which means that an [[RFC5766](#)] compliant TURN server can safely ignore it.

The CHECK-ALTERNATE attribute has no value part and thus the attribute length field is 0.

### [3.2.](#) Receiving an Allocate Request

When a server receives an Allocate request, it first processes the request as per the TURN specification [[RFC5766](#)]. After determining that a success response will be prepared, a TURN server that supports peer-specific redirection checks for a CHECK-ALTERNATE attribute. If one exists, the server stores this information as part of the allocation state. There is no need for the server to indicate that the attribute was accepted in the success response.

### [3.3.](#) Forming a CreatePermission or ChannelBind Request

When sending a CreatePermission or a ChannelBind request, the XOR-OTHER-ADDRESS STUN attribute allows the TURN client to provide an alternate peer address that can be used by the server to identify the network geographic location of the peer when performing the peer-specific redirection check. Use of this attribute is only necessary if the XOR-PEER-ADDRESS already contained in the CreatePermission or ChannelBind request does not adequately serve this purpose, which should only be true when both peers require a TURN relay for end-to-end data flow. In this case, the TURN CreatePermission or ChannelBind request will provide the peer's TURN relay address as the

XOR-PEER-ADDRESS value. If the RTT between the peer and its TURN relay server is very small, the TURN relay address might still be an appropriate address to use for the peer-specific redirection check. As the RTT grows, the TURN relay address will become less suitable for this purpose. For this reason, it is generally the case that the peer's public address (i.e. its host or reflexive address) is a better indication of its network geographic location than its TURN relay address.

When forming an XOR-OTHER-ADDRESS attribute, the STUN Type is TBD-XOA. To support backward compatibility, this type is in the comprehension-optional range, which means that an [[RFC5766](#)] compliant TURN server can safely ignore it.

The XOR-OTHER-ADDRESS value specifies an address and port suitable

for identification of the peer's network geographic location. It is encoded in the same way as XOR-MAPPED-ADDRESS [[I-D.ietf-tram-stunbis](#)].

A CreatePermission request is allowed to contain multiple XOR-PEER-ADDRESS attributes. When multiple peer addresses are provided in a CreatePermission request, it would be difficult for the TURN server to associate an XOR-OTHER-ADDRESS attribute with the correct XOR-PEER-ADDRESS. For this reason, a TURN client MUST form a separate CreatePermission request for an XOR-PEER-ADDRESS request when an XOR-OTHER-ADDRESS attribute will be included in the request.

The XOR-OTHER-ADDRESS attribute SHOULD NOT be included in a request if its value will be identical to the request's XOR-PEER-ADDRESS attribute. Its value would be redundant and a waste of space in the message.

#### [3.4.](#) Receiving a CreatePermission or ChannelBind Request

When a TURN server receives a CreatePermission or ChannelBind request for an allocation that included the CHECK-ALTERNATE attribute, it processes the request as per the TURN specification [[RFC5766](#)] plus the specific rules mentioned here.

If an XOR-OTHER-ADDRESS attribute is present, the server validates the number of XOR-PEER-ADDRESS attributes. If there is more than one XOR-PEER-ADDRESS attribute in the request, the server MUST reject the request with an error response using error code 400 (Bad Request). If there is only one XOR-PEER-ADDRESS attribute, the request is accepted and the value of the XOR-OTHER-ADDRESS attribute is stored with the permission state for use when checking for an alternate server.

The mechanism for deciding when and how to check for an alternate server is implementation dependent. This activity could be timer driven (e.g. check for an alternate server once every 120 seconds), it could be event driven (e.g. check for an alternate server on every permission or binding refresh), or another mechanism appropriate for the internal implementation could be chosen. Likewise, the decision of which specific address(es) to check for alternate servers is also implementation dependent. A server could check for alternates for

all active permissions, it could check just for permissions that have relayed non-ICE data, or another selection method appropriate for the implementation could be chosen.

When checking for an alternate server for a permission where the XOR-OTHER-ADDRESS attribute was provided, the server SHOULD use this address for peer location identification. Otherwise, the server SHOULD use the XOR-PEER-ADDRESS value.

The TURN client will retransmit a CreatePermission or ChannelBind request if the response is not received. [[I-D.ietf-tram-stunbis](#)] recommends that the retransmit timeout be greater than 500 ms, but does not require this, so it is important to avoid unnecessary delays in request processing. For this reason, the mechanism for driving alternate server checks SHOULD be asynchronous relative to processing of the associated CreatePermission or ChannelBind request and SHOULD NOT delay transmission of the response message.

### [3.5.](#) Forming a Redirect Indication

When an alternate server for a specific permission has been identified, the server notifies the client using a Redirect indication. The server MUST NOT send Redirect indications if the client did not indicate support by including a CHECK-ALTERNATE attribute in its Allocate request.

The codepoint for Redirect indication is TBD-RI.

A Redirect indication MUST contain a single ALTERNATE-SERVER attribute to provide the address for the alternate server. The message MAY contain one or more XOR-PEER-ADDRESS attributes to indicate a subset of peer addresses for redirection. If all peer addresses are to be redirected, no XOR-PEER-ADDRESS attribute is required. The message MUST contain either a MESSAGE-INTEGRITY or MESSAGE-INTEGRITY-SHA256 attribute (see [Section 5](#) for an explanation of the rationale).

Because this message codepoint is for indications only, the TURN client will not send a success response, and the TURN server will have no way to determine whether the message was received. For

improved reliability, the TURN server MAY retransmit the indication



multiple times, following the request retransmission semantics described in [[I-D.ietf-tram-stunbis](#)]. Retransmission requirements for indications might differ from those for requests, since requests are only retransmitted if no response was received. For this reason, an implementation that retransmits Redirect indications SHOULD provide separate configuration settings to control the maximum number of Redirect retransmissions and the minimum RT0.

### [3.6.](#) Receiving a Redirect Indication

When a TURN client receives a Redirect indication, it checks that the indication contains both an ALTERNATE-SERVER attribute and one of either a MESSAGE-INTEGRITY or a MESSAGE-INTEGRITY-SHA256 attribute and discards it if it does not. If XOR-PEER-ADDRESS attributes are present, it checks that all specified addresses are recognized peers for the allocation and discards the indication if any addresses are not recognized. Finally, it verifies the integrity attribute's value and discards the message if that value is invalid.

After validating the message, the ALTERNATE-SERVER value and associated peer addresses are delivered to the ICE implementation. Interactions with ICE are described below ([Section 4](#)).

See [Section 5](#) below for discussion of how the client should respond when receiving a Redirect indication when redirection was not requested.

## [4.](#) ICE Interactions

With "Vanilla" ICE as defined in [[RFC5245](#)], candidate gathering is complete before the offer/answer exchange. When a client using standard ICE receives a valid Redirect indication, it first checks whether it already has an active allocation on the specified server. If not, it adds the new relay to its list of servers and forms an allocation. This generates a new relayed candidate to be added to the local ICE candidates list, which requires ICE restart.

With trickle ICE as defined in [[I-D.ietf-ice-trickle](#)], if the end-of-candidates announcement has not yet been sent, it could be possible to complete the TURN Allocate request and include the new relayed candidate in the candidates list for the current ICE negotiation. However, on some implementations, it could be true that candidate gathering is already complete, even though the end-of-candidates announcement hasn't been sent. In other words, a trickle ICE agent might need to restart ICE in order to restart candidate gathering. If the end-of-candidates announcement has already been sent, ICE restart is necessary in order to add the new relayed candidate.

It is possible for a TURN relay to send multiple Redirect indications on the same allocation within a short time frame, each for a different set of peers. For example, consider the case of a remote peer with two interfaces, one wifi and the other 4G on a different Internet service provider. It is possible that the network geography for each interface requires a different relay for best performance and therefore a unique Redirect indication. After receiving a Redirect indication that does not apply to all peers associated with the allocation, it might be beneficial for the ICE agent to delay ICE restart by a small interval in order to avoid restarting ICE multiple times within a short time frame. However, selecting a good delay interval could be difficult, since the time between indications could vary due to packet loss and retransmission timeouts.

The authors have considered alternative Redirect indication formats that would allow all concurrent redirects to be provided in a single indication, which would avoid the above described issue. Feedback is desired on the question of whether the problem of minimizing ICE restarts is important enough to add greater complexity to the building and parsing of Redirect indications.

## [5.](#) Security Considerations

This section considers attacks that are possible in a TURN deployment through the specified protocol extension, and discusses how they are mitigated by mechanisms in the protocol or recommended practices in the implementation.

The specified mechanism affects the use of TURN CreatePermission request messages, ChannelBind request messages, and Redirect indication messages. Each of these TURN message types requires a STUN message integrity attribute (either MESSAGE-INTEGRITY or MESSAGE-INTEGRITY-SHA256), which limits attacks that attempt to make use of the specified mechanism to authenticated clients and servers.

### [5.1.](#) Permission Flood

A compromised TURN client could send a large number of CreatePermission or ChannelBind request messages with distinct peer address values, which would drive increased load on the TURN server. The mechanism described in this document does not make such an attack more likely, though it could make it possible to increase the impact of such an attack due to the additional load associated with determining whether an alternate server should be used by the client. The TURN server MAY be configured to disable or rate limit alternate server checks under some conditions in order to limit the associated

load. The conditions under which it is appropriate for a TURN server

to ignore disable or rate limit such checks are implementation dependent.

## [5.2.](#) Unsolicited or Invalid Redirect Indication

A compromised TURN server could send Redirect indications for allocations that did not include the CHECK-ALTERNATE attribute. For a client that does not support this mechanism, receiving such indications is no worse than receiving messages with any other unrecognized message type. A client that recognizes the message type MUST ignore Redirect indications for allocations where CHECK-ALTERNATE was not specified, and in particular, to avoid unnecessary authentication overhead, the client SHOULD drop such indications before attempting to validate the message integrity attribute.

A compromised TURN server could send an invalid ALTERNATE-SERVER attribute value in a Redirect indication message, where the value refers to an unaffiliated TURN server to which the sending TURN server is not allowed to redirect traffic. Such an attack is already allowed by the use of Try Alternate errors in response to Allocate request messages. Use of the ALTERNATE-SERVER attribute in the context of peer-specific redirection does not make such an attack more likely, though it could make it possible to increase the scale of such an attack by allowing multiple ALTERNATE-SERVER attributes to each client, one per requested permission or binding. A client SHOULD ignore all future Redirect indications received from the TURN server after an authentication failure with any server identified via an ALTERNATE-SERVER attribute. A client MAY discontinue use of the associated TURN allocation after an authentication failure with any server identified via an ALTERNATE-SERVER attribute.

An external attacker could send an invalid ALTERNATE-SERVER attribute value in a Redirect indication message. The client must have some way to detect when this occurs, which is the purpose of including a message integrity attribute in the Redirect indication. Without the message integrity attribute, it would be possible for an attacker to spoof a Redirect indication from the TURN server and drive the client to attempt to connect to a bad relay server.

### [5.3.](#) Replayed Redirect Indication

An in-path attacker could capture and replay Redirect indications. If the client has been redirected again after the replayed Redirect indication was received, the replay could drive the client to carry out unnecessary work to establish a new allocation and restart ICE if the results of the previous Redirect indication have since been discarded. It could also drive unexpected load from the client to a server that has since become overloaded, potentially degrading

performance for not only the target client but also all others now connected to the alternate server.

Multiple potential mitigations for this attack exist. For example, a client that maintains a complete list of all TURN servers used throughout the life of the session could keep track of the Transaction ID for each Redirect indication received, which would allow the client to recognize and reject a replayed indication. Alternatively, a client could rate-limit its responses to Redirect indications, requiring a configurable interval to expire between Redirect indications before accepting a new one.

The authors have also considered the option of adding a timestamp attribute to the Redirect indication message. The timestamp could be used to minimize the window of opportunity for a Redirect indication replay attack. However, such use of timestamps is fragile in the presence of potential clock skew problems between the client and the server and so has not been included in the specification.

## [6.](#) IANA Considerations

[Paragraphs below in braces should be removed by the RFC Editor upon publication]

[The CHECK-ALTERNATE attribute requires that IANA allocate a value in the "STUN Attributes Registry" from the comprehension-optional range (0x8000-0xFFFF), to be replaced for TBD-CA throughout this document]

This document defines the CHECK-ALTERNATE STUN attribute, described in [Section 3.1](#). IANA has allocated the comprehension-optional codepoint TBD-CA for this attribute.

[The XOR-OTHER-ADDRESS attribute requires that IANA allocate a value in the "STUN Attributes Registry" from the comprehension-optional range (0x8000-0xFFFF), to be replaced for TBD-XOA throughout this document]

This document defines the XOR-OTHER-ADDRESS STUN attribute, described in [Section 3.3](#). IANA has allocated the comprehension-optional codepoint TBD-XOA for this attribute.

[The Redirect indication codepoint requires that IANA allocate a value in the "STUN Methods Registry", to be replaced for TBD-RI throughout this document.]

This document defines the Redirect indication method type, described in [Section 3.5](#). IANA has allocated the codepoint TBD-RI for this method type.

## [7.](#) Acknowledgements

Many thanks to J. Uberti for his suggestions regarding ICE interactions.

## [8.](#) References

### [8.1.](#) Normative References

- [I-D.ietf-tram-stunbis]  
Petit-Huguenin, M., Salgueiro, G., Rosenberg, J., Wing, D., Mahy, R., and P. Matthews, "Session Traversal Utilities for NAT (STUN)", [draft-ietf-tram-stunbis-04](#) (work in progress), March 2015.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), DOI 10.17487/RFC2119, March 1997,  
<<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC5766] Mahy, R., Matthews, P., and J. Rosenberg, "Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN)", [RFC 5766](#), DOI 10.17487/RFC5766, April 2010,  
<<http://www.rfc-editor.org/info/rfc5766>>.

## [8.2.](#) Informative References

[I-D.ietf-ice-trickle]

Ivov, E., Rescorla, E., Uberti, J., and P. Saint-Andre, "Trickle ICE: Incremental Provisioning of Candidates for the Interactive Connectivity Establishment (ICE) Protocol", [draft-ietf-ice-trickle-00](#) (work in progress), October 2015.

[RFC5245] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", [RFC 5245](#), DOI 10.17487/RFC5245, April 2010, <<http://www.rfc-editor.org/info/rfc5245>>.

## [Appendix A.](#) Change History

[Note to RFC Editor: Please remove this section prior to publication.]

Williams & Reddy

Expires May 5, 2016

[Page 12]

---

Internet-Draft

Peer Redirect for TURN

November 2015

### [A.1.](#) Changes from version 03 to 04

Introduced Redirect indication to redefine the mechanism as push-like notification.

Moved CHECK-ALTERNATE to Allocation request.

Added a short section on ICE interactions.

Changed STUN reference to STUNbis, since the doc now references STUNbis content. Left other references as they are.

### [A.2.](#) Changes from version 02 to 03

Minor copy-editing.

### [A.3.](#) Changes from version 01 to 02

Add warning about the difference between physical geography and network geography.

Add load balancing use case.

#### [A.4.](#) Changes from version 00 to 01

Expand discussion of when/how to use CHECK-ALTERNATE and XOR-OTHER-ADDRESS.

#### Authors' Addresses

Brandon Williams  
Akamai, Inc.  
8 Cambridge Center  
Cambridge, MA 02142  
USA

Email: [brandon.williams@akamai.com](mailto:brandon.williams@akamai.com)

Tirumaleswar Reddy  
Cisco Systems, Inc.  
Cessna Business Park, Varthur Hobli  
Sarjapur Marathalli Outer Ring Road  
Bangalore, Karnataka 560103  
India

Email: [tiredy@cisco.com](mailto:tiredy@cisco.com)