TSVWG
Internet-Draft
Intended status: Standards Track
Expires: March 14, 2015

D. Wing
T. Reddy
Cisco Systems
B. Williams
Akamai, Inc.
R. Ravindranath
Cisco Systems

September 10, 2014

# TURN extension to convey flow characteristics
draft-wing-tsvwg-turn-flowdata-01

## Abstract

TURN server and the network in which it is hosted due to load could adversely impact the traffic relayed through it. During such high load event, it is desirable to shed some traffic but TURN server lack requirements about the flows to prioritize them. This document defines such a mechanism to communicate flow characteristics from the TURN client to its TURN server.

## Status of This Memo

## Copyright Notice

Table of Contents

## 1. Introduction

Traversal Using Relay NAT (TURN) [RFC5766] is a protocol that is often used to improve the connectivity of P2P applications. TURN allows a connection to be established when one or both sides is incapable of a direct P2P connection. A TURN server could be provided by an enterprise network, an access network, an application service provider or a third party provider. A TURN server could be used to relay media streams, WebRTC data channels [I-D.ietf-rtcweb-overview] , gaming, file transfer etc. A TURN server and the network in which it is hosted could have insufficient bandwidth or other characteristics that could adversely impact the traffic relayed through it and need a mechanism to identify and provide differentiated service to flows relayed through the TURN server.

This specification provides a mechanism for the client to signal the flow characteristics of a relay channel to the TURN server, so that certain relay channels can receive service that is differentiated from others. The TURN server authorizes the request and signals back to the client that it can (fully or partially) accommodate the flow. This sort of signaling will be useful for long-lived flows such as media streams, WebRTC data channels etc traversing through the TURN server. The TURN server can further communicate the flow information to a number of on-path devices in its network using a Policy decision Point (e.g. SDN controller). This way the network hosting the TURN server can accommodate the flow. With this mechanism, a TURN client can request the TURN server to provide certain characteristics for the relayed channel on both legs (client-to-server, server-to-peer). Applications using TURN as a communication relay would benefit from such an arrangement as it would improve the Quality of Experience (QOE) of the end user.

Note: It is not the intent of this document to advocate in favor of prioritizing relayed candidates over host, server-reflexive candidates, but to highlight the proposed mechanism only when TURN server is selected for various reasons like privacy, ICE connectivity checks with local host/server-reflexive candidates have failed etc.

## 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. Design considerations

1. TURN client can choose to either use Send and Data indications or channels to exchange data with its peer. For bandwidth intensive applications (like video, audio, WebRTC data channels) using Send indication or Data indication adds 36 bytes overhead to the application data and substantially increases the bandwidth required between the client and the server. Hence channels are commonly used for bandwidth intensive applications to exchange data. The other problem with using Send/Data indications is that if the TURN server determines that a flow can only be partially accommodated then this feedback cannot be conveyed back to the client. Hence in this specification focuses on conveying the flow characteristics only in ChannelBind request/response.

2. DSCP style markings can also help provide QOS, but has the following limitations:

   * DiffServ style packet marking can help provide QoS in some environments but DSCP markings are often modified or removed at various points in the network or when crossing network boundaries. DSCP markings set by the client may be modified or removed by the intervening network(s) before it reaches the TURN server.

  - DSCP values are site specific, with each site selecting its own code points for each QoS level, hence it may not work across domains. However [I-D.ietf-tsvwg-rtcweb-qos] recommends default set of DSCP values for browsers when there is no site specific information.

  - TURN client may not be able to set DSCP values for outgoing packets because of OS limitations.

  - DSCP provides differentiated service only in the outgoing direction of a flow.

The mechanism described in this document has none of the above limitations and the following useful properties:

- Usable at the application level to the TURN client, without needing operating system support.

- Robust metadata support, to convey sufficient information to the TURN server about the flow.

## 4. Solution Overview

When a channel binding is initiated by the client, it may also indicate certain characteristics of its flow to the TURN server. The TURN server uses that information to prioritize the flow in its network and signals back to the client that it can fully or partially accommodate the flow.

This specification defines one new comprehension-optional STUN attribute: FLOWDATA. If a TURN client wishes to signal the flow characteristics of the relay channel it MUST insert this attribute in ChannelBind request. This attribute if used MUST be sent only in the ChannelBind request. Other specifications in future may extend this attribute to be used in other STUN methods. The TURN server determines if it can accommodate that flow, making configuration changes if necessary to accommodate the flow, and returns information in the FLOWDATA attribute indicating its ability to accommodate the described flow.

### 4.1. Sending a ChannelBind Request

The TURN client sends ChannelBind request with the FLOWDATA STUN attribute to signal the flow characteristics of the relay channel to the TURN server. If the flow characteristics of a relay channel change then the client MAY send ChannelBind request with an updated FLOWDATA STUN attribute to refresh the binding. Similarly if the binding is refreshed using ChannelBind request then the client can also signal updated FLOWDATA STUN attribute if the flow characteristics of the relay channel have changed.

### 4.2. Receiving a ChannelBind Request

When a TURN server receives a ChannelBind request that includes a FLOWDATA attribute, it processes the request as per the TURN specification [RFC5766] plus the specific rules mentioned below.

The TURN server will determine if it can provide the flow resources requested by the client. The TURN server determines if the flow can be fully or partially accommodated, it returns values in the FLOWDATA fields that it can accommodate or returns 0 in those FLOWDATA fields where it has no information. In other words if the request indicated a low tolerance for delay but the TURN server determines that only high delay is available, the FLOWDATA response indicates high delay is available. The same sort of processing occurs on all of the FLOWDATA fields of the response (upstream and downstream delay tolerance, loss tolerance, jitter tolerance, minimum bandwidth, maximum bandwidth). If the TURN server determines the flow can only be partially accommodated and the client has also signaled CHECK-ALTERNATE attribute [I-D.williams-peer-redirect] then the TURN server MAY re-direct the client to an alternate TURN server that could accommodate the flow characteristics conveyed by the client.

If the TURN server can accommodate the flow as described in the FLOWDATA attribute, it sends a success response and includes the FLOWDATA attribute filled in according to Section 5. The TURN server SHOULD include the FLOWDATA attribute in ChannelBind response only when the client had signaled FLOWDATA attribute in ChannelBind request.

#### 4.2.1. Conflict Resolution

It is possible that two hosts send requests with different thresholds for delay or jitter in each direction for the same flow, and their requests arrive at the same TURN server. If this occurs, it is RECOMMENDED that the TURN server uses the stricter delay/loss tolerance (that is, the lower tolerance to delay or jitter). The diagram below depicts a conflict message flow.

```
   TURN Client A             TURN server          TURN Client B
        |                         |                       |
        |-loss=med, delay=med---->|<-loss=hi, delay=hi----|
        |                         |                       |
        |                    (conflict!)                  |
        |                         |                       |
        |                         |--loss=med, delay=med->|
        |                         |                       |
        |<--loss=med, delay=med---|                       |
```

Figure 1: Message diagram, resolving conflict

In the above example if the TURN server has already responded to client B before it receives the request from client A then the TURN server can correct the conflict only when the client B refreshes the binding.

### 4.3. Receiving a ChannelBind Response

When the client receives a ChannelBind success response, it proceeds with processing the response according to the TURN specification [RFC5766]. If the message does not include an FLOWDATA attribute, no additional processing is required. The returned FLOWDATA attribute, if present, indicates the accommodation of this flow the TURN server will perform. This document does not define what the TURN client might do with that information, but it could choose among several TURN servers or use it for other purposes.

## 5. FLOWDATA format

This section describes the format of the new STUN attribute FLOWDATA. FLOWDATA will have a type TBD-CA and length of 4 bytes. The FLOWDATA attribute in the request has the following format.

```
      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
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     | Attribute Type (TBD-CA)   |          Length (4)               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     | uDT | uLT | uJT |   RSVD1     | dDT | dLT | dJT |    RSVD2    |
     +---------------------------------------------------------------+
     |                  Upstream Minimum Bandwidth                   |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                 Downstream Minimum Bandwidth                  |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                  Upstream Maximum Bandwidth                   |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                 Downstream Maximum Bandwidth                  |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: FLOWDATA attribute in request

Description of the fields:

uDT: Upstream Delay Tolerance, 0=no information available, 1=very low, 2=low, 3=medium, 4=high.

uLT: Upstream Loss Tolerance, 0=no information available, 1=very low, 2=low, 3=medium, 4=high.

uJT: Upstream Jitter Tolerance, 0=no information available, 1=very low, 2=low, 3=medium, 4=high.

RSVD1: Reserved (7 bits), MUST be ignored on reception and MUST be 0 on transmission.

dDT: Downstream Delay Tolerance, 0=no information available, 1=very low, 2=low, 3=medium, 4=high.

dLT: Downstream Loss Tolerance, 0=no information available, 1=very low, 2=low, 3=medium, 4=high.

dJT: Downstream Jitter Tolerance, 0=no information available, 1=very low, 2=low, 3=medium, 4=high.

RSVD2: Reserved (7 bits), MUST be ignored on reception and MUST be 0 on transmission.

Upstream Minimum Bandwidth: Measures bandwidth sent by the client. Value is in octets per second. The value 0 means no information is available.

Downstream Minimum Bandwidth: Measures bandwidth sent to the client. Value is in octets per second. The value 0 means no information is available.

Upstream Maximum Bandwidth: Measures bandwidth sent by the client. Value is in octets per second. The value 0 means no information is available.

Downstream Maximum Bandwidth: Measures bandwidth sent to the client. Value is in octets per second. The value 0 means no information is available.

Different applications have different needs for their flows. The following table is derived from [RFC4594] to serve as a guideline for tolerance to loss, delay and jitter for some sample applications. The range 0 to 4 used for the fields in FLOWDATA attribute, meets the need to convey the tolerance levels for the traffic serviced by the service classes in the below table.

| Service Class | | Tolerance to | | |
|---|---|---|---|---|
| Name | Traffic Characteristics | Loss | Delay | Jitter |
| Network Control | Variable size packets, mostly inelastic short messages, but traffic can also burst (e.g., OSPF) | Low | Low | High |
| Telephony | Fixed-size small packets, constant emission rate, inelastic and low-rate flows (e.g., G.711, G.729) | Very Low | Very Low | Very Low |
| Signaling | Variable size packets, some what bursty short-lived flows | Low | Low | High |
| Multimedia Conferencing | Variable size packets, constant transmit interval, rate adaptive, reacts to loss | Low - Medium | Very Low | Low |
| Real-Time Interactive | RTP/UDP streams, inelastic, mostly variable rate | Low | Very Low | Low |
| Multimedia Streaming | Variable size packets, elastic with variable rate | Low - Medium | Medium | High |
| Broadcast Video | Constant and variable rate, inelastic, non-bursty flows | Very Low | Medium | Low |
| Low-Latency Data | Variable rate, bursty short-lived elastic flows | Low | Low - Medium | High |
| OAM | Variable size packets, elastic & inelastic flows | Low | Medium | High |
| High-Throughput Data | Variable rate, bursty long-lived elastic flows | Low | Medium - High | High |
| Standard | A bit of everything | 0 | 0 | 0 |
| Low-Priority Data | Non-real-time and elastic (e.g., network backup) | High | High | High |

Figure 3: Flow characteristics

The FLOWDATA attribute in the response has the following format

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Attribute Type (TBD-CA)   |          Length (4)                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| AuDT| AuLT| AuJT|   RSVD1     | AdDT| AdLT| AdJT|    RSVD2      |
+---------------------------------------------------------------+
|            Accommodated Upstream Minimum Bandwidth            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Accommodated Downstream Minimum Bandwidth           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Accommodated Upstream Maximum Bandwidth            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Accommodated Downstream Maximum Bandwidth           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4: FLOWDATA attribute in response

Description of the fields:

AuDT: Accommodated Upstream Delay Tolerance, 0=no information available, 1=able to accommodate very low, 2=able to accommodate low, 3=able to accommodate medium, 4=able to accommodate high.

AuLT: Accommodated Upstream Loss Tolerance, 0=no information available, 1=able to accommodate very low, 2=able to accommodate low, 3=able to accommodate medium, 4=able to accommodate high.

AuJT: Accommodated Upstream Jitter Tolerance, 0=no information available, 1=able to accommodate very low, 2=able to accommodate low, 3=able to accommodate medium, 4=able to accommodate high.

RSVD1: Reserved (7 bits), MUST be ignored on reception and MUST be 0 on transmission.

AdDT: Accommodated Downstream Delay Tolerance, 0=no information available, 1=able to accommodate very low, 2=able to accommodate low, 3=able to accommodate medium, 4=able to accommodate high.

AdLT: Accommodated Downstream Loss Tolerance, 0=no information available, 1=able to accommodate very low, 2=able to accommodate low, 3=able to accommodate medium, 4=able to accommodate high.

AdJT: Accommodated Downstream Jitter Tolerance, 0=no information available, 1=able to accommodate very low, 2=able to accommodate low, 3=able to accommodate medium, 4=able to accommodate high.

RSVD2: Reserved (7 bits), MUST be ignored on reception and MUST be 0 on transmission.

Accommodated Upstream Minimum Bandwidth: Bandwidth accommodated for this flow. Value in bytes per second. 0 means no information is available.

Accommodated Downstream Minimum Bandwidth: Bandwidth accommodated for this flow. Value in bytes per second. 0 means no information is available.

Accommodated Upstream Maximum Bandwidth: Bandwidth accommodated for this flow. Value in bytes per second. 0 means no information is available.

Accommodated Downstream Maximum Bandwidth: Bandwidth accommodated for this flow Value in bytes per second, 0 means no information is available.

## 6. Security Considerations

On some networks, only certain users or certain applications are authorized to signal the priority of a flow. This authorization can be achieved with STUN long-term authentication [RFC5389].

## 7. IANA Considerations

This document defines the FLOWDATA STUN attribute, described in Section 5. IANA has allocated the comprehension-optional codepoint TBD-CA for this attribute.

## 8. Acknowledgement

Authors would like to thank Anca Zamfir and Charles Eckel for their comments and review.

## 9. References

### 9.1. Normative References

[I-D.ietf-rtcweb-overview]
Alvestrand, H., "Overview: Real Time Protocols for Browser-based Applications", draft-ietf-rtcweb-overview-11 (work in progress), August 2014.

[I-D.ietf-tsvwg-rtcweb-qos]
Dhesikan, S., Jennings, C., Druta, D., Jones, P., and J. Polk, "DSCP and other packet markings for RTCWeb QoS", draft-ietf-tsvwg-rtcweb-qos-02 (work in progress), June 2014.

[I-D.williams-peer-redirect]
Williams, B. and T. Reddy, "Peer-specific Redirection for Traversal Using Relays around NAT (TURN)", draft-williams-peer-redirect-01 (work in progress), June 2014.

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

[RFC5389] Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, "Session Traversal Utilities for NAT (STUN)", RFC 5389, October 2008.

[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, April 2010.

## 9.2. Informative References

[RFC4594] Babiarz, J., Chan, K., and F. Baker, "Configuration Guidelines for DiffServ Service Classes", RFC 4594, August 2006.

## Authors’ Addresses

Dan Wing
Cisco Systems
821 Alder Drive
Milpitas, California 95035
USA

Email: dwing@cisco.com

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

Email: tireddy@cisco.com

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

Email: brandon.williams@akamai.com

Ram Mohan Ravindranath
Cisco Systems
Cessna Business Park
Sarjapur-Marathahalli Outer Ring Road
Bangalore, Karnataka  560103
India

Email: rmohanr@cisco.com