Network Working Group
Request for Comments: 3890
Category: Standards Track

M. Westerlund
Ericsson

September 2004

A Transport Independent Bandwidth Modifier for the Session Description Protocol (SDP)

Status of this Memo

Copyright Notice

Abstract

This document defines a Session Description Protocol (SDP) Transport Independent Application Specific Maximum (TIAS) bandwidth modifier that does not include transport overhead; instead an additional packet rate attribute is defined. The transport independent bit-rate value together with the maximum packet rate can then be used to calculate the real bit-rate over the transport actually used.

The existing SDP bandwidth modifiers and their values include the bandwidth needed for the transport and IP layers. When using SDP with protocols like the Session Announcement Protocol (SAP), the Session Initiation Protocol (SIP), and the Real-Time Streaming Protocol (RTSP), and when the involved hosts has different transport overhead, for example due to different IP versions, the interpretation of what lower layer bandwidths are included is not clear.

Table of Contents

1. Introduction

This specification is structured in the following way: In this section, some information regarding SDP bandwidth modifiers, and different mechanisms that affect transport overhead are asserted. In section 3, the problems found are described, including problems that are not solved by this specification. In section 4 the scope of the problems this specification solves is presented. Section 5 contains the requirements applicable to the problem scope. Section 6 defines the solution, which is a new bandwidth modifier, and a new maximum packet rate attribute. Section 7 looks at the protocol interaction for SIP, RTSP, and SAP. The security considerations are discussed in section 8. The remaining sections are the necessary IANA considerations, acknowledgements, reference list, author's address, and copyright and IPR notices.

Today the Session Description Protocol (SDP) [1] is used in several types of applications. The original application is session information and configuration for multicast sessions announced with Session Announcement Protocol (SAP) [5]. SDP is also a vital component in media negotiation for the Session Initiation Protocol (SIP) [6] by using the offer answer model [7]. The Real-Time Streaming Protocol (RTSP) [8] also makes use of SDP to declare to the client what media and codec(s) comprise a multi-media presentation.

1.1. The Bandwidth Attribute

In SDP [1] there exists a bandwidth attribute, which has a modifier used to specify what type of bit-rate the value refers to. The attribute has the following form:

```
b=<modifier>:<value>
```

Today there are four defined modifiers used for different purposes.

1.1.1. Conference Total

The Conference Total is indicated by giving the modifier "CT". Conference total gives a maximum bandwidth that a conference session will use. Its purpose is to decide if this session can co-exist with any other sessions, defined in RFC 2327 [1].

1.1.2. Application Specific Maximum

The Application Specific maximum bandwidth is indicated by the modifier "AS". The interpretation of this attribute is dependent on the application's notion of maximum bandwidth. For an RTP application, this attribute is the RTP session bandwidth as defined

in RFC 3550 [4]. The session bandwidth includes the bandwidth that the RTP data traffic will consume, including the lower layers, down to the IP layer. Therefore, the bandwidth is in most cases calculated over RTP payload, RTP header, UDP, and IP, defined in RFC 2327 [1].

1.1.3. RTCP Report Bandwidth

In RFC 3556 [9], two bandwidth modifiers are defined. These modifiers, "RS" and "RR", define the amount of bandwidth that is assigned for RTCP reports by active data senders and RTCP reports by other participants (receivers), respectively.

1.2. IPv6 and IPv4

Today there are two IP versions, 4 [14] and 6 [13], used in parallel on the Internet, creating problems. However, there exist a number of possible transition mechanisms.

- The nodes which wish to communicate must share the IP version; typically this is done by deploying dual-stack nodes. For example, an IPv4 only host cannot communicate with an IPv6 only host.

- If communication between nodes which do not share a protocol version is required, use of a translation or proxying mechanism would be required. Work is underway to specify such a mechanism for this purpose.

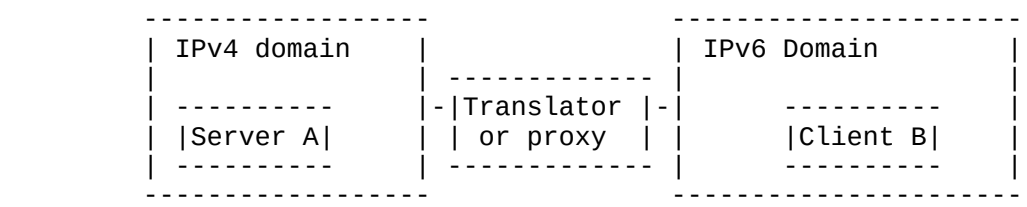

Figure 1. Translation or proxying between IPv6 and IPv4 addresses.

- IPv6 nodes belonging to different domains running IPv6, but lacking IPv6 connectivity between them, solve this by tunneling over the IPv4 net, see Figure 2. Basically, the IPv6 packets are sent as payload in IPv4 packets between the tunneling end-points at the edge of each IPv6 domain. The bandwidth required over the IPv4 domain will be different from IPv6 domains. However, as the tunneling is normally not performed by the application end-point, this scenario can not usually be taken into consideration.

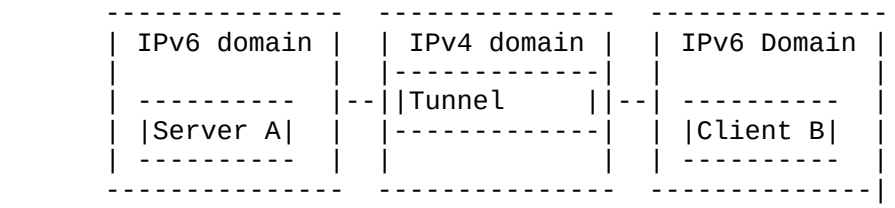

Figure 2. Tunneling through a IPv4 domain

IPv4 has a minimum header size of 20 bytes, while the fixed part of the IPv6 header is 40 bytes.

The difference in header sizes means that the bit-rate required for the two IP versions is different. The significance of the difference depends on the packet rate and payload size of each packet.

1.3. Further Mechanisms that Change the Bandwidth Utilization

There exist a number of other mechanisms that also may change the overhead at layers below media transport. We will briefly cover a few of these here.

1.3.1. IPsec

IPsec [19] can be used between end points to provide confidentiality through the application of the IP Encapsulating Security Payload (ESP) [21] or integrity protection using the IP Authentication Header (AH) [20] of the media stream. The addition of the ESP and AH headers increases each packet's size.

To provide virtual private networks, complete IP packets may be encapsulated between an end node and the private networks security gateway, thus providing a secure tunnel that ensures confidentiality, integrity, and authentication of the packet stream. In this case, the extra IP and ESP header will significantly increase the packet size.

1.3.2. Header Compression

Another mechanism that alters the actual overhead over links is header compression. Header compression uses the fact that most network protocol headers have either static or predictable values in their fields within a packet stream. Compression is normally only done on a per hop basis, i.e., on a single link. The normal reason for doing header compression is that the link has fairly limited bandwidth and significant gain in throughput is achieved.

There exist several different header compression standards. For compressing IP headers only, there is RFC 2507 [10]. For compressing packets with IP/UDP/RTP headers, CRTP [11] was created at the same time. More recently, the Robust Header Compression (ROHC) working group has been developing a framework and profiles [12] for compressing certain combinations of protocols, like IP/UDP, and IP/UDP/RTP.

2. Definitions

2.1. Glossary

ALG - Application Level Gateway.
bps - bits per second.
RTSP - Real-Time Streaming Protocol, see [8].
SDP - Session Description Protocol, see [1].
SAP - Session Announcement Protocol, see [5].
SIP - Session Initiation Protocol, see [6].
TIAS - Transport Independent Application Specific maximum, a bandwidth modifier.

2.2. 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 BCP 14, RFC 2119 [3].

3. The Bandwidth Signaling Problems

When an application wants to use SDP to signal the bandwidth required for this application, some problems become evident due to the inclusion of the lower layers in the bandwidth values.

3.1. What IP Version is Used

If one signals the bandwidth in SDP, for example, using "b=AS:" as an RTP based application, one cannot know if the overhead is calculated for IPv4 or IPv6. An indication of which protocol has been used when calculating the bandwidth values is given by the "c=" connection address line. This line contains either a multicast group address or a unicast address of the data source or sink. The "c=" line's address type may be assumed to be of the same type as the one used in the bandwidth calculation, although no document specifying this point seems to exist.

In cases of SDP transported by RTSP, this is even less clear. The normal usage for a unicast on-demand streaming session is to set the connection data address to a null address. This null address does

have an address type, which could be used as an indication. However, this is also not clarified anywhere.

Figure 1, illustrates a connection scenario between a streaming server A and a client B over a translator. When B receives the SDP from A over RTSP, it will be very difficult for B to know what the bandwidth values in the SDP represent. The following possibilities exist:

1. The SDP is unchanged and the "c=" null address is of type IPv4. The bandwidth value represents the bandwidth needed in an IPv4 network.

2. The SDP has been changed by an Application Level Gateway (ALG). The "c=" address is changed to an IPv6 type. The bandwidth value is unchanged.

3. The SDP is changed and both "c=" address type and bandwidth value is converted. Unfortunately, this can seldom be done, see 3.3.

In case 1, the client can understand that the server is located in an IPv4 network and that it uses IPv4 overhead when calculating the bandwidth value. The client can almost never convert the bandwidth value, see section 3.3.

In case 2, the client does not know that the server is in an IPv4 network and that the bandwidth value is not calculated with IPv6 overhead. In cases where a client uses this value to determine if its end of the network has sufficient resources the client will underestimate the required bit-rate, potentially resulting in bad application performance.

In case 3, everything works correctly. However, this case will be very rare. If one tries to convert the bandwidth value without further information about the packet rate, significant errors may be introduced into the value.

3.2. Taking Other Mechanisms into Account

Section 1.2 and 1.3 lists a number of reasons, like header compression and tunnels, that would change lower layer header sizes. For these mechanisms there exist different possibilities to take them into account.

Using IPsec directly between end-points should definitely be known to the application, thus enabling it to take the extra headers into account. However the same problem also exists with the current SDP bandwidth modifiers where a receiver is not able to convert these values taking the IPsec headers into account.

It is less likely that an application would be aware of the existence of a virtual private network. Thus the generality of the mechanism to tunnel all traffic may prevent the application from even considering whether it would be possible to convert the values.

When using header compression, the actual overhead will be less deterministic, but in most cases an average overhead can be determined for a certain application. If a network node knows that some type of header compression is employed, this can be taken into consideration. For RSVP [15], there exists an extension, RFC 3006 [16], that allows the data sender to inform network nodes about the compressibility of the data flow. To be able to do this with any accuracy, the compression factor and packet rate or size is needed, as RFC 3006 provides.

3.3. Converting Bandwidth Values

If one would like to convert a bandwidth value calculated using IPv4 overhead to IPv6 overhead, the packet rate is required. The new bandwidth value for IPv6 is normally "IPv4 bandwidth" + "packet rate" * 20 bytes, where 20 bytes is the usual difference between IPv6 and IPv4 headers. The overhead difference may be some other value in cases when IPv4 options [14] or IPv6 extension headers [13] are used.

As converting requires the packet rate of the stream, this is not possible in the general case. Many codecs have either multiple possible packet/frame rates or can perform payload format aggregation, resulting in many possible rates. Therefore, some extra information in the SDP will be required. The "a=ptime:" parameter may be a possible candidate. However, this parameter is normally only used for audio codecs. Its definition [1] is that it is only a recommendation, which the sender may disregard. A better parameter is needed.

3.4. RTCP Problems

When RTCP is used between hosts in IPv4 and IPv6 networks over translator, similar problems exist. The RTCP traffic going from the IPv4 domain will result in a higher RTCP bit-rate than intended in the IPv6 domain due to the larger headers. This may result in up to a 25% increase in required bandwidth for the RTCP traffic. The largest increase will be for small RTCP packets when the number of

IPv4 hosts is much larger than the number of IPv6 hosts. Fortunately, as RTCP has a limited bandwidth compared to RTP, it will only result in a maximum of 1.75% increase of the total session bandwidth when RTCP bandwidth is 5% of RTP bandwidth. The RTCP randomization may easily result in short term effects of the same magnitude, so this increase may be considered tolerable. The increase in bandwidth will in most cases be less.

At the same time, this results in unfairness in the reporting between an IPv4 and IPv6 node. In the worst case scenario, the IPv6 node may report with 25% longer intervals.

These problems have been considered insignificant enough to not be worth any complex solutions. Therefore, only a simple algorithm for deriving RTCP bandwidth is defined in this specification.

3.5. Future Development

Today there is work in the IETF to design a new datagram transport protocol suitable for real-time media. This protocol is called the Datagram Congestion Control Protocol (DCCP). It will most probably have a different header size than UDP, which is the protocol most often used for real-time media today. This results in even more possible transport combinations. This may become a problem if one has the possibility of using different protocols, which will not be determined prior to actual protocol SETUP. Thus, pre-calculating this value will not be possible, which is one further motivation why a transport independent bandwidth modifier is needed.

DCCP's congestion control algorithms will control how much bandwidth can really be utilized. This may require further work with specifying SDP bandwidth modifiers to declare the dynamic possibilities of an application's media stream. For example, min and max media bandwidth the application is capable of producing at all, or for media codecs only capable of producing certain bit-rates, enumerating possible rates. However, this is for future study and outside the scope of the present solution.

3.6. Problem Conclusion

A shortcoming of the current SDP bandwidth modifiers is that they also include the bandwidth needed for lower layers. It is in many cases difficult to determine which lower layers and their versions were included in the calculation, especially in the presence of translation or proxying between different domains. This prevents a receiver from determining if given bandwidth needs to be converted based on the actual lower layers being used.

Secondly, an attribute to give the receiver an explicit determination of the maximum packet rate that will be used does not exist. This value is necessary for accurate conversion of any bandwidth values if the difference in overhead is known.

4. Problem Scope

The problems described in section 3 are common and effect application level signaling using SDP, other signaling protocols, and also resource reservation protocols. However, this document targets the specific problem of signaling the bit-rate in SDP. The problems need to be considered in other affected protocols and in new protocols being designed. In the MMUSIC WG there is work on a replacement of SDP called SDP-NG. It is recommended that the problems outlined in this document be considered when designing solutions for specifying bandwidth in the SDP-NG [17].

As this specification only targets carrying the bit-rate information within SDP, it will have a limited applicability. As SDP information is normally transported end-to-end by an application protocol, nodes between the end-points will not have access to the bit-rate information. It will normally only be the end points that are able to take this information into account. An interior node will need to receive the information through a means other than SDP, and that is outside the scope of this specification.

Nevertheless, the bit-rate information provided in this specification is sufficient for cases such as first-hop resource reservation and admission control. It also provide information about the maximum codec rate, which is independent of lower-level protocols.

This specification does NOT try to solve the problem of detecting NATs or other middleboxes.

5. Requirements

The problems outlined in the preceding sections and with the above applicability, should meet the following requirements:

- The bandwidth value SHALL be given in a way such that it can be calculated for all possible combinations of transport overhead.

6. Solution

6.1. Introduction

This chapter describes a solution for the problems outlined in this document for the Application Specific (AS) bandwidth modifier, thus enabling the derivation of the required bit-rate for an application, or RTP session's data and RTCP traffic. The solution is based upon the definition of a new Transport Independent Application Specific (TIAS) bandwidth modifier and a new SDP attribute for the maximum packet rate (maxprate).

The CT is a session level modifier and cannot easily be dealt with. To address the problems with different overhead, it is RECOMMENDED that the CT value be calculated using reasonable worst case overhead. An example of how to calculate a reasonable worst case overhead is: Take the overhead of the largest transport protocol (using average size if variable), add that to the largest IP overhead that is expected for use, plus the data traffic rate. Do this for every individual media stream used in the conference and add them together.

The RR and RS modifiers [9] will be used as defined and include transport overhead. The small unfairness between hosts is deemed acceptable.

6.2. The TIAS Bandwidth Modifier

6.2.1. Usage

A new bandwidth modifier is defined to be used for the following purposes:

- Resource reservation. A single bit-rate can be enough for use as a resource reservation. Some characteristics can be derived from the stream, codec type, etc. In cases where more information is needed, another SDP parameter will be required.

- Maximum media codec rate. With the definition below of "TIAS", the given bit-rate will mostly be from the media codec. Therefore, it gives a good indication of the maximum codec bit-rate required to be supported by the decoder.

- Communication bit-rate required for the stream. The "TIAS" value together with "maxprate" can be used to determine the maximum communication bit-rate the stream will require. Using session level values or by adding all maximum bit-rates from the streams in a session together, a receiver can determine if its communication resources are sufficient to handle the stream. For

example, a modem user can determine if the session fits his modem's capabilities and the established connection.

- Determine the RTP session bandwidth and derive the RTCP bandwidth. The derived transport dependent attribute will be the RTP session bandwidth in case of RTP based transport. The TIAS value can also be used to determine the RTCP bandwidth to use when using implicit allocation. RTP [4] specifies that if not explicitly stated, additional bandwidth, equal to 5% of the RTP session bandwidth, shall be used by RTCP. The RTCP bandwidth can be explicitly allocated by using the RR and RS modifiers defined in [9].

6.2.2. Definition

A new session and media level bandwidth modifier is defined:

b=TIAS:<bandwidth-value> ; see section 6.6 for ABNF definition.

The Transport Independent Application Specific Maximum (TIAS) bandwidth modifier has an integer bit-rate value in bits per second. A fractional bandwidth value SHALL always be rounded up to the next integer. The bandwidth value is the maximum needed by the application (SDP session level) or media stream (SDP media level) without counting IP or other transport layers like TCP or UDP.

At the SDP session level, the TIAS value is the maximal amount of bandwidth needed when all declared media streams are used. This MAY be less than the sum of all the individual media streams values. This is due to the possibility that not all streams have their maximum at the same point in time. This can normally only be verified for stored media streams.

For RTP transported media streams, TIAS at the SDP media level can be used to derive the RTP "session bandwidth", defined in section 6.2 of [4]. In the context of RTP transport, the TIAS value is defined as:

> Only the RTP payload as defined in [4] SHALL be used in the calculation of the bit-rate, i.e., excluding the lower layers (IP/UDP) and RTP headers including RTP header, RTP header extensions, CSRC list, and other RTP profile specific fields. Note that the RTP payload includes both the payload format header and the data. This may allow one to use the same value for RTP-based media transport, non-RTP transport, and stored media.

Note 1: The usage of bps is not in accordance with RFC 2327 [1]. This change has no implications on the parser, only the interpreter of the value must be aware. The change is done to allow for better resolution, and has also been used for the RR and RS bandwidth modifiers, see [9].

Note 2: RTCP bandwidth is not included in the bandwidth value. In applications using RTCP, the bandwidth used by RTCP is either 5% of the RTP session bandwidth including lower layers or as specified by the RR and RS modifiers [9]. A specification of how to derive the RTCP bit-rate when using TIAS is presented in chapter 6.5.

6.2.3. Usage Rules

"TIAS" is primarily intended to be used at the SDP media level. The "TIAS" bandwidth attribute MAY be present at the session level in SDP, if all media streams use the same transport. In cases where the sum of the media level values for all media streams is larger than the actual maximum bandwidth need for all streams, it SHOULD be included at session level. However, if present at the session level it SHOULD be present also at the media level. "TIAS" SHALL NOT be present at the session level unless the same transport protocols is used for all media streams. The same transport is used as long as the same combination of protocols is used, like IPv6/UDP/RTP.

To allow for backwards compatibility with applications of SDP that do not implement "TIAS", it is RECOMMENDED to also include the "AS" modifier when using "TIAS". The presence of a value including lower-layer overhead, even with its problems, is better than none. However, an SDP application implementing TIAS SHOULD ignore the "AS" value and use "TIAS" instead when both are present.

When using TIAS for an RTP-transported stream, the "maxprate" attribute, if possible to calculate, defined next, SHALL be included at the corresponding SDP level.

6.3. Packet Rate Parameter

To be able to calculate the bandwidth value including the lower layers actually used, a packet rate attribute is also defined.

The SDP session and media level maximum packet rate attribute is defined as:

```
a=maxprate:<packet-rate> ; see section 6.6 for ABNF definition.
```

The <packet-rate> is a floating-point value for the stream's maximum packet rate in packets per second. If the number of packets is variable, the given value SHALL be the maximum the application can produce in case of a live stream, or for stored on-demand streams, has produced. The packet rate is calculated by adding the number of packets sent within a 1 second window. The maxprate is the largest value produced when the window slides over the entire media stream. In cases that this can't be calculated, i.e., a live stream, a estimated value of the maximum packet rate the codec can produce for the given configuration and content SHALL be used.

Note: The sliding window calculation will always yield an integer number. However the attributes field is a floating-point value because the estimated or known maximum packet rate per second may be fractional.

At the SDP session level, the "maxprate" value is the maximum packet rate calculated over all the declared media streams. If this can't be measured (stored media) or estimated (live), the sum of all media level values provides a ceiling value. Note: the value at session level can be less then the sum of the individual media streams due to temporal distribution of media stream's maximums. The "maxprate" attribute MUST NOT be present at the session level if the media streams use different transport. The attribute MAY be present if the media streams use the same transport. If the attribute is present at the session level, it SHOULD also be present at the media level for all media streams.

"maxprate" SHALL be included for all transports where a packet rate can be derived and TIAS is included. For example, if you use TIAS and a transport like IP/UDP/RTP, for which the max packet rate (actual or estimated) can be derived, then "maxprate" SHALL be included. However, if either (a) the packet rate for the transport cannot be derived, or (b) TIAS is not included, then, "maxprate" is not required to be included.

6.4. Converting to Transport-Dependent Values

When converting the transport-independent bandwidth value (bw-value) into a transport-dependent value including the lower layers, the following MUST be done:

1. Determine which lower layers will be used and calculate the sum of the sizes of the headers in bits (h-size). In cases of variable header sizes, the average size SHALL be used. For RTP-transported media, the lower layers SHALL include the RTP header with header extensions, if used, the CSRC list, and any profile-specific extensions.

2. Retrieve the maximum packet rate from the SDP (prate = maxprate).

3. Calculate the transport overhead by multiplying the header sizes by the packet rate (t-over = h-size * prate).

4. Round the transport overhead up to nearest integer in bits (t-over = CEIL(t-over)).

5. Add the transport overhead to the transport independent bandwidth value (total bit-rate = bw-value + t-over)

When the above calculation is performed using the "maxprate", the bit-rate value will be the absolute maximum the media stream may use over the transport assumed in the calculations.

6.5. Deriving RTCP Bandwidth

This chapter does not solve the fairness and possible bit-rate change introduced by IPv4 to IPv6 translation. These differences are considered small enough, and known solutions introduce code changes to the RTP/RTCP implementation. This section provides a consistent way of calculating the bit-rate to assign to RTCP, if not explicitly given.

First the transport-dependent RTP session bit-rate is calculated, in accordance with section 6.4, using the actual transport layers used at the end point where the calculation is done. The RTCP bit-rate is then derived as usual based on the RTP session bandwidth, i.e., normally equal to 5% of the calculated value.

6.5.1. Motivation for this Solution

Giving the exact same RTCP bit-rate value to both the IPv4 and IPv6 hosts will result in the IPv4 host having a higher RTCP sending rate. The sending rate represents the number of RTCP packets sent during a given time interval. The sending of RTCP is limited according to rules defined in the RTP specification [4]. For a 100-byte RTCP packet (including UDP/IPv4), the IPv4 sender has an approximately 20% higher sending rate. This rate falls with larger RTCP packets. For example, 300-byte packets will only give the IPv4 host a 7% higher sending rate.

The above rule for deriving RTCP bandwidth gives the same behavior as fixed assignment when the RTP session has traffic parameters giving a large TIAS/maxprate ratio. The two hosts will be fair when the TIAS/maxprate ratio is approximately 40 bytes/packet, given 100-byte RTCP packets. For a TIAS/maxprate ratio of 5 bytes/packet, the IPv6 host will be allowed to send approximately 15-20% more RTCP packets.

The larger the RTCP packets become, the more it will favor the IPv6 host in its sending rate.

The conclusions is that, within the normal useful combination of transport-independent bit rates and packet rates, the difference in fairness between hosts on different IP versions with different overhead is acceptable. For the 20-byte difference in overhead between IPv4 and IPv6 headers, the RTCP bandwidth actually used in a unicast connection case will not be larger than approximately 1% of the total session bandwidth.

6.6. ABNF Definitions

This chapter defines in ABNF from RFC 2234 [2] the bandwidth modifier and the packet rate attribute.

The bandwidth modifier:

```
TIAS-bandwidth-def = "b" "=" "TIAS" ":" bandwidth-value CRLF

bandwidth-value = 1*DIGIT
```

The maximum packet rate attribute:

```
max-p-rate-def = "a" "=" "maxprate" ":" packet-rate CRLF

packet-rate = 1*DIGIT ["." 1*DIGIT]
```

6.7. Example

```
v=0
o=Example_SERVER 3413526809 0 IN IP4 server.example.com
s=Example of TIAS and maxprate in use
c=IN IP4 0.0.0.0
b=AS:60
b=TIAS:50780
t=0 0
a=control:rtsp://server.example.com/media.3gp
a=range:npt=0-150.0
a=maxprate:28.0
m=audio 0 RTP/AVP 97
b=AS:12
b=TIAS:8480
a=maxprate:10.0
a=rtpmap:97 AMR/8000
a=fmtp:97 octet-align;
a=control:rtsp://server.example.com/media.3gp/trackID=1
m=video 0 RTP/AVP 99
```

```
b=AS:48
b=TIAS:42300
a=maxprate:18.0
a=rtpmap:99 MP4V-ES/90000
a=fmtp:99 profile-level-id=8;
config=000001B008000001B509000001010000012000884006682C2090A21F
a=control:rtsp://server.example.com/media.3gp/trackID=3
```

In this SDP example of a streaming session's SDP, there are two media streams, one audio stream encoded with AMR and one video stream encoded with the MPEG-4 Video encoder. AMR is used here to produce a constant rate media stream and uses a packetization resulting in 10 packets per second. This results in a TIAS bandwidth rate of 8480 bits per second, and the claimed 10 packets per second. The video stream is more variable. However, it has a measured maximum payload rate of 42,300 bits per second. The video stream also has a variable packet rate, despite the fact that the video is 15 frames per second, where at least one instance in a second long window contains 18 packets.

7. Protocol Interaction

7.1. RTSP

The "TIAS" and "maxprate" parameters can be used with RTSP as currently specified. To be able to calculate the transport dependent bandwidth, some of the transport header parameters will be required. There should be no problem for a client to calculate the required bandwidth(s) prior to an RTSP SETUP. The reason is that a client supports a limited number of transport setups. The one actually offered to a server in a SETUP request will be dependent on the contents of the SDP description. The "m=" line(s) will signal the desired transport profile(s) to the client.

7.2. SIP

The usage of "TIAS" together with "maxprate" should not be different from the handling of the "AS" modifier currently in use. The needed transport parameters will be available in the transport field in the "m=" line. The address class can be determined from the "c=" field and the client's connectivity.

7.3. SAP

In the case of SAP, all available information to calculate the transport dependent bit-rate should be present in the SDP. The "c=" information gives the address family used for the multicast. The transport layer, e.g., RTP/UDP, for each media is evident in the media line ("m=") and its transport field.

8. Security Consideration

The bandwidth value that is supplied by the parameters defined here can be altered, if not integrity protected. By altering the bandwidth value, one can fool a receiver into reserving either more or less bandwidth than actually needed. Reserving too much may result in unwanted expenses on behalf of the user, while also blocking resources that other parties could have used. If too little bandwidth is reserved, the receiving user's quality may be effected. Trusting a too-large TIAS value may also result in the receiver rejecting the session due to insufficient communication and decoding resources.

Due to these security risks, it is strongly RECOMMENDED that the SDP be integrity protected and source authenticated so tampering can not be performed, and the source can be trusted. It is also RECOMMENDED that any receiver of the SDP perform an analysis of the received bandwidth values to verify that they are reasonable expected values for the application. For example, a single channel AMR-encoded voice stream claiming to use 1000 kbps is not reasonable.

Please note that some of the above security requirements are in conflict with that required to make signaling protocols using SDP work through a middlebox, as discussed in the security considerations of RFC 3303 [18].

9. IANA Considerations

This document registers one new SDP session and media level attribute "maxprate", see section 6.3.

A new SDP [1] bandwidth modifier (bwtype) "TIAS" is also registered in accordance with the rules requiring a standards-track RFC. The modifier is defined in section 6.2.

10. Acknowledgments

The author would like to thank Gonzalo Camarillo and Hesham Soliman for their work reviewing this document. A very big thanks goes to Stephen Casner for reviewing and helping fix the language, and identifying some errors in the previous versions. Further thanks for suggestion to improvements go to Colin Perkins, Geetha Srikantan, and Emre Aksu.

The author would also like to thank all persons on the MMUSIC working group's mailing list that have commented on this specification.

11. References

11.1. Normative References

[1] Handley, M. and V. Jacobson, "SDP: Session Description Protocol", RFC 2327, April 1998.

[2] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, November 1997.

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

[4] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.

11.2. Informative References

[5] Handley, M., Perkins, C., and E. Whelan, "Session Announcement Protocol", RFC 2974, October 2000.

[6] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[7] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, June 2002.

[8] Schulzrinne, H., Rao, A., and R. Lanphier, "Real Time Streaming Protocol (RTSP)", RFC 2326, April 1998.

[9] Casner, S., "Session Description Protocol (SDP) Bandwidth Modifiers for RTP Control Protocol (RTCP) Bandwidth", RFC 3556, July 2003.

[10] Degermark, M., Nordgren, B., and S. Pink, "IP Header Compression", RFC 2507, February 1999.

[11] Casner, S. and V. Jacobson, "Compressing IP/UDP/RTP Headers for Low-Speed Serial Links", RFC 2508, February 1999.

[12] Bormann, C., Burmeister, C., Degermark, M., Fukushima, H., Hannu, H., Jonsson, L-E., Hakenberg, R., Koren, T., Le, K., Liu, Z., Martensson, A., Miyazaki, A., Svanbro, K., Wiebke, T., Yoshimura, T., and H. Zheng, "RObust Header Compression (ROHC): Framework and four profiles: RTP, UDP, ESP, and uncompressed ", RFC 3095, July 2001.

[13] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

[14] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.

[15] Braden, R., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, September 1997.

[16] Davie, B., Iturralde, C., Oran, D., Casner, S., and J. Wroclawski, "Integrated Services in the Presence of Compressible Flows", RFC 3006, November 2000.

[17] Kutscher, Ott, Bormann, "Session Description and Capability Negotiation," Work in Progress, March 2003.

[18] Srisuresh, P., Kuthan, J., Rosenberg, J., Molitor, A., and A. Rayhan, "Middlebox communication architecture and framework", RFC 3303, August 2002.

[19] Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.

[20] Kent, S. and R. Atkinson, "IP Authentication Header", RFC 2402, November 1998.

[21] Kent, S. and R. Atkinson, "IP Encapsulating Security Payload (ESP)", RFC 2406, November 1998.

12. Author's Address

Magnus Westerlund
Ericsson Research
Ericsson AB
Torshamnsgatan 23
SE-164 80 Stockholm, SWEDEN

Phone: +46 8 7190000
EMail: Magnus.Westerlund@ericsson.com

13. Full Copyright Statement

Intellectual Property

Acknowledgement

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