Network Working Group
Request for Comments: 4412
Category: Standards Track
H. Schulzrinne
Columbia U.
J. Polk
Cisco Systems
February 2006

Communications Resource Priority for the Session Initiation Protocol (SIP)

Status of This Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (2006).

Abstract

This document defines two new Session Initiation Protocol (SIP) header fields for communicating resource priority, namely, "Resource-Priority" and "Accept-Resource-Priority". The "Resource-Priority" header field can influence the behavior of SIP user agents (such as telephone gateways and IP telephones) and SIP proxies. It does not directly influence the forwarding behavior of IP routers.

Table of Contents

1. Introduction

During emergencies, communications resources (including telephone circuits, IP bandwidth, and gateways between the circuit-switched and IP networks) may become congested. Congestion can occur due to heavy usage, loss of resources caused by the natural or man-made disaster, and attacks on the network during man-made emergencies. This congestion may make it difficult for persons charged with emergency assistance, recovery, or law enforcement to coordinate their efforts. As IP networks become part of converged or hybrid networks, along with public and private circuit-switched (telephone) networks, it becomes necessary to ensure that these networks can assist during such emergencies.

Also, users may want to interrupt their lower-priority communications activities and dedicate their end-system resources to the high-priority communications attempt if a high-priority communications request arrives at their end system.

There are many IP-based services that can assist during emergencies. This memo only covers real-time communications applications involving the Session Initiation Protocol (SIP) [RFC3261], including voice-over-IP, multimedia conferencing, instant messaging, and presence.

SIP applications may involve at least five different resources that may become scarce and congested during emergencies. These resources include gateway resources, circuit-switched network resources, IP network resources, receiving end-system resources, and SIP proxy resources. IP network resources are beyond the scope of SIP signaling and are therefore not considered here.

Even if the resources at the SIP element itself are not scarce, a SIP gateway may mark outgoing calls with an indication of priority, e.g., on an ISUP (ISDN User Part) IAM (Initial Address Message) originated by a SIP gateway with the Public Switched Telephone Network (PSTN).

In order to improve emergency response, it may become necessary to prioritize access to SIP-signaled resources during periods of emergency-induced resource scarcity. We call this "resource prioritization". The mechanism itself may well be in place at all times, but may only materially affect call handling during times of resource scarcity.

Currently, SIP does not include a mechanism that allows a request originator to indicate to a SIP element that it wishes the request to invoke such resource prioritization. To address this need, this document adds a SIP protocol element that labels certain SIP requests.

This document defines (Section 3) two new SIP header fields for communications resource priority, called 'Resource-Priority' and 'Accept-Resource-Priority'. The 'Resource-Priority' header field MAY be used by SIP user agents, including Public Switched Telephone Network (PSTN) gateways and terminals, and SIP proxy servers to influence their treatment of SIP requests, including the priority afforded to PSTN calls. For PSTN gateways, the behavior translates into analogous schemes in the PSTN, for example, the ITU Recommendation Q.735.3 [Q.735.3] prioritization mechanism, in both the PSTN-to-IP and IP-to-PSTN directions. ITU Recommendation I.255.3 [I.255.3] is another example.

A SIP request with a 'Resource-Priority' indication can be treated differently in these situations:

1. The request can be given elevated priority for access to PSTN gateway resources, such as trunk circuits.

2. The request can interrupt lower-priority requests at a user terminal, such as an IP phone.

3. The request can carry information from one multi-level priority domain in the telephone network (e.g., using the facilities of Q.735.3 [Q.735.3]) to another, without the SIP proxies themselves inspecting or modifying the header field.

4. In SIP proxies and back-to-back user agents, requests of higher priorities may displace existing signaling requests or bypass PSTN gateway capacity limits in effect for lower priorities.

This header field is related to, but differs in semantics from, the 'Priority' header field ([RFC3261], Section 20.26). The 'Priority' header field describes the importance that the SIP request should have for the receiving human or its agent. For example, that header may be factored into decisions about call routing to mobile devices and assistants and about call acceptance when the call destination is busy. The 'Priority' header field does not affect the usage of PSTN gateway or proxy resources, for example. In addition, any User Agent Client (UAC) can assert any 'Priority' value, and usage of 'Resource-Priority' header field values is subject to authorization.

While the 'Resource-Priority' header field does not directly influence the forwarding behavior of IP routers or the use of communications resources such as packet forwarding priority, procedures for using this header field to cause such influence may be defined in other documents.

Existing implementations of RFC 3261 that do not participate in the resource priority mechanism follow the normal rules of RFC 3261, Section 8.2.2: "If a UAS does not understand a header field in a request (that is, the header field is not defined in this specification or in any supported extension), the server MUST ignore that header field and continue processing the message". Thus, the use of this mechanism is wholly invisible to existing implementations unless the request includes the Require header field with the resource-priority option tag.

The mechanism described here can be used for emergency preparedness in emergency telecommunications systems, but is only a small part of an emergency preparedness network and is not restricted to such use.

The mechanism aims to satisfy the requirements in [RFC3487]. It is structured so that it works in all SIP and Real-Time Transport Protocol (RTP) [RFC3550] transparent networks, defined in [RFC3487]. In such networks, all network elements and SIP proxies let valid SIP requests pass through unchanged. This is important since it is likely that this mechanism will often be deployed in networks where the edge networks are unaware of the resource priority mechanism and provide no special privileges to such requests. The request then reaches a PSTN gateway or set of SIP elements that are aware of the mechanism.

For conciseness, we refer to SIP proxies and user agents (UAs) that act on the 'Resource-Priority' header field as RP actors.

It is likely to be common that the same SIP element will handle requests that bear the 'Resource-Priority' header fields and those that do not.

Government entities and standardization bodies have developed several different priority schemes for their networks. Users would like to be able to obtain authorized priority handling in several of these networks, without changing SIP clients. Also, a single call may traverse SIP elements that are run by different administrations and subject to different priority mechanisms. Since there is no global ordering among those priorities, we allow each request to contain more than one priority value drawn from these different priority lists, called a namespace in this document. Typically, each SIP element only supports one such namespace, but we discuss what happens if an element needs to support multiple namespaces in Section 8.

Since gaining prioritized access to resources offers opportunities to deny service to others, it is expected that all such prioritized calls are subject to authentication and authorization, using standard SIP security (Section 11) or other appropriate mechanisms.

The remainder of this document is structured as follows. After defining terminology in Section 2, we define the syntax for the two new SIP header fields in Section 3 and then describe protocol behavior in Section 4. The two principal mechanisms for differentiated treatment of SIP requests (namely, preemption and queueing) are described in Section 4.5. Error conditions are covered in Section 4.6. Sections 4.7.1 through 4.7.3 detail the behavior of specific SIP elements. Third-party authentication is briefly summarized in Section 5. Section 6 describes how this feature affects existing systems that do not support it.

Since calls may traverse multiple administrative domains with different namespaces or multiple elements with the same namespace, it is strongly suggested that all such domains and elements apply the same algorithms for the same namespace, as otherwise the end-to-end experience of privileged users may be compromised.

Protocol examples are given in Section 7. Section 8 discusses what happens if a request contains multiple namespaces or an element can handle more than one namespace. Section 9 enumerates the information that namespace registrations need to provide. Section 10 defines the properties of five namespaces that are registered through this document. Security issues are considered in Section 11, but this document does not define new security mechanisms. Section 12 discusses IANA considerations and registers parameters related to this document.

2. Terminology

In this document, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in BCP 14, RFC 2119 [RFC2119], and indicate requirement levels for compliant implementations.

3. The Resource-Priority and Accept-Resource-Priority SIP Header Fields

This section defines the 'Resource-Priority' and 'Accept-Resource-Priority' SIP header field syntax. Behavior is described in Section 4.

3.1. The 'Resource-Priority' Header Field

The 'Resource-Priority' request header field marks a SIP request as desiring prioritized access to resources, as described in the introduction.

There is no protocol requirement that all requests within a SIP dialog or session use the 'Resource-Priority' header field. Local administrative policy MAY mandate the inclusion of the 'Resource-Priority' header field in all requests. Implementations of this specification MUST allow inclusion to be either by explicit user request or automatic for all requests.

The syntax of the 'Resource-Priority' header field is described below. The "token-nodot" production is copied from [RFC3265].

```
Resource-Priority  = "Resource-Priority" HCOLON
                     r-value *(COMMA r-value)
r-value            = namespace "." r-priority
namespace          = token-nodot
r-priority         = token-nodot
token-nodot        = 1*( alphanum / "-"  / "!" / "%" / "*"
                            / "_" / "+" / "`" / "'" / "~" )
```

An example 'Resource-Priority' header field is shown below:

```
Resource-Priority: dsn.flash
```

The 'r-value' parameter in the 'Resource-Priority' header field indicates the resource priority desired by the request originator. Each resource value (r-value) is formatted as 'namespace' '.' 'priority value'. The value is drawn from the namespace identified by the 'namespace' token. Namespaces and priorities are case-insensitive ASCII tokens that do not contain periods. Thus, "dsn.flash" and "DSN.Flash", for example, are equivalent. Each namespace has at least one priority value. Namespaces and priority values within each namespace MUST be registered with IANA (Section 12). Initial namespace registrations are described in Section 12.5.

Since a request may traverse multiple administrative domains with multiple different namespaces, it is necessary to be able to enumerate several different namespaces within the same message. However, a particular namespace MUST NOT appear more than once in the same SIP message. These may be expressed equivalently as either comma-separated lists within a single header field, as multiple header fields, or as some combination. The ordering of 'r-values' within the header field has no significance. Thus, for example, the following three header snippets are equivalent:

```
Resource-Priority: dsn.flash, wps.3

Resource-Priority: wps.3, dsn.flash
```

Resource-Priority: wps.3
Resource-Priority: dsn.flash

3.2. The 'Accept-Resource-Priority' Header Field

The 'Accept-Resource-Priority' response header field enumerates the resource values (r-values) a SIP user agent server is willing to process. (This does not imply that a call with such values will find sufficient resources and succeed.) The syntax of the 'Accept-Resource-Priority' header field is as follows:

```
Accept-Resource-Priority = "Accept-Resource-Priority" HCOLON
                           [r-value *(COMMA r-value)]
```

An example is given below:

```
Accept-Resource-Priority: dsn.flash-override,
    dsn.flash, dsn.immediate, dsn.priority, dsn.routine
```

Some administrative domains MAY choose to disable the use of the 'Accept-Resource-Priority' header for revealing too much information about that domain in responses. However, this behavior is NOT RECOMMENDED, as this header field aids in troubleshooting.

3.3. Usage of the 'Resource-Priority' and 'Accept-Resource-Priority' Header Fields

The following table extends the values in Table 2 of RFC 3261 [RFC3261]. (The PRACK method, labeled as PRA, is defined in [RFC3262], the SUBSCRIBE (labeled SUB) and NOTIFY (labeled NOT) methods in [RFC3265], the UPDATE (UPD) method in [RFC3311], the MESSAGE (MSG) method in [RFC3428], the REFER (REF) method in [RFC3515], the INFO (INF) method in [RFC2976], and the PUBLISH (PUB) method in [RFC3903].)

Header field	where	proxy	INV	ACK	CAN	BYE	REG	OPT	PRA
Resource-Priority	R	amdr	o	o	o	o	o	o	o
Accept-Resource-Priority	200	amdr	o	-	o	o	o	o	o
Accept-Resource-Priority	417	amdr	o	-	o	o	o	o	o

Header field	where	proxy	SUB	NOT	UPD	MSG	REF	INF	PUB
Resource-Priority	R	amdr	o	o	o	o	o	o	o
Accept-Resource-Priority	200	amdr	o	o	o	o	o	o	o
Accept-Resource-Priority	417	amdr	o	o	o	o	o	o	o

Other request methods MAY define their own handling rules; unless otherwise specified, recipients MAY ignore these header fields.

3.4. The 'resource-priority' Option Tag

This document also defines the "resource-priority" option tag. The behavior is described in Section 4.3, and the IANA registration is in Section 12.3.

4. Behavior of SIP Elements That Receive Prioritized Requests

4.1. Introduction

All SIP user agents and proxy servers that support this specification share certain common behavior, which we describe below in Section 4.2. The behavior when a 'resource-priority' option tag is encountered in a 'Require' header field is described in Section 4.3. Section 4.4 describes the treatment of OPTIONS requests. The two fundamental resource contention resolution mechanisms, preemption and queueing, are described in Section 4.5. Section 4.6 explains what happens when requests fail. Behavior specific to user agent clients, servers, and proxy servers is covered in Section 4.7.

4.2. General Rules

The 'Resource-Priority' header field is potentially applicable to all SIP request messages. At a minimum, implementations of the following request types MUST support the Resource-Priority header to be in compliance with this specification:

- INVITE [RFC3261]
- ACK [RFC3261]
- PRACK [RFC3262]
- UPDATE [RFC3311]
- REFER [RFC3515]

Implementations SHOULD support the 'Resource-Priority' header field in the following request types:

- MESSAGE [RFC3428]
- SUBSCRIBE [RFC3265]
- NOTIFY [RFC3265]

Note that this does not imply that all implementations have to support all request methods listed.

If a SIP element receives the 'Resource-Priority' header field in a request other than those listed above, the header MAY be ignored, according to the rules of [RFC3261].

In short, an RP actor performs the following steps when receiving a prioritized request. Error behavior is described in Section 4.6.

1. If the RP actor recognizes none of the name spaces, it treats the request as if it had no 'Resource-Priority' header field.

2. It ascertains that the request is authorized according to local policy to use the priority levels indicated. If the request is not authorized, it rejects it. Examples of authorization policies are discussed in Security Considerations (Section 11).

3. If the request is authorized and resources are available (no congestion), it serves the request as usual. If the request is authorized but resources are not available (congestion), it either preempts other current sessions or inserts the request into a priority queue, as described in Section 4.5.

4.3. Usage of Require Header with Resource-Priority

Following standard SIP behavior, if a SIP request contains the 'Require' header field with the 'resource-priority' option tag, a SIP user agent MUST respond with a 420 (Bad Extension) if it does not support the SIP extensions described in this document. It then lists "resource-priority" in the 'Unsupported' header field included in the response.

The use of the 'resource-priority' option tag in 'Proxy-Require' header field is NOT RECOMMENDED.

4.4. OPTIONS Request with Resource-Priority

An OPTIONS request can be used to determine if an element supports the mechanism. A compliant implementation SHOULD return an 'Accept-Resource-Priority' header field in OPTIONS responses enumerating all valid resource values, but an RP actor MAY be configured not to return such values or only to return them to authorized requestors.

Following standard SIP behavior, OPTIONS responses MUST include the 'Supported' header field that includes the 'resource-priority' option tag.

According to RFC 3261, Section 11, proxies that receive a request with a 'Max-Forwards' header field value of zero MAY answer the OPTIONS request, allowing a UAC to discover the capabilities of both proxy and user agent servers.

4.5. Approaches for Preferential Treatment of Requests

SIP elements may use the resource priority mechanism to modify a variety of behaviors, such as routing requests, authentication requirements, override of network capacity controls, or logging. The resource priority mechanism may influence the treatment of the request itself, the marking of outbound PSTN calls at a gateway, or of the session created by the request. (Here, we use the terms session and call interchangeably, both implying a continuous data stream between two or more parties. Sessions are established by SIP dialogs.)

Below, we define two common algorithms, namely, preemption and priority queueing. Preemption applies only to sessions created by SIP requests, while both sessions and request handling can be subject to priority queueing. Both algorithms can sometimes be combined in the same element, although none of the namespaces described in this document do this. Algorithms can be defined for each namespace or, in some cases, can be specific to an administrative domain. Other behavior, such as request routing or network management controls, is not defined by this specification.

Naturally, only SIP elements that understand this mechanism and the namespace and resource value perform these algorithms. Section 4.6.2 discusses what happens if an RP actor does not understand priority values contained in a request.

4.5.1. Preemption

An RP actor following a preemption policy may disrupt an existing session to make room for a higher-priority incoming session. Since sessions may require different amounts of bandwidth or a different number of circuits, a single higher-priority session may displace more than one lower-priority session. Unless otherwise noted, requests do not preempt other requests of equal priority. As noted above, the processing of SIP requests itself is not preempted. Thus, since proxies do not manage sessions, they do not perform preemption.

[RFC4411] contains more details and examples of this behavior.

UAS behavior for preemption is discussed in Section 4.7.2.1.

4.5.2. Priority Queueing

In a priority queueing policy, requests that find no available resources are queued to the queue assigned to the priority value. Unless otherwise specified, requests are queued in first-come, first-served order. Each priority value may have its own queue, or several priority values may share a single queue. If a resource becomes available, the RP actor selects the request from the highest-priority non-empty queue according to the queue service policy. For first-come, first-served policies, the request from that queue that has been waiting the longest is served. Each queue can hold a finite number of pending requests. If the per-priority-value queue for a newly arriving request is full, the request is rejected immediately, with the status codes specified in Section 4.6.5 and Section 4.6.6. In addition, a priority queueing policy MAY impose a waiting time limit for each priority class, whereby requests that exceed a specified waiting time are ejected from the queue and a 408 (Request Timeout) failure response is returned to the requestor.

Finally, an RP actor MAY impose a global queue size limit summed across all queues and drop waiting lower-priority requests with a 408 (Request Timeout) failure response. This does not imply preemption, since the session has not been established yet.

UAS behavior for queueing is discussed in Section 4.7.2.2.

4.6. Error Conditions

4.6.1. Introduction

In this section, we describe the error behavior that is shared among multiple types of RP actors (including various instances of UAS such as trunk gateways, line gateways, and IP phones) and proxies.

A request containing a resource priority indication can fail for four reasons:

- o the RP actor does not understand the priority value (Section 4.6.2),
- o the requestor is not authenticated (Section 4.6.3),
- o an authenticated requestor is not authorized to make such a request (Section 4.6.4), or
- o there are insufficient resources for an authorized request (Section 4.6.5).

We treat these error cases in the order that they typically arise in the processing of requests with Resource-Priority headers. However, this order is not mandated. For example, an RP actor that knows that a particular resource value cannot be served or queued MAY, as a matter of local policy, forgo authorization, since it would only add processing load without changing the outcome.

4.6.2. No Known Namespace or Priority Value

If an RP actor does not understand any of the resource values in the request, the treatment depends on the presence of the 'Require' 'resource-priority' option tag:

1. Without the option tag, the RP actor treats the request as if it contained no 'Resource-Priority' header field and processes it with default priority. Resource values that are not understood MUST NOT be modified or deleted.

2. With the option tag, it MUST reject the request with a 417 (Unknown Resource-Priority) response code.

Making case (1) the default is necessary since otherwise there would be no way to successfully complete any calls in the case where a proxy on the way to the UAS shares no common namespaces with the UAC, but the UAC and UAS do have such a namespace in common.

In general, as noted, a SIP request can contain more than one 'Resource-Priority' header field. This is necessary if a request needs to traverse different administrative domains, each with its own set of valid resource values. For example, the ETS namespace might be enabled for United States government networks that also support the DSN and/or DRSN namespaces for most individuals in those domains.

A 417 (Unknown Resource-Priority) response MAY, according to local policy, include an 'Accept-Resource-Priority' header field enumerating the acceptable resource values.

4.6.3. Authentication Failure

If the request is not authenticated, a 401 (Unauthorized) or 407 (Proxy Authentication Required) response is returned in order to allow the requestor to insert appropriate credentials.

4.6.4. Authorization Failure

If the RP actor receives an authenticated request with a namespace and priority value it recognizes but the originator is not authorized for that level of service, the element MUST return a 403 (Forbidden) response.

4.6.5. Insufficient Resources

Insufficient resource conditions can occur on proxy servers and user agent servers, typically trunk gateways, if an RP actor receives an authorized request, has insufficient resources, and the request neither preempts another session nor is queued. A request can fail because the RP actor has either insufficient processing capacity to handle the SIP request or insufficient bandwidth or trunk capacity to establish the requested session for session-creating SIP requests.

If the request fails because the RP actor cannot handle the signaling load, the RP actor responds with 503 (Service Unavailable).

If there is not enough bandwidth, or if there is an insufficient number of trunks, a 488 (Not Acceptable Here) response indicates that the RP actor is rejecting the request due to media path availability, such as insufficient gateway resources. In that case, [RFC3261] advises that a 488 response SHOULD include a 'Warning' header field with a reason for the rejection; warning code 370 (Insufficient Bandwidth) is typical.

For systems implementing queueing, if the request is queued, the UAS will return 408 (Request Timeout) if the request exceeds the maximum configured waiting time in the queue.

4.6.6. Busy

Resource contention also occurs when a call request arrives at a UAS that is unable to accept another call, because the UAS either has just one line appearance or has active calls on all line appearances. If the call request indicates an equal or lower priority value when compared to all active calls present on the UAS, the UAS returns a 486 (Busy here) response.

If the request is queued instead, the UAS will return a 408 (Request Timeout) if the request exceeds the maximum configured waiting time in the device queue.

If a proxy gets 486 (Busy Here) responses on all branches, it can then return a 600 (Busy Everywhere) response to the caller.

4.7. Element-Specific Behaviors

4.7.1. User Agent Client Behavior

SIP UACs supporting this specification MUST be able to generate the 'Resource-Priority' header field for requests that require elevated resource access priority. As stated previously, the UAC SHOULD be able to generate more than one resource value in a single SIP request.

Upon receiving a 417 (Unknown Resource-Priority) response, the UAC MAY attempt a subsequent request with the same or different resource value. If available, it SHOULD choose authorized resource values from the set of values returned in the 'Accept-Resource-Priority' header field.

4.7.1.1. User Agent Client Behavior with a Preemption Algorithm

A UAC that requests a priority value that may cause preemption MUST understand a Reason header field in the BYE request explaining why the session was terminated, as discussed in [RFC4411].

4.7.1.2. User Agent Client Behavior with a Queueing Policy

By standard SIP protocol rules, a UAC MUST be prepared to receive a 182 (Queued) response from an RP actor that is currently at capacity, but that has put the original request into a queue. A UAC MAY indicate this queued status to the user by some audio or visual indication to prevent the user from interpreting the call as having failed.

4.7.2. User Agent Server Behavior

The precise effect of the 'Resource-Priority' indication depends on the type of UAS, the namespace, and local policy.

4.7.2.1. User Agent Servers and Preemption Algorithm

A UAS compliant with this specification MUST terminate a session established with a valid namespace and lower-priority value in favor of a new session set up with a valid namespace and higher relative priority value, unless local policy has some form of call-waiting capability enabled. If a session is terminated, the BYE method is used with a 'Reason' header field indicating why and where the preemption took place.

Implementors have a number of choices in how to implement preemption at IP phones with multiple line presences, i.e., with devices that can handle multiple simultaneous sessions. Naturally, if that device has exhausted the number of simultaneous sessions, one of the sessions needs to be replaced. If the device has spare sessions, an implementation MAY choose to alert the callee to the arrival of a higher-priority call. Details may also be set by local or namespace policy.

[RFC4411] provides additional information in the case of purposeful or administrative termination of a session by including the Reason header in the BYE message that states why the BYE was sent (in this case, a preemption event). The mechanisms in that document allow indication of where the termination occurred ('at the UA', 'within a reservation', 'at a IP/PSTN gateway') and include call flow examples of each reason.

4.7.2.2. User Agent Servers and Queue-Based Policy

A UAS compliant with this specification SHOULD generate a 182 (Queued) response if that element's resources are busy, until it is able to handle the request and provide a final response. The frequency of such provisional messages is governed by [RFC3261].

4.7.3. Proxy Behavior

SIP proxies MAY ignore the 'Resource-Priority' header field. SIP proxies MAY reject any unauthenticated request bearing that header field.

When the 'Require' header field is included in a message, it ensures that in parallel forking, only branches that support the resource-priority mechanism succeed.

If S/MIME encapsulation is used according to Section 23 of RFC 3261, special considerations apply. As tabulated in Section 3.3, the 'Resource-Priority' header field can be modified by proxies and thus is exempted from the integrity checking described in Section 23.4.1.1 of RFC 3261. Since it may need to be inspected or modified by proxies, the header field MUST also be placed in the "outer" message if the UAC would like proxy servers to be able to act on the header information. Similar considerations apply if parts of the message are integrity protected or encrypted as described in [RFC3420].

If S/MIME is not used, or if the 'Resource-Priority' header field is in the "outer" header, SIP proxies MAY downgrade or upgrade the 'Resource-Priority' of a request or insert a new 'Resource-Priority' header if allowed by local policy.

If a stateful proxy has authorized a particular resource priority level, and if it offers differentiated treatment to responses containing resource priority levels, the proxy SHOULD ignore any higher value contained in responses, to prevent colluding user agents from artificially raising the priority level.

A SIP proxy MAY use the 'Resource-Priority' indication in its routing decisions, e.g., to retarget to a SIP node or SIP URI that is reserved for a particular resource priority.

There are no special considerations for proxies when forking requests containing a resource priority indication.

Otherwise, the proxy behavior is the same as for user agent servers described in Section 4.7.2.

5. Third-Party Authentication

In some cases, the RP actor may not be able to authenticate the requestor or determine whether an authenticated user is authorized to make such a request. In these circumstances, the SIP entity may avail itself of general SIP mechanisms that are not specific to this application. The authenticated identity management mechanism [RFC3893] allows a third party to verify the identity of the requestor and to certify this towards an RP actor. In networks with mutual trust, the SIP-asserted identity mechanism [RFC3325] can help the RP actor determine the identity of the requestor.

6. Backwards Compatibility

The resource priority mechanism described in this document is fully backwards compatible with SIP systems following [RFC3261]. Systems that do not understand the mechanism can only deliver standard, not elevated, service priority. User agent servers and proxies can ignore any 'Resource-Priority' header field just like any other unknown header field and then treat the request like any other request. Naturally, the request may still succeed.

7. Examples

The SDP message body and the BYE and ACK exchanges are the same as in RFC 3665 [RFC3665] and are omitted for brevity.

7.1. Simple Call

```
   User A                  User B
     |                        |
     |       INVITE F1        |
     |----------------------->|
     |    180 Ringing F2      |
     |<-----------------------|
     |                        |
     |       200 OK F3        |
     |<-----------------------|
     |         ACK F4         |
     |----------------------->|
     |   Both Way RTP Media   |
     |<======================>|
     |                        |
```

In this scenario, User A completes a call to User B directly. The call from A to B is marked with a resource priority indication.

F1 INVITE User A -> User B

```
INVITE sip:UserB@biloxi.example.com SIP/2.0
Via: SIP/2.0/TCP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
Max-Forwards: 70
From: BigGuy <sip:UserA@atlanta.example.com>;tag=9fxced76sl
To: LittleGuy <sip:UserB@biloxi.example.com>
Call-ID: 3848276298220188511@atlanta.example.com
CSeq: 1 INVITE
Resource-Priority: dsn.flash
Contact: <sip:UserA@client.atlanta.example.com;transport=tcp>
Content-Type: application/sdp
Content-Length: ...

...
```

F2 180 Ringing User B -> User A

```
SIP/2.0 180 Ringing
Via: SIP/2.0/TCP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
  ;received=192.0.2.101
From: BigGuy <sip:UserA@atlanta.example.com>;tag=9fxced76sl
To: LittleGuy <sip:UserB@biloxi.example.com>;tag=8321234356
Call-ID: 3848276298220188511@atlanta.example.com
CSeq: 1 INVITE
Contact: <sip:UserB@client.biloxi.example.com;transport=tcp>
Content-Length: 0
```

F3 200 OK User B -> User A

```
SIP/2.0 200 OK
Via: SIP/2.0/TCP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
  ;received=192.0.2.101
From: BigGuy <sip:UserA@atlanta.example.com>;tag=9fxced76sl
To: LittleGuy <sip:UserB@biloxi.example.com>;tag=8321234356
Call-ID: 3848276298220188511@atlanta.example.com
CSeq: 1 INVITE
Contact: <sip:UserB@client.biloxi.example.com;transport=tcp>
Content-Type: application/sdp
Content-Length: ...

...
```

7.2. Receiver Does Not Understand Namespace

In this example, the receiving UA does not understand the "dsn" namespace and thus returns a 417 (Unknown Resource-Priority) status code. We omit the message details for messages F5 through F7, since they are essentially the same as in the first example.

```
User A                  User B
  |                        |
  |       INVITE F1        |
  |----------------------->|
  | 417 R-P failed F2      |
  |<-----------------------|
  |         ACK F3         |
  |----------------------->|
  |                        |
  |       INVITE F4        |
  |----------------------->|
  |    180 Ringing F5      |
  |<-----------------------|
  |       200 OK F6        |
  |<-----------------------|
  |         ACK F7         |
  |----------------------->|
  |                        |
  |   Both Way RTP Media   |
  |<======================>|
```

F1 INVITE User A -> User B

```
INVITE sip:UserB@biloxi.example.com SIP/2.0
Via: SIP/2.0/TCP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
Max-Forwards: 70
From: BigGuy <sip:UserA@atlanta.example.com>;tag=9fxced76sl
To: LittleGuy <sip:UserB@biloxi.example.com>
Call-ID: 3848276298220188511@atlanta.example.com
CSeq: 1 INVITE
Require: resource-priority
Resource-Priority: dsn.flash
Contact: <sip:UserA@client.atlanta.example.com;transport=tcp>

Content-Type: application/sdp
Content-Length: ...

...
```

F2 417 Resource-Priority failed User B -> User A

```
SIP/2.0 417 Unknown Resource-Priority
Via: SIP/2.0/TCP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
  ;received=192.0.2.101
From: BigGuy <sip:UserA@atlanta.example.com>;tag=9fxced76sl
To: LittleGuy <sip:UserB@biloxi.example.com>;tag=8321234356
Call-ID: 3848276298220188511@atlanta.example.com
CSeq: 1 INVITE
Accept-Resource-Priority: q735.0, q735.1, q735.2, q735.3, q735.4
Contact: <sip:UserB@client.biloxi.example.com;transport=tcp>
Content-Type: application/sdp
Content-Length: 0
```

F3 ACK User A -> User B

```
ACK sip:UserB@biloxi.example.com SIP/2.0
Via: SIP/2.0/TCP client.atlanta.example.com:5060;branch=z9hG4bK74bd5
Max-Forwards: 70
From: BigGuy <sip:UserA@atlanta.example.com>;tag=9fxced76sl
To: LittleGuy <sip:UserB@biloxi.example.com>;tag=8321234356
Call-ID: 3848276298220188511@atlanta.example.com
CSeq: 1 ACK
Content-Length: 0
```

F4 INVITE User A -> User B

```
INVITE sip:UserB@biloxi.example.com SIP/2.0
Via: SIP/2.0/TCP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
Max-Forwards: 70
From: BigGuy <sip:UserA@atlanta.example.com>;tag=9fxced76sl
To: LittleGuy <sip:UserB@biloxi.example.com>
Call-ID: 3848276298220188511@atlanta.example.com
CSeq: 2 INVITE
Require: resource-priority
Resource-Priority: q735.3
Contact: <sip:UserA@client.atlanta.example.com;transport=tcp>

Content-Type: application/sdp
Content-Length: ...
...
```

8. Handling Multiple Concurrent Namespaces

8.1. General Rules

A single SIP request MAY contain resource values from multiple namespaces. As noted earlier, an RP actor disregards all namespaces it does not recognize. This specification only addresses the case where an RP actor then selects one of the remaining resource values for processing, usually choosing the one with the highest relative priority.

If an RP actor understands multiple namespaces, it MUST create a local total ordering across all resource values from these namespaces, maintaining the relative ordering within each namespace. It is RECOMMENDED that the same ordering be used across an administrative domain. However, there is no requirement that such ordering be the same across all administrative domains.

8.2. Examples of Valid Orderings

Below are a set of examples of an RP actor that supports two namespaces, foo and bar. Foo’s priority-values are 3 (highest), then 2, and then 1 (lowest), and bar’s priority-values are C (highest), then B, and then A (lowest).

Below are five lists of acceptable priority orders the SIP element may use:

```
    Foo.3        Foo.3       Bar.C    (highest priority)
    Foo.2        Bar.C       Foo.3
    Foo.1   or   Foo.2   or  Foo.2
    Bar.C        Bar.B       Foo.1
    Bar.B        Foo.1       Bar.B
    Bar.A        Bar.A       Bar.A    (lowest priority)


           Bar.C       (highest priority)
        Foo.3  Bar.B   (both treated with equal priority (FIFO))
 or     Foo.2  Bar.A   (both treated with equal priority (FIFO))
           Foo.1       (lowest priority)


        Bar.C     (highest priority)
        Foo.3
 or     Foo.2
        Foo.1     (lowest priority)
```

In the last example above, Bar.A and Bar.B are ignored.

8.3. Examples of Invalid Orderings

Based on the priority order of the namespaces above, the following combinations are examples of orderings that are NOT acceptable and MUST NOT be configurable:

```
       Example 1    Example 2   Example 3
       ---------    ---------   ---------
         Foo.3        Foo.3       Bar.C
         Foo.2        Bar.A       Foo.1
         Foo.1   or   Foo.2   or  Foo.3
         Bar.C        Bar.B       Foo.2
         Bar.A        Foo.1       Bar.A
         Bar.B        Bar.C       Bar.B

              Example 4
              ---------
                Bar.C
             Foo.1  Bar.B
      or     Foo.3  Bar.A
                Foo.2
```

These examples are invalid since the following global orderings are not consistent with the namespace-internal order:

- o In Example 1, Bar.A is ordered higher than Bar.B.
- o In Example 2, Bar.A is ordered higher than Bar.B and Bar.C.
- o In Example 3, Foo.1 is ordered higher than Foo.2 and Foo.3.
- o In Example 4, Foo.1 is ordered higher than Foo.3 and Foo.2.

9. Registering Namespaces

Organizations considering the use of the Resource-Priority header field should investigate whether an existing combination of namespace and priority-values meets their needs. For example, emergency first responders around the world are discussing utilizing this mechanism for preferential treatment in future networks. Jurisdictions SHOULD attempt to reuse existing IANA registered namespaces where possible, as a goal of this document is not to have unique namespaces per jurisdiction serving the same purpose, with the same usage of priority levels. This will greatly increase interoperability and reduce development time, and probably reduce future confusion if there is ever a need to map one namespace to another in an interworking function.

Below, we describe the steps necessary to register a new namespace.

A new namespace MUST be defined in a Standards Track RFC, following the 'Standards Action' policy in [RFC2434], and MUST include the following facets:

- o It must define the namespace label, a unique namespace label within the IANA registry for the SIP Resource-Priority header field.
- o It must enumerate the priority levels (i.e., 'r-priority' values) the namespace is using. Note that only finite lists are permissible, not unconstrained integers or tokens, for example.
- o The priority algorithm (Section 4.5), identifying whether the namespace is to be used with priority queueing ("queue") or preemption ("preemption"). If queueing is used, the namespace MAY indicate whether normal-priority requests are queued. If there is a new "intended algorithm" other than preemption or priority queueing, the algorithm must be described, taking into account all RP actors (UAC, UAS, proxies).

- o A namespace may either reference an existing list of priority values or define a new finite list of priority values in relative priority order for IANA registration within the sip-parameters Resource-Priority priority-values registry. New priority-values SHOULD NOT be added to a previously IANA-registered list associated with a particular namespace, as this may cause interoperability problems. Unless otherwise specified, it is assumed that all priority values confer higher priority than requests without a priority value.

- o Any new SIP response codes unique to this new namespace need to be explained and registered.

- o The reference document must specify and describe any new Warning header field warn-codes (RFC 3261, Section 27.2).

- o The document needs to specify a new row for the following table that summarizes the features of the namespace and is included into IANA Resource-Priority Namespace registration:

Namespace	Levels	Intended algorithm	New warn-code	New resp. code	Reference
<label>	<# of levels>	<preemption or queue>	<new warn code>	<new resp. code>	<RFC>

If information on new response codes, rejection codes, or error behaviors is omitted, it is to be assumed that the namespace defines no new parameters or behaviors.

10. Namespace Definitions

10.1. Introduction

This specification defines five unique namespaces below: DSN, DRSN, Q735, ETS, and WPS, constituting their registration with IANA. Each IANA registration contains the facets defined in Section 9. For recognizability, we label the namespaces in capital letters, but note that namespace names are case insensitive and are customarily rendered as lowercase in protocol requests.

10.2. The "DSN" Namespace

The DSN namespace comes from the name of a US government network called "The Defense Switched Network".

The DSN namespace has a finite list of relative priority-values, listed below from lowest priority to highest priority:

```
(lowest)  dsn.routine
          dsn.priority
          dsn.immediate
          dsn.flash
(highest) dsn.flash-override
```

The DSN namespace uses the preemption algorithm (Section 4.5.1).

10.3. The "DRSN" Namespace

The DRSN namespace comes from the name of a US government network, called "The Defense RED Switched Network".

The DRSN namespace defines the following resource values, listed from lowest priority to highest priority:

```
(lowest)  drsn.routine
          drsn.priority
          drsn.immediate
          drsn.flash
          drsn.flash-override
(highest) drsn.flash-override-override
```

The DRSN namespace uses the preemption algorithm (Section 4.5.1).

The DRSN namespace differs in one algorithmic aspect from the DSN and Q735 namespaces. The behavior for the ’flash-override-override’ priority value differs from the other values. Normally, requests do not preempt those of equal priority, but a newly arriving ’flash-override-override’ request will displace another one of equal priority if there are insufficient resources. This can also be expressed as saying that ’flash-override-override’ requests defend themselves as ’flash-override’ only.

10.4. The "Q735" Namespace

Q.735.3 [Q.735.3] was created to be a commercial version of the operationally equivalent DSN specification for Multi-Level Precedence and Preemption (MLPP). The Q735 namespace is defined here in the same manner.

The Q735 namespace defines the following resource values, listed from lowest priority to highest priority:

```
(lowest)  q735.4
          q735.3
          q735.2
          q735.1
(highest) q735.0
```

The Q735 namespace operates according to the preemption (Section 4.5.1) algorithm.

10.5. The "ETS" Namespace

The ETS namespace derives its name indirectly from the name of the US government telecommunications service, called "Government Emergency Telecommunications Service" (or GETS), though the organization responsible for the GETS service chose the acronym "ETS" for its GETS over IP service, which stands for "Emergency Telecommunications Service".

The ETS namespace defines the following resource values, listed from lowest priority to highest priority:

```
(lowest)  ets.4
          ets.3
          ets.2
          ets.1
(highest) ets.0
```

The ETS namespace operates according to the priority queueing algorithm (Section 4.5.2).

10.6. The "WPS" Namespace

The WPS namespace derives its name from the "Wireless Priority Service", defined in GSM and other wireless technologies.

The WPS namespace defines the following resource values, listed from lowest priority to highest priority:

```
(lowest)  wps.4
          wps.3
          wps.2
          wps.1
(highest) wps.0
```

The WPS namespace operates according to the priority queueing algorithm (Section 4.5.2).

11. Security Considerations

11.1. General Remarks

Any resource priority mechanism can be abused to obtain resources and thus deny service to other users. An adversary may be able to take over a particular PSTN gateway, cause additional congestion during emergencies affecting the PSTN, or deny service to legitimate users. In SIP end systems, such as IP phones, this mechanism could inappropriately terminate existing sessions and calls.

Thus, while the indication itself does not have to provide separate authentication, SIP requests containing this header are very likely to have higher authentication requirements than those without.

These authentication and authorization requirements extend to users within the administrative domain, as later interconnection with other administrative domains may invalidate earlier assumptions on the trustworthiness of users.

Below, we describe authentication and authorization aspects, confidentiality and privacy requirements, protection against denial-of-service attacks, and anonymity requirements. Naturally, the general discussion in RFC 3261 [RFC3261] applies.

All user agents and proxy servers that support this extension MUST implement SIP over TLS [RFC3546], the 'sips' URI scheme as described in Section 26.2 of RFC 3261, and Digest Authentication [RFC2617] as described in Section 22 of RFC 3261. In addition, user agents that support this extension SHOULD also implement S/MIME [RFC3851] as described in Section 23 of RFC 3261 to allow for signing and verification of signatures over requests that use this extension.

11.2. Authentication and Authorization

Prioritized access to network and end-system resources imposes particularly stringent requirements on authentication and authorization mechanisms, since access to prioritized resources may impact overall system stability and performance and not just result in theft of, say, a single phone call.

Under certain emergency conditions, the network infrastructure, including its authentication and authorization mechanism, may be under attack.

Given the urgency during emergency events, normal statistical fraud detection may be less effective, thus placing a premium on reliable authentication.

Common requirements for authentication mechanisms apply, such as resistance to replay, cut-and-paste, and bid-down attacks.

Authentication MAY be SIP based or use other mechanisms. Use of Digest authentication and/or S/MIME is RECOMMENDED for UAS authentication. Digest authentication requires that the parties share a common secret, thus limiting its use across administrative domains. SIP systems employing resource priority SHOULD implement S/MIME at least for integrity, as described in Section 23 of [RFC3261]. However, in some environments, receipt of asserted identity [RFC3325] from a trusted entity may be sufficient authorization. Section 5 describes third-party authentication.

Trait-based authorization [TRAIT] "entails an assertion by a authorization service of attributes associated with an identity" and may be appropriate for this application. With trait-based authorization, a network element can directly determine, by inspecting the certificate, that a request is authorized to obtain a particular type of service, without having to consult a mapping mechanism that converts user identities to authorizations.

Authorization may be based on factors besides the identity of the caller, such as the requested destination. Namespaces MAY also impose particular authentication or authorization considerations that are stricter than the baseline described here.

11.3. Confidentiality and Integrity

Calls that use elevated resource priority levels provided by the 'Resource-Priority' header field are likely to be sensitive and often need to be protected from intercept and alteration. In particular, requirements for protecting the confidentiality of communications relationships may be higher than those for normal commercial service. For SIP, the 'To', 'From', 'Organization', and 'Subject' header fields are examples of particularly sensitive information. Systems MUST implement encryption at the transport level using TLS and MAY implement other transport-layer or network-layer security mechanisms. UACs SHOULD use the "sips" URI to request a secure transport association to the destination.

The 'Resource-Priority' header field can be carried in the SIP message header or can be encapsulated in a message fragment carried in the SIP message body [RFC3420]. To be considered valid authentication for the purposes of this specification, S/MIME-signed

SIP messages or fragments MUST contain, at a minimum, the Date, To, From, Call-ID, and Resource-Priority header fields. Encapsulation in S/MIME body parts allows the user to protect this header field against inspection or modification by proxies. However, in many cases, proxies will need to authenticate and authorize the request, so encapsulation would be undesirable.

Removal of a Resource-Priority header field or downgrading its priority value affords no additional opportunities to an adversary, since that man-in-the-middle could simply drop or otherwise invalidate the SIP request and thus prevent call completion.

Only SIP elements within the same administrative trust domain employing a secure channel between their SIP elements will trust a Resource-Priority header field that is not appropriately signed. Others will need to authenticate the request independently. Thus, insertion of a Resource-Priority header field or upgrading the priority value has no further security implications except causing a request to fail (see discussion in the previous paragraph).

11.4. Anonymity

Some users may wish to remain anonymous to the request destination. Anonymity for requests with resource priority is no different from that for any other authenticated SIP request. For the reasons noted earlier, users have to authenticate themselves towards the SIP elements carrying the request where they desire resource priority treatment. The authentication may be based on capabilities and noms, not necessarily their civil name. Clearly, they may remain anonymous towards the request destination, using the network-asserted identity and general privacy mechanism described in [RFC3323].

11.5. Denial-of-Service Attacks

As noted, systems described here are likely to be subject to deliberate denial-of-service (DoS) attacks during certain types of emergencies. DoS attacks may be launched on the network itself as well as on its authentication and authorization mechanism. As noted, systems should minimize the amount of state, computation, and network resources that an unauthorized user can command. The system must not amplify attacks by causing the transmission of more than one packet to a network address whose reachability has not been verified.

12. IANA Considerations

12.1. Introduction

This section defines two new SIP headers (Section 12.2), one SIP option tag (Section 12.3), one new 4xx error code (Section 12.4), a new registry within the sip-parameters section of IANA for Resource-Priority namespaces (Section 12.5), and a new registry within the sip-parameters section of IANA for Resource-Priority and priority-values (Section 12.6).

Additional namespaces and priority values MUST be registered with IANA, as described in Section 9.

The SIP Change Process [RFC3427] establishes a policy for the registration of new SIP extension headers. Resource priority namespaces and priority values have similar interoperability requirements to those of SIP extension headers. Consequently, registration of new resource priority namespaces and priority values requires documentation in an RFC using the extension header approval process specified in RFC 3427.

Registration policies for new namespaces are defined in Section 9.

12.2. IANA Registration of 'Resource-Priority' and 'Accept-Resource-Priority' Header Fields

The following is the registration for the 'Resource-Priority' header field:

RFC number: 4412
Header name: 'Resource-Priority'
Compact form: none

The following is the registration for the 'Accept-Resource-Priority' header field:

RFC number: 4412
Header name: Accept-Resource-Priority
Compact form: none

12.3. IANA Registration for Option Tag resource-priority

RFC number: 4412
Name of option tag: 'resource-priority'
Descriptive text: Indicates or requests support for the resource priority mechanism.

12.4. IANA Registration for Response Code 417

RFC number: 4412
Response code: 417
Default reason phrase: Unknown Resource-Priority

12.5. IANA Resource-Priority Namespace Registration

A new registry ("Resource-Priority Namespaces") in the sip-parameters section of IANA has been created, taking a form similar to this table below:

Namespace	Levels	Intended Algorithm	New warn-code	New resp. code	Reference
dsn	5	preemption	no	no	[RFC4412]
drsn	6	preemption	no	no	[RFC4412]
q735	5	preemption	no	no	[RFC4412]
ets	5	queue	no	no	[RFC4412]
wps	5	queue	no	no	[RFC4412]

Legend

Namespace	The unique string identifying the namespace.
Levels	The number of priority-values within the namespace.
Algorithm	Intended operational behavior of SIP elements implementing this namespace.
New Warn code	New Warning Codes (warn-codes) introduced by this namespace.
New Resp. code	New SIP response codes introduced by this namespace.
Reference	IETF document reference for this namespace.

12.6. IANA Priority-Value Registrations

A new registry ("Resource-Priority Priority-values") in the sip-parameters section of IANA has been created, taking a form similar to this table below:

Namespace: drsn
Reference: RFC 4412
Priority-Values (least to greatest): "routine", "priority",
"immediate", "flash", "flash-override", "flash-override-override"

Namespace: dsn
Reference: RFC 4412
Priority-Values (least to greatest): "routine", "priority",
"immediate", "flash", "flash-override"

Namespace: q735
Reference: RFC 4412
Priority values (least to greatest): "4", "3", "2", "1", "0"

Namespace: ets
Reference: RFC 4412
Priority values (least to greatest): "4", "3", "2", "1", "0"

Namespace: wps
Reference: RFC 4412
Priority values (least to greatest): "4", "3", "2", "1", "0"

13. Acknowledgements

Ben Campbell, Ken Carlberg, Paul Kyzivat, Rohan Mahy, Allison Mankin, Xavier Marjou, Piers O'Hanlon, Mike Pierce, Samir Srivastava, and Dale Worley provided helpful comments.

Dean Willis provided much help with this effort.

Martin Dolly, An Nguyen, and Niranjan Sandesara assisted with the ETS and WPS namespaces.

Janet Gunn helped improve the text on queueing-based priority.

14. References

14.1. Normative References

[I.255.3] International Telecommunications Union, "Integrated Services Digital Network (ISDN) - General Structure and Service Capabilities - Multi-Level Precedence and Preemption", Recommendation I.255.3, July 1990.

[Q.735.3] International Telecommunications Union, "Stage 3 description for community of interest supplementary services using Signalling System No. 7: Multi-level precedence and preemption", Recommendation Q.735.3, March 1993.

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

[RFC2434] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.

[RFC3261] 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.

[RFC3262] Rosenberg, J. and H. Schulzrinne, "Reliability of Provisional Responses in Session Initiation Protocol (SIP)", RFC 3262, June 2002.

[RFC3265] Roach, A.B., "Session Initiation Protocol (SIP)-Specific Event Notification", RFC 3265, June 2002.

[RFC3311] Rosenberg, J., "The Session Initiation Protocol (SIP) UPDATE Method", RFC 3311, October 2002.

[RFC3420] Sparks, R., "Internet Media Type message/sipfrag", RFC 3420, November 2002.

[RFC3428] Campbell, B., Rosenberg, J., Schulzrinne, H., Huitema, C., and D. Gurle, "Session Initiation Protocol (SIP) Extension for Instant Messaging", RFC 3428, December 2002.

[RFC4411] Polk, J., "Extending the Session Initiation Protocol (SIP) Reason Header for Preemption Events", RFC 4411, February 2006.

14.2. Informative References

[RFC2617] Franks, J., Hallam-Baker, P., Hostetler, J., Lawrence, S., Leach, P., Luotonen, A., and L. Stewart, "HTTP Authentication: Basic and Digest Access Authentication", RFC 2617, June 1999.

[RFC2976] Donovan, S., "The SIP INFO Method", RFC 2976, October 2000.

[RFC3323] Peterson, J., "A Privacy Mechanism for the Session Initiation Protocol (SIP)", RFC 3323, November 2002.

[RFC3325] Jennings, C., Peterson, J., and M. Watson, "Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks", RFC 3325, November 2002.

[RFC3427] Mankin, A., Bradner, S., Mahy, R., Willis, D., Ott, J., and B. Rosen, "Change Process for the Session Initiation Protocol (SIP)", BCP 67, RFC 3427, December 2002.

[RFC3487] Schulzrinne, H., "Requirements for Resource Priority Mechanisms for the Session Initiation Protocol (SIP)", RFC 3487, February 2003.

[RFC3515] Sparks, R., "The Session Initiation Protocol (SIP) Refer Method", RFC 3515, April 2003.

[RFC3546] Blake-Wilson, S., Nystrom, M., Hopwood, D., Mikkelsen, J., and T. Wright, "Transport Layer Security (TLS) Extensions", RFC 3546, June 2003.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.

[RFC3665] Johnston, A., Donovan, S., Sparks, R., Cunningham, C., and K. Summers, "Session Initiation Protocol (SIP) Basic Call Flow Examples", BCP 75, RFC 3665, December 2003.

[RFC3851] Ramsdell, B., "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.1 Message Specification", RFC 3851, July 2004.

[RFC3893] Peterson, J., "Session Initiation Protocol (SIP) Authenticated Identity Body (AIB) Format", RFC 3893, September 2004.

[RFC3903] Niemi, A., "Session Initiation Protocol (SIP) Extension for Event State Publication", RFC 3903, October 2004. for Event State Publication", RFC 3903, October 2004.

[TRAIT] Peterson, J., Polk, J., Sicker, D., and H. Tschofenig, "Trait-based Authorization Requirements for the Session Initiation Protocol (SIP)", Work in Progress, February 2005.

Authors' Addresses

Henning Schulzrinne
Columbia University
Department of Computer Science
450 Computer Science Building
New York, NY 10027
US

Phone: +1 212 939 7004
EMail: hgs@cs.columbia.edu
URI: http://www.cs.columbia.edu

James Polk
Cisco Systems
2200 East President George Bush Turnpike
Richardson, TX 75082
US

EMail: jmpolk@cisco.com

Full Copyright Statement

Copyright (C) The Internet Society (2006).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgement

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).