

Sigtran Working Group	N. Stewart	
Internet-Draft	G. Hunt	
Intended status: Standards Track	BT	
Expires: September 3, 2009	D. Chohan	
	Ftel	
	March 02, 2009	

[TOC](#)

IUA Extension for Rate Control Message draft-hunt-sigtran-iaa-rate-message-00.txt

Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on September 3, 2009.

Copyright Notice

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

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

Abstract

This document describes a new message, its associated acknowledgement message, and a new parameter to extend the ISDN Q.921-User Adaptation (IUA) protocol (RFC4233). The protocol extension is to support the use of an Overload Control Agent in a Signaling Gateway (SG). The Overload

Control Agent is able to restrict the admission of new originating ISDN calls (sessions) messages from the ISDN End Point to each Application Server Process (ASP). Both messages defined here contain a single mandatory parameter, the Call (Session) Admission Rate. An ASP is able to use this protocol extension to control the rate of new calls admitted towards that ASP by the Overload Control Agent.

The new message and its acknowledgement message are added to the Application Server Process Traffic Maintenance (ASPTM) message class. As the DPNSS1/DASS2 Extension to IUA (DUA, RFC4129) also uses the ASPTM message class, the IUA protocol extension described in this document also applies to DUA.

For backward compatibility, a Signaling Gateway which does not support the new message is expected to follow standard IUA behaviour by discarding the message, and returning an error code of "Unsupported Message Type" to the sender.

Table of Contents

1.	Introduction
1.1.	Requirement for overload control
1.2.	Scope
1.3.	Analogy with etsi_nr overload control
1.4.	Applicability to DUA
2.	Requirements Notation
3.	Definitions
4.	New Admission Rate and Acknowledgement Messages and New Parameter
4.1.	New Message Types
4.2.	New Parameter
4.3.	ASP Call (Session) Admission Rate (ASPCAR) Message Format
4.4.	ASP Call (Session) Admission Rate Ack (ASPCAR Ack) Message Format
5.	Procedures
5.1.	Basic procedures
5.2.	ASP States and the ASPCAR message
5.3.	Applicability to DUA
5.4.	Procedures for use of the rate parameter acknowledgment
6.	IANA Considerations
7.	Security Considerations
8.	Informative Appendix: Message Sequence Diagrams for Acknowledgements
8.1.	Message sequence for normal transmission
8.2.	Message sequence for a lost message
8.3.	Message sequence for late acknowledgement
8.4.	Message sequence for updated rate
8.5.	Message sequence for lost message and updated rate
8.6.	Message sequence for lost rate update
8.7.	Message sequence for sequencing error

8.8.	Message sequence for sequencing error and lost acknowledgement
8.9.	Message sequence for updated rate and lost acknowledgement
9.	Contributors
10.	References
10.1.	Normative References
10.2.	Informative References
§	Authors' Addresses

1. Introduction

[TOC](#)

This document describes a new message type, its associated acknowledgement message type, a new parameter type, and the associated procedures as an extension to the current IUA protocol [\[RFC4233\]](#) ([Morneault, K., "Integrated Services Digital Network \(ISDN\) Q.921-User Adaptation Layer," January 2006.](#)). The protocol extension supports the use of an Overload Control Agent in the Signaling Gateway.

1.1. Requirement for overload control

[TOC](#)

In IUA applications, many ISDN End Points (EPs) may be connected via a smaller number of Signaling Gateways (SGs) and thence to a small number of Application Server Processes (ASPs), possibly distributed over hardware resources at one or more hosts. It is possible that an increase in call attempt levels across many or all of the SGs will cause some or all of the ASPs' host hardware resources to become overloaded. This can lead to congestion collapse of some or all of the ASPs, resulting in denial of service. Use of this protocol extension allows an ASP to request a reduction in traffic offered to that ASP by an SG.

Ideally the resulting aggregated offered traffic results in a processing load on the ASP which is close to system capacity. Overload control mechanisms are invoked when load is high and systems are consequently stressed. Although the protocol extension described here is intended for control of load offered to an ASP, the SG may also be overloaded at the same time. Whilst IUA runs over a reliable transport (SCTP), the SG's internal load controls may involve tail drop at internal queues during SG overload. Protocol messages, including messages associated with overload control, may be lost. Hence it is desirable to build in to the protocol extension some robustness for the ASP to ensure that the SG's rate control is operating at the rate most recently commanded by the ASP, rather than at some other rate. For example, during overload, it is important for the ASP to ensure that a

rate restriction has been received by the Overload Control Agent, and to be permitted to resend the rate until the state of the Overload Control Agent's rate restriction is known to be correct. Less obviously, it is important to be sure that a restrictive rate control has been removed from the SG following abatement of the overload. If a restrictive rate control is not removed, normal service may be adversely affected for an extended period.

1.2. Scope

[TOC](#)

The scope of this document is limited to describing:

- *a proposed protocol extension to IUA to support rate control by the ASP of:

- new originating ISDN call admission requests from the SG to the ASP; and

- new originating requests from the SG to the ASP to set up user packet sessions

- *recommended behaviours at the SG and at the ASP to make use of the rate acknowledgement mechanism defined in the protocol extension.

In descriptive text in the remainder of the document, references to the setting up of "new originating ISDN calls" should be taken to include the setting up of new originating user packet sessions.

It may be helpful to state explicitly that the following items are outside the scope of this document:

- *the design and operation of overload control algorithms at the ASP (or higher) application layer, which calculate the Call (Session) Admission Rate values to be transported by the protocol extension described here;

- *the design and operation of the Overload Control Agent in the SG, for example how it identifies new originating ISDN calls, the detailed mechanism for admitting new originating ISDN calls at the required rate, how it satisfies the ISDN protocol requirements of the ISDN End Point whilst not involving the ASP in processing a call setup message, and how it might give priority to certain call types (such as emergency calls or calls from high priority lines) whilst restricting the overall rate of new originating ISDN calls offered to the ASP;

*the application of the mechanism described here to more complex cases such as load-sharing between an SG and a number of ASPs. However it is believed that the protocol mechanism described, which allows independent control for each pairwise relationship between an SG and an ASP, provides the protocol support required for effective control in more complex cases.

Other Standards Development Organisations could produce recommendations covering these and other aspects of a system for overload control, referring to the current document for details of the IUA protocol extension and associated procedures.

1.3. Analogy with etsi_nr overload control

[TOC](#)

The proposed mechanism is analogous to the etsi_nr (Notification Rate) mechanism defined in [\[ETSI NR\] \(ETSI Standard, "NGN Overload Control Architecture; Part 4: Adaptative Control for the MGC," April 2007.\)](#). The protocol extension defined here allows an ASP to regulate originating ISDN calls that are offered to it using ISDN signaling. The mechanism of [\[ETSI NR\] \(ETSI Standard, "NGN Overload Control Architecture; Part 4: Adaptative Control for the MGC," April 2007.\)](#) allows an MGC to regulate originating analogue calls that are offered to it using the Gateway Control Protocol [\[H.248\] \(ITU-T Recommendation, "Gateway control protocol: Version 3," September 2005.\)](#). Discussion in [\[ETSI NR\] \(ETSI Standard, "NGN Overload Control Architecture; Part 4: Adaptative Control for the MGC," April 2007.\)](#) provides additional background to the requirements and implementation of this type of control.

1.4. Applicability to DUA

[TOC](#)

DUA [\[RFC4129\] \(Mukundan, R., "Digital Private Network Signaling System \(DPNSS\)/ Digital Access Signaling System 2 \(DASS 2\) Extensions to the IUA Protocol," August 2005.\)](#) uses the IUA ASPTM message class and hence an ASP MAY use this extension to IUA to regulate the rate of originating DASS2/DPNSS calls offered to the ASP.

[TOC](#)

2. Requirements Notation

The requirements notation used in this document is defined in [\[RFC2119\] \(Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," March 1997.\)](#).

This protocol extension is OPTIONAL, both at the SG and at the ASP. Requirements notation used below is to be interpreted in this context. For example, if text such as "The entity SHALL..." is used below, it is to be interpreted as meaning "If the entity (SG or ASP) implements this OPTIONAL protocol extension, then it SHALL...".

3. Definitions

[TOC](#)

AS

Application Server, as described in [\[RFC4233\] \(Morneault, K., "Integrated Services Digital Network \(ISDN\) Q.921-User Adaptation Layer," January 2006.\)](#).

ASP

Application Server Process, as described in [\[RFC4233\] \(Morneault, K., "Integrated Services Digital Network \(ISDN\) Q.921-User Adaptation Layer," January 2006.\)](#).

MGC

Media Gateway Controller, as described in [\[RFC2719\] \(Ong, L., "Framework Architecture for Signaling Transport," October 1999.\)](#). The use of MGCs in the context of IUA is further described in [\[RFC4233\] \(Morneault, K., "Integrated Services Digital Network \(ISDN\) Q.921-User Adaptation Layer," January 2006.\)](#).

SG

Signaling Gateway as described in [\[RFC2719\] \(Ong, L., "Framework Architecture for Signaling Transport," October 1999.\)](#). The use of SGs in the context of IUA is further described in [\[RFC4233\] \(Morneault, K., "Integrated Services Digital Network \(ISDN\) Q.921-User Adaptation Layer," January 2006.\)](#).

Overload Control Agent

The functional entity residing in the SG that is responsible for ensuring that new originating calls (sessions) admitted to the ASP conform to the rate signalled by the ASP.

Originating ISDN Call

An attempt to establish a transmission path used for transporting voice or voice band data which is initiated by the ISDN user sending an ISDN call establishment message, such as a DSS1 SETUP message.

4. New Admission Rate and Acknowledgement Messages and New Parameter

[TOC](#)

[RFC4233] (Morneault, K., "Integrated Services Digital Network (ISDN) Q.921-User Adaptation Layer," January 2006.) defines an Application Server Process Traffic Maintenance (ASPTM) Message Class within IUA including the messages types ASP Active, ASP Active Ack, ASP Inactive, ASP Inactive Ack.

This document defines two new messages as part of the ASPTM message class to support the operation of an Overload Control Agent in an SG. The first new message is called "ASP Call (Session) Admission Rate". The second new message is the corresponding acknowledgement, and is called "ASP Call (Session) Admission Rate Ack".

This document also defines a new parameter to convey the required originating ISDN Call (Session) Admission Rate value.

For backward compatibility, a Signaling Gateway which does not support the new ASP Call (Session) Admission Rate message MUST follow standard IUA behaviour by discarding the message, and returning an Error (ERR) message with Error Code Unsupported Message Type to the sender.

4.1. New Message Types

[TOC](#)

The following list contains the new message names for the messages defined in this document.

Application Server Process Traffic Maintenance (ASPTM) messages

Value	Message Name	Short Name
TBD1	ASP Call (Session) Admission Rate	ASPCAR
TBD2	ASP Call (Session) Admission Rate Ack	ASPCAR Ack

NOTE TO RFC EDITOR - please substitute IANA allocations for TBD1 and TBD2 above.

In common with all ASPTM messages, both new messages use only the Common Message Header defined in Section 3.1 of [RFC4233] (Morneault, K., "Integrated Services Digital Network (ISDN) Q.921-User Adaptation Layer," January 2006.).

4.2. New Parameter

[TOC](#)

The following new TLV parameter is defined in this document

Parameter ID	Parameter Name
TBD3	Call (Session) Admission Rate

NOTE TO RFC EDITOR - please substitute IANA allocation for TBD3 above.

4.3. ASP Call (Session) Admission Rate (ASPCAR) Message Format

[TOC](#)

The ASP Call (Session) Admission Rate (ASPCAR) message is sent by an ASP to an SG. An SG which supports the new message type SHOULD limit the mean rate of new originating ISDN call setup messages from the SG to the ASP, to no more than the indicated value. The SG MAY implement the rate restriction using a leaky bucket but further details of the means by which the SG achieves this rate-limitation are outside the scope of this document.

The ASPCAR message contains the following parameters:

Parameter	Mandatory/Optional
Call (Session) Admission Rate	(Mandatory)
INFO String	(Optional)

The format for ASPCAR Message parameters is as follows:

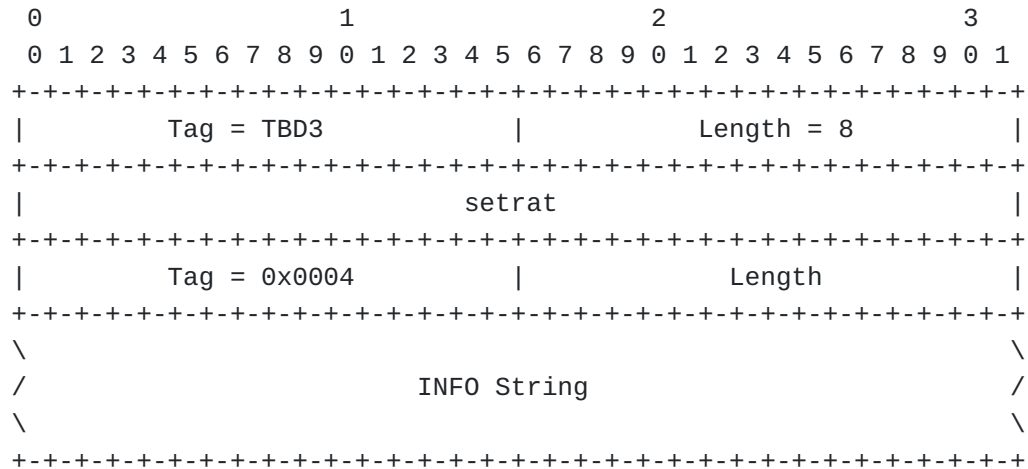


Figure 1: ASP Call (Session) Admission Rate

NOTE TO RFC EDITOR - please substitute IANA allocation for TBD3 above.

setrat: 32-bit 2's complement signed integer

The parameter setrat is the mandatory Call (Session) Admission Rate parameter. It represents a rate, in units of one-thousandth calls/s. For example, a value 5730 indicates that the SG SHOULD limit new calls offered to the ASP to a rate of 5.730 calls/s.

A value of 0 means that the SG SHALL NOT admit any new originating ISDN calls to the particular ASP.

A value >0 means that calls the SG SHALL admit new originating ISDN calls to the particular ASP at a mean rate not exceeding the specified rate. Methods for measurement of the mean rate are application-dependent and are outside the scope of this document.

A value of <0 means that the SG SHALL admit all new originating ISDN calls to the particular ASP.

INFO String

The optional INFO String parameter can carry any meaningful 8-bit ASCII [\[ascii\]](#) (ANSI, "Coded Character Set 7-Bit American Standard Code for Information Interchange," 1986.) character string along with the message. Length of the INFO String parameter is from 0 to 255 characters. No procedures are presently identified for its use, but the INFO String MAY be used for debugging purposes.

4.4. ASP Call (Session) Admission Rate Ack (ASPCAR Ack) Message Format

[TOC](#)

The ASP Call (Session) Admission Rate Ack (ASPCAR Ack) message is used by an SG to acknowledge an ASPCAR message from an ASP at a remote IUA peer.

The ASPCAR Ack message contains the following parameters:

Parameter	Mandatory/Optional
Call (Session) Admission Rate	(Mandatory)
INFO String	(Optional)

The format for ASPCAR Ack Message parameters is as follows:

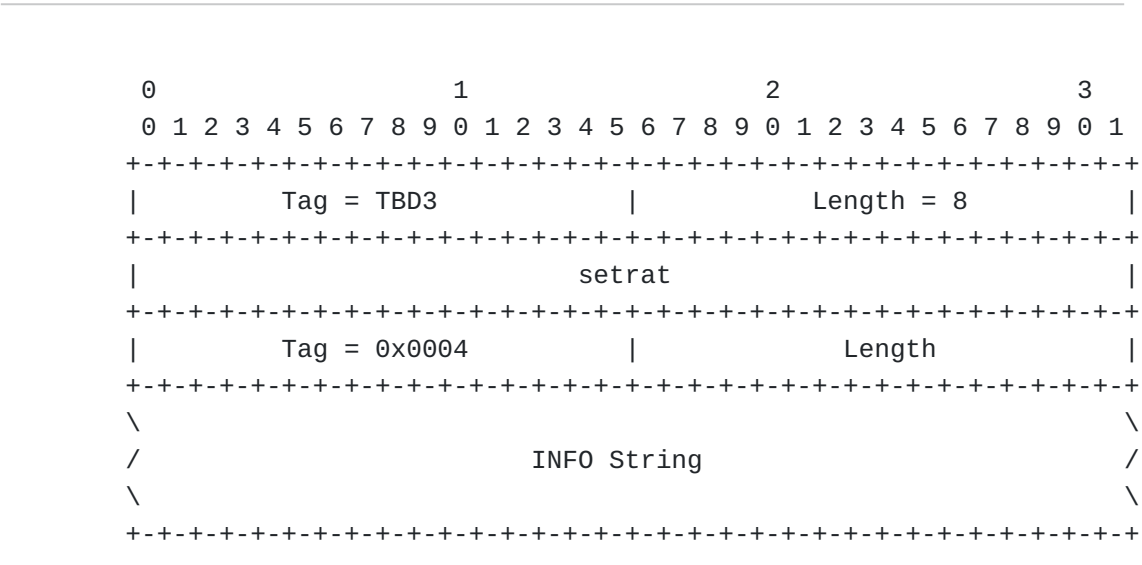


Figure 2: ASP Call (Session) Admission Rate Ack

NOTE TO RFC EDITOR - please substitute IANA allocation for TBD3 above.

setrat: 32-bit 2's complement signed integer

The parameter setrat is the mandatory Call (Session) Admission Rate parameter. For the units of this parameter, see the description in [Section 4.3 \(ASP Call \(Session\) Admission Rate \(ASPCAR\) Message Format\)](#) above.

In the ASPCAR Ack message, the parameter setrat MUST be set by the SG to the setrat value from the ASPCAR message which is acknowledged by this ASPCAR Ack.

The setrat parameter in the ASPCAR Ack provides positive confirmation to the ASP of the successful reception of a specific rate value by the Overload Control Agent. See [Section 1.1 \(Requirement for overload control\)](#) for motivation for this acknowledgement. See [Section 5.4 \(Procedures for use of the rate parameter acknowledgment\)](#) for recommended procedures associated with the use of this parameter.

INFO String

The optional INFO String parameter can carry any meaningful 8-bit ASCII [\[ascii\] \(ANSI, "Coded Character Set 7-Bit American Standard Code for Information Interchange," 1986.\)](#) character string along with the message. Length of the INFO String parameter is from 0 to 255

characters. No procedures are presently identified for its use, but the INFO String MAY be used for debugging purposes.

5. Procedures

[TOC](#)

This section describes procedures associated with the use of the protocol extension described in this document. Support of these extension protocol procedures is OPTIONAL for both the ASP and the SG.

5.1. Basic procedures

[TOC](#)

To limit the SG's call admission rate towards itself, an ASP SHALL send an ASPCAR message containing a Call (Session) Admission Rate parameter with the required setrat value.

On receipt of a valid ASPCAR message the SG SHALL respond to the sender with an ASPCAR Ack containing a Call (Session) Admission Rate parameter which echoes the received setrat value. The SG SHOULD retain the received setrat value as the new current value for rate restriction to that ASP.

The ASP SHALL use the receipt of the ASPCAR Ack message in response to an ASPCAR message as a positive acknowledgement that the SG supports this optional protocol extension and that the new setrat value has been registered by the SG's Overload Control Agent.

If the ASP receives an ERR message with an Error Code "Unsupported Message Type" in response to an ASPCAR message whilst it is awaiting an ASPCAR Ack message, then:

- *it SHOULD send no further ASPCAR messages to that SG

- *it SHOULD stop timer T(ack), providing it can correlate the ERR message with the offending ASPCAR message (e.g. via the optional diagnostic field contained in the ERR message).

- *it MAY follow an alternative strategy of overload control which is outside the scope of this document

5.2. ASP States and the ASPCAR message

[TOC](#)

An SG SHALL accept an ASPCAR message from an ASP when its ASP state is either ASP-ACTIVE or ASP-INACTIVE, see Figure 6 of [\[RFC4233\]](#) (Morneault, K., "Integrated Services Digital Network (ISDN) Q.921-User

[Adaptation Layer," January 2006.](#)). This provides the ASP with the flexibility to apply rate control prior to entry to the ASP-ACTIVE state.

If the SG receives an ASPCAR message whilst in the ASP-DOWN state, then it SHALL return an ERR message with an Error Code "Protocol Error". The SG SHALL NOT take any further action on the ASPCAR message.

When the SG's ASP state (Figure 6 of [\[RFC4233\] \(Morneault, K., "Integrated Services Digital Network \(ISDN\) Q.921-User Adaptation Layer," January 2006.\)](#)) enters state ASP-INACTIVE or ASP-DOWN from any other state, the SG SHALL cease restriction for that ASP and ensure that all offered originating ISDN calls are admitted to the ASP. The ASP SHOULD send ASPCAR only when it is likely that the SG's ASP state is ASP-INACTIVE or ASP-ACTIVE.

The SG will admit all calls following an SG transition to state ASP-INACTIVE, as described above. Hence if it is necessary to maintain a rate control parameter >0 after the SG has moved to state ASP-INACTIVE, the ASP SHOULD send an ASPCAR message to set this rate.

5.3. Applicability to DUA

[TOC](#)

DUA [\[RFC4129\] \(Mukundan, R., "Digital Private Network Signaling System \(DPNSS\)/ Digital Access Signaling System 2 \(DASS 2\) Extensions to the IUA Protocol," August 2005.\)](#) uses the IUA ASPTM message class and hence this proposed extension to IUA can be used to regulate DUA signaling as well as IUA signaling.

In cases where an SG sends new originating ISDN calls to a single ASP using both IUA and DUA signaling, the rate conveyed in ASPCAR messages applies to the aggregate (IUA and DUA traffic combined) of new originating ISDN calls.

5.4. Procedures for use of the rate parameter acknowledgment

[TOC](#)

This section describes the SG and ASP behaviour, including the use of the setrat field in the ASPCAR Ack message, so that the ASP may ensure that the SG is applying the correct rate restriction. It is assumed here that IUA provides reliable in-sequence bidirectional delivery of messages between an ASP and an SG.

Sequencing of Call (Session) Admission Rate messages for a given ASP-SG association SHOULD be preserved within ASPs and SGs. This implies that Call (Session) Admission Rate messages and acknowledgements for a given ASP-SG association SHOULD NOT be routed via different queues, or handled by different processes or threads, in an SG or ASP implementation which uses such techniques internally. However it is recognised that Call (Session) Admission Rate messages and message

acknowledgements might sometimes be dropped in an overloaded SG or an overloaded ASP. Hence if an ASP sends a sequence of ASP Call (Session) Admission Rate messages to an SG (usually intermingled with other protocol traffic), the corresponding sequence of ASP Call (Session) Admission Ack messages will always reach the ASP in order, but might suffer from deletions. An ASP cannot determine whether a loss affected the ASP Call (Session) Admission Rate message in transit to the SG, or the ASP Call (Session) Admission Rate Ack message in transit to the ASP.

Provided that procedures given here are followed, and ordering of ASPCAR and ASPCAR Ack messages is preserved, either the SG Overload Control (OLC) rate will be set to the value requested in a message, or the ASP will not receive an acknowledgement for this value; it is not possible for the ASP to receive an acknowledgement stating a rate equal to the local copy of the current rate, whilst the SG operates a different rate. However, if ordering is not preserved, even if all other aspects of these procedures are followed, it is possible that the latest acknowledgement received and accepted by the ASP at the end of a sequence of rate updates will contain a rate parameter which is not equal to the rate in force at the SG after this sequence of updates. An SG which supports the ASPCAR message type MUST send the ASPCAR Ack message to acknowledge its having successfully received the ASPCAR message, and MUST set the setrat parameter in the ASPCAR Ack message to the value of the setrat parameter which the Overload Control Agent has received from the ASPCAR message being acknowledged.

It is OPTIONAL for the ASP to use the procedure described below to ensure that the rates in the SG and ASP are aligned. However if the ASP uses the rate value returned in ASPCAR Ack to improve robustness, it SHOULD use this procedure.

When the ASP sends an ASP Call (Session) Admission Rate message, it starts (or re-starts) timer T(ack) and stores the value of the Call (Session) Admission Rate parameter. The duration of timer T(ack) is provisionable, with a default of 2 seconds. Note that an ASP Call (Session) Admission Rate message MAY be sent before the ASP has seen an acknowledgement of an earlier ASP Call (Session) Admission Rate message. This second ASPCAR message MAY contain the same value of the Call (Session) Admission Rate parameter as that in the earlier (as yet unacknowledged) ASPCAR message, or MAY contain a different value.

If the ASP receives an ASP Call (Session) Admission Rate Ack message whilst timer T(ack) is running, it compares the value of the Call (Session) Admission Rate parameter from the acknowledgement message with its local stored value:

- *if the values are different, the ASP Call (Session) Admission Rate Ack message is silently discarded, the local stored value (the most-recently transmitted rate) is retained, and timer T(ack) continues to run;

*if the values are the same, the timer T(ack) is stopped, but the local stored value (the most-recently transmitted rate) is retained.

If the ASP receives an ASP Call (Session) Admission Rate Ack message whilst timer T(ack) is not running (an "unexpected ack"), it compares the value of the Call (Session) Admission Rate parameter from the acknowledgement message with its local stored value:

*if the values are the same, the ASP Call (Session) Admission Rate Ack message is silently discarded, and the local stored value (the most-recently transmitted rate) is retained;

*if the values are different, the ASP sends an ASP Call (Session) Admission Rate message, starts timer T(ack) and sets the new local stored value equal to the Call (Session) Admission Rate parameter in the outgoing message. The value of the Call (Session) Admission Rate parameter for the outgoing message MAY be obtained from the old local stored value (the rate which the ASP sent to the SG in the preceding outgoing ASPCAR message) or MAY be an updated value obtained from the ASP's rate control algorithm.

If the timer T(ack) expires before an acknowledgement is received with a rate parameter which matches the local stored value (see above), the ASP sends an ASP Call (Session) Admission Rate message, starts timer T(ack) and stores the value of the Call (Session) Admission Rate parameter. The value of the Call (Session) Admission Rate parameter in the outgoing ASP Call (Session) Admission Rate message MAY be set to the local stored value in existence when T(ack) expired (that is, the last rate which the ASP sent to the SG) or MAY be an updated value obtained from the ASP's rate control algorithm.

In SGs where the implementation of the IUA protocol is separated from the implementation of the overload control function, the IUA implementation SHALL NOT autonomously send the ASPCAR Ack message following receipt of an ASPCAR message. This recommendation is provided because autonomous sending could result in the ASP's receiving an ASPCAR Ack for a rate request which was not made effective at the SG. In ASPs where the implementation of the IUA protocol is separated from the implementation of the overload control function, the IUA implementation MAY implement the functionality described above for processing acknowledgements, including maintaining the per-SG T(ack) timers and the local copy of the current requested rate for each SG, and re-sending ASPCAR messages when required by the mechanism. Alternatively this functionality MAY be implemented in the ASP's overload control function.

6. IANA Considerations

[TOC](#)

This document requests IANA to allocate two new Message Types of the ASPTM Message Class within the registry of Signaling User Adaptation Layer Assignments, Message Types. The new message types are:

Value	Description	Short Name	Reference
TBD1	ASP Call (Session) Admission Rate	ASPCAR	&rfc.number
TBD2	ASP Call (Session) Admission Rate Ack	ASPCAR Ack	&rfc.number

NOTE TO RFC EDITOR - please substitute IANA allocations for TBD1 and TBD2 above.

This document also requests IANA to allocate one new Message Parameter within the registry of Signaling User Adaptation Layer Assignments, Message Parameters. The new message parameter is:

Value	Description	Reference
TBD3	Call (Session) Admission Rate	&rfc.number

NOTE TO RFC EDITOR - please substitute IANA allocation for TBD3 above.

7. Security Considerations

[TOC](#)

[\[RFC4233\]](#) (Morneault, K., "Integrated Services Digital Network (ISDN) Q.921-User Adaptation Layer," January 2006.), defining IUA, refers to [\[RFC3788\]](#) (Loughney, J., "Security Considerations for Signaling Transport (SIGTRAN) Protocols," June 2004.) for security considerations applying to IUA as an instance of a SIGTRAN protocol. The extension to add the ASP Call (Session) Admission Rate parameter, described here, adds a rate control mechanism within the IUA protocol. This mechanism might be used by an attacker, who had obtained control of the protocol traffic, to cut off signalling traffic between an SG and an ASP. However there are already mechanisms within the protocol, including the sending of existing messages in the ASPTM class, which might be used for this purpose by such an attacker. Hence we believe that the extension described here does not introduce any new security considerations.

[TOC](#)

8. Informative Appendix: Message Sequence Diagrams for Acknowledgements

Message sequence diagrams in the following sub-sections provide informative illustrations of the operation of the rate acknowledgement mechanism which is described normatively in [Section 5.4 \(Procedures for use of the rate parameter acknowledgment\)](#) above.

In these diagrams, the overload control entities in the SG and ASP (SG OLC and ASP OLC respectively) are separated from the IUA protocol layer entities at the SG and ASP (SG IUA and ASP IUA respectively).

In these diagrams:

- *the quantity "R" shown under the SG OLC represents the rate in force at the SG;
- *the quantity "r" is a rate parameter requested by the ASP OLC entity, and "in transit" either internally at the ASP, in an ASPCAR message to the SG, or internally at the SG;
- *the quantity "a" is a rate parameter acknowledged by the SG OLC entity, and "in transit" either internally at the SG, in an ASPCAR Ack message to the ASP, or internally at the ASP;
- *the quantity "c" shown under the ASP OLC is the ASP's local copy of the rate most recently sent out as an "r" value in a request; and
- *the notation "a == c" indicates that the rate parameter returned in an ASPCAR Ack message is equal to the ASP's locally stored value of the current rate, and "a != c" indicates that the rate parameter returned in an ASPCAR Ack message is not equal to the ASP's locally stored value of the current rate.

The positive values (1 and 2) for rates in the examples, implying 0.001 and 0.002 calls per second, are unrealistically low for most deployments but are used only for illustration.

8.1. Message sequence for normal transmission

[TOC](#)

[Figure 3 \(Message sequence for normal transmission\)](#) shows the simplest case where an ASPCAR message is sent with requested rate 1, and T(ack) is started. The message is successfully delivered and acknowledged, and the acknowledgement arrives at the ASP before T(ack) expires. The rate value in the acknowledgement matches the ASP's locally-stored value for the most recently requested rate, so T(ack) is stopped.

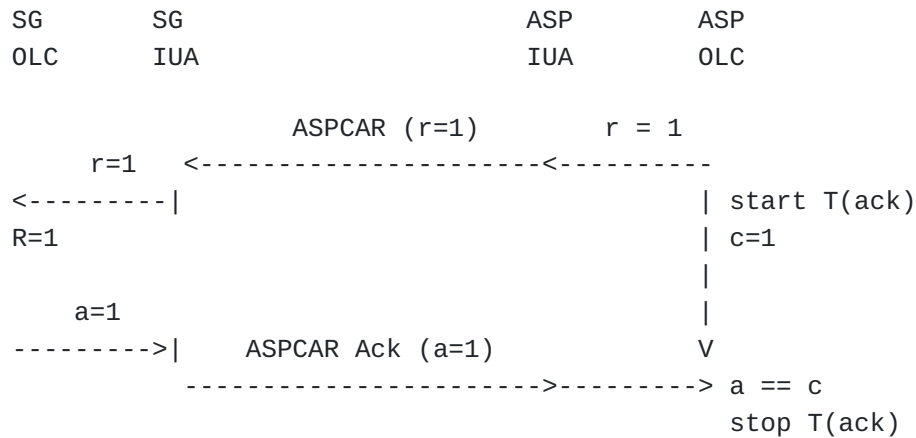


Figure 3: Message sequence for normal transmission

8.2. Message sequence for a lost message

[TOC](#)

[Figure 4 \(Message sequence for a lost message\)](#) shows the case where an ASPCAR message is sent with requested rate 1, but the message is lost in transit to the SG OLC, for example as a result of tail drop in a message queue within the SG application. A SCTP SACK has been sent for the ASPCAR message by the SG and received at the ASP and thus no retransmissions of the ASPCAR will occur at the SCTP layer. However T(ack) will expire, so a second ASPCAR message is sent. In this case, the requested rate in the second ASPCAR message is the same as that in the first ASPCAR message. The second message is received and acknowledged successfully.

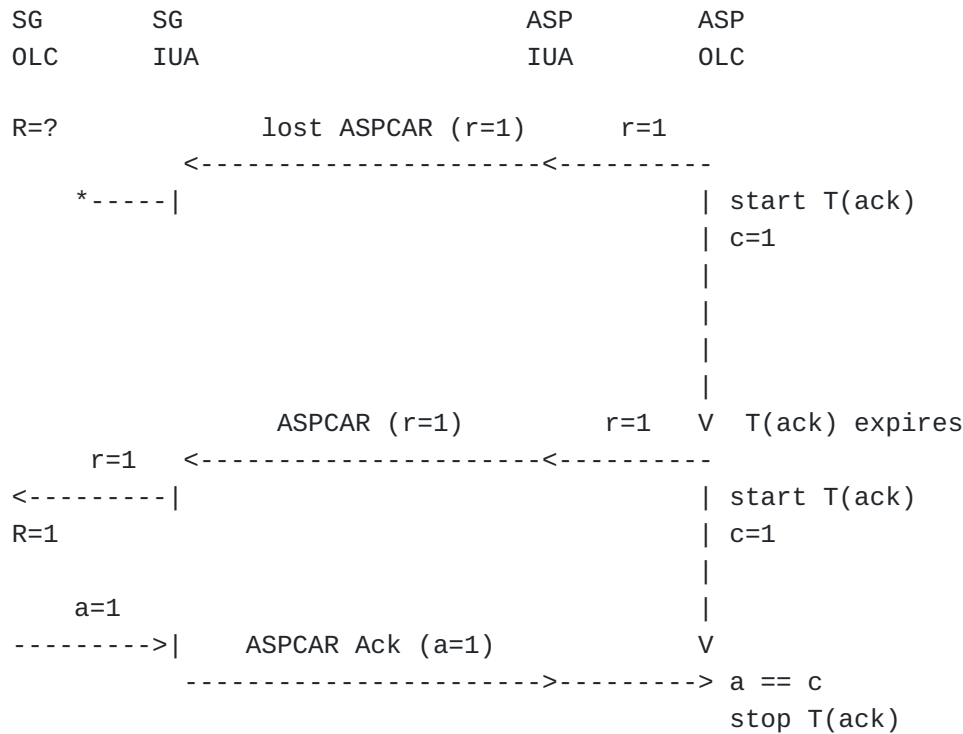


Figure 4: Message sequence for a lost message

8.3. Message sequence for late acknowledgement

[TOC](#)

[Figure 5 \(Message sequence for late acknowledgement\)](#) shows the case where an ASPCAR message is sent with requested rate 1, but the SG is slow to acknowledge the message.

For illustration purposes only, the rate parameter values have been tagged with "x" and "y", but neither IUA nor the protocol extension described here provides any means to distinguish two messages bearing the same parameter value.

T(ack) expires, so a second ASPCAR message is sent. In this case, the requested rate in the second ASPCAR message is the same as that in the first ASPCAR message.

The delayed acknowledgement for the first message arrives after the second message has been sent. However the rate parameter in the acknowledgement matches the ASP local copy of the current requested rate, so T(ack) is stopped. Later the acknowledgement for the second request arrives as an "unexpected ack" (one which arrives whilst no T(ack) is running). Because the rate parameter in the unexpected ack matches the ASP's local copy of the current requested rate, the unexpected ack is silently discarded.

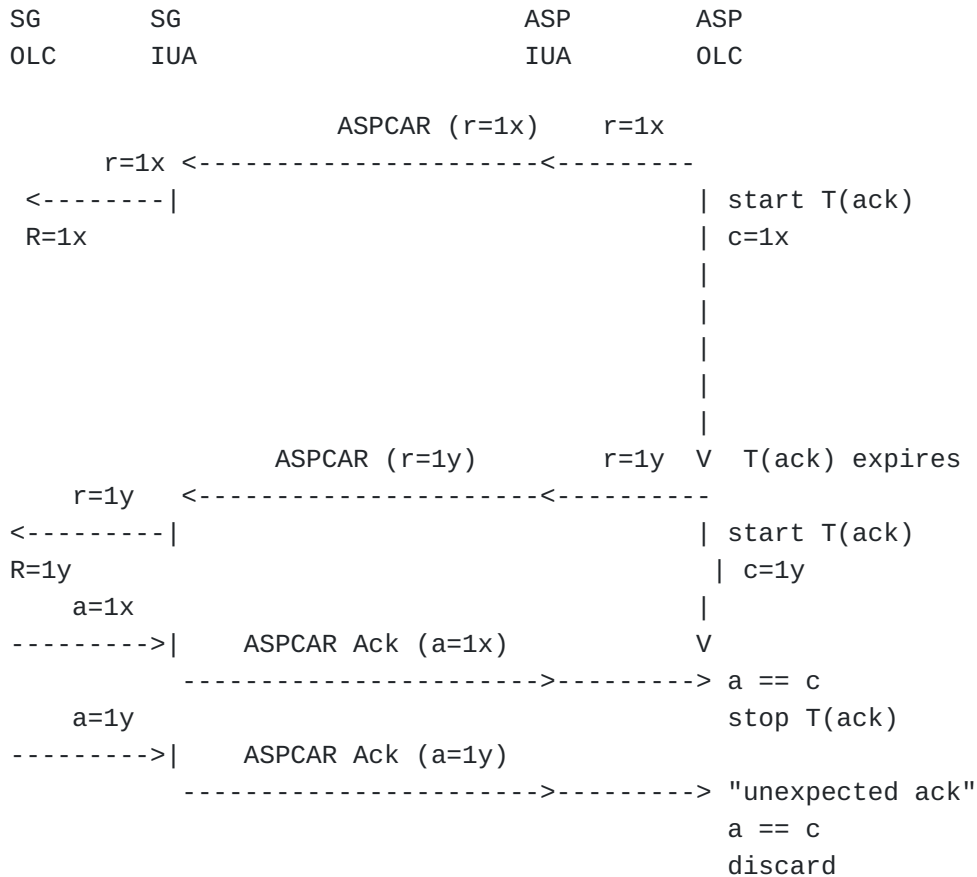


Figure 5: Message sequence for late acknowledgement

8.4. Message sequence for updated rate

[TOC](#)

In [Figure 6 \(Message sequence for updated rate\)](#) the ASP OLC wishes to update the requested rate shortly after an earlier rate request was sent, before the ASP receives the acknowledgement of the earlier request and before expiry of T(ack) for the earlier request. This behaviour is permitted. T(ack) is restarted when the second request is sent, and the local copy of the current rate request is set to the rate value of the second request. The acknowledgement for the first request is received whilst T(ack) is running for the second request message, but because the rate parameter in the acknowledgement does not match the ASP's local copy of the current rate, the acknowledgement is discarded and T(ack) continues to run. Eventually, the acknowledgement

for the second request arrives with a matching rate value, and T(ack) is stopped.

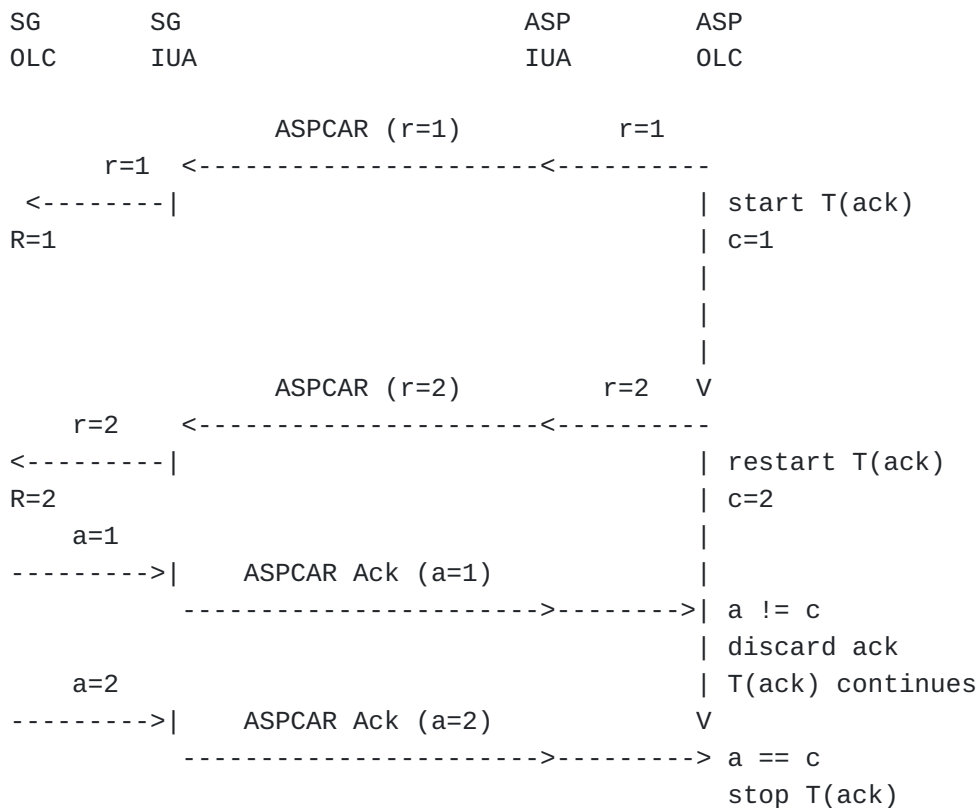


Figure 6: Message sequence for updated rate

8.5. Message sequence for lost message and updated rate

[TOC](#)

[Figure 7 \(Message sequence for lost message and updated rate\)](#) shows a sequence where an updated rate request message is sent shortly after an earlier request, but the first request is lost as a result of tail drop in a message queue within the SG application. A SCTP SACK has been sent by the SG for the first ASPCAR message and received at the ASP and thus no retransmissions of the first ASPCAR message will occur at the SCTP layer. Hence, the first request is not acknowledged. T(ack) is restarted and the ASP's local copy is updated when the second rate request message is sent. The second message is acknowledged normally.

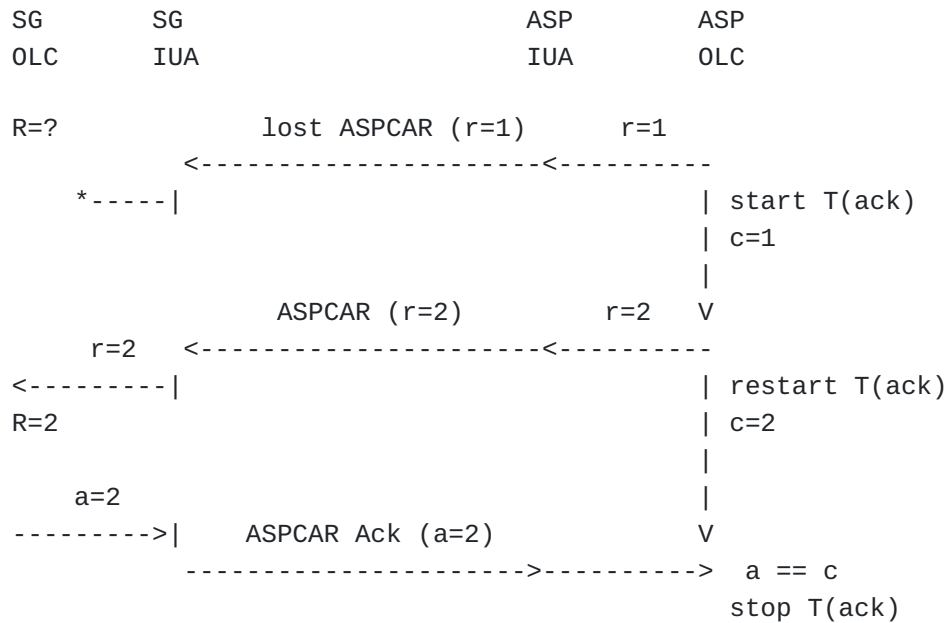


Figure 7: Message sequence for lost message and updated rate

8.6. Message sequence for lost rate update

[TOC](#)

[Figure 8 \(Message sequence for lost rate update\)](#) shows a sequence where an updated rate request message is sent shortly after an earlier request, but the second request is lost as a result of tail drop in a message queue within the SG application. A SCTP SACK acknowledging the second request has been sent by the SG and received at the ASP and thus no retransmission will occur at the SCTP layer for the second request. The acknowledgement for the first request is received whilst T(ack) is running for the second request, but the rate parameter in the acknowledgement does not match the ASP's local copy so the acknowledgement is discarded and T(ack) continues to run. Eventually T(ack) expires, causing a third request to be sent to ensure that the SG knows the ASP's current rate. This third request is delivered and acknowledged normally.

by the SG, and the rate parameter in the ASPCAR Ack matches the ASP's local copy of the current rate, so T(ack) is stopped. The ASP assumes that the rate in force at the SG is 2. Then the first message is processed (out of sequence) by the SG, and the SG's rate controller is set to rate 1. A second ASPCAR Ack is sent, and arrives at the ASP as an unexpected ack, that is, when T(ack) is not running. Because the rate parameter in the second ASPCAR Ack does not match the ASP's local copy, the ASP resends the current rate (2), and the resend is delivered and acknowledged successfully.

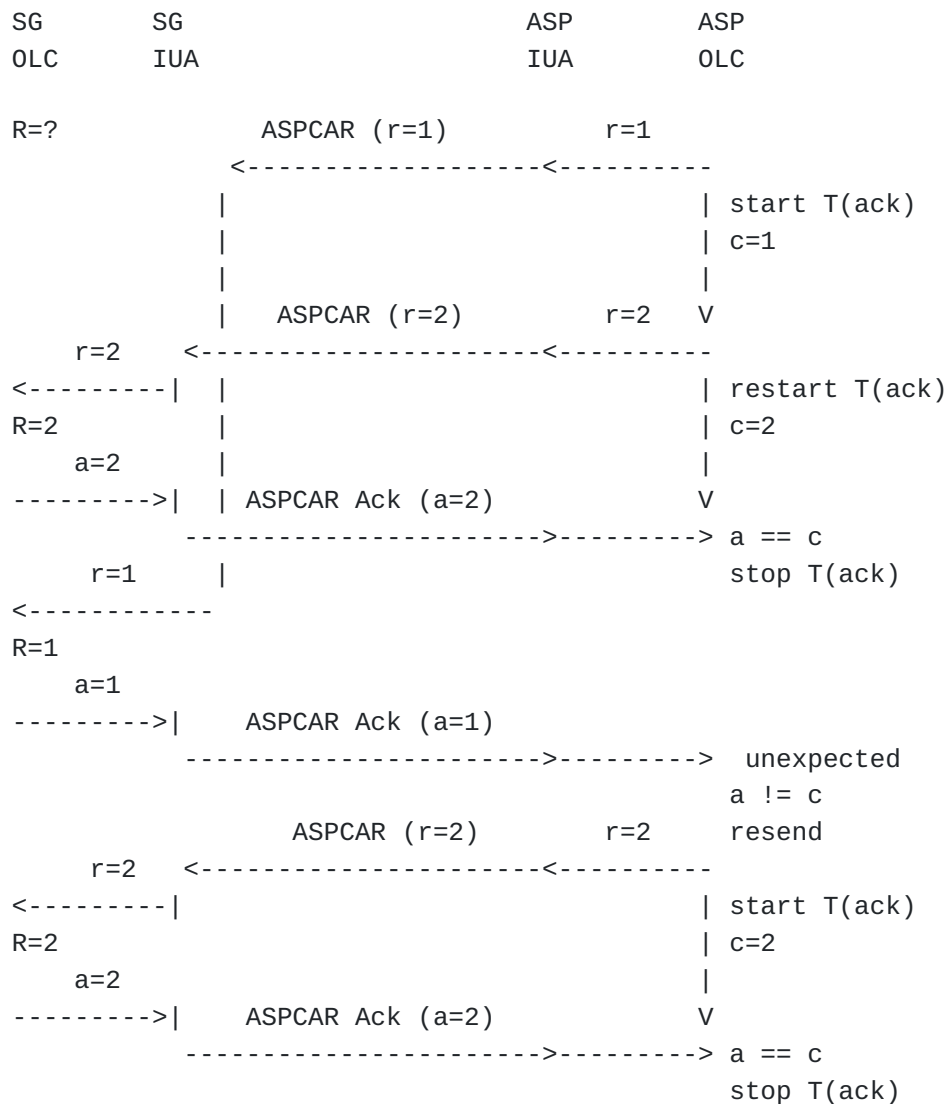


Figure 9: Message sequence for sequencing error

8.8. Message sequence for sequencing error and lost acknowledgement

[TOC](#)

[Figure 10 \(Message sequence for sequencing error and lost acknowledgement\)](#)

shows a case where the mechanism fails to recover from a sequencing error in the SG, which might be caused by successive ASPCAR messages being processed in different queues or by different processing threads. This illustrates the need for the requirement that SGs and ASP SHOULD NOT permit out-of-order delivery.

Two ASPCAR messages are sent in quick succession with rates 1 and -1 respectively. The value -1 was chosen for illustration because it means "admit all calls" and is typically sent at the end of an overload incident. It is very likely that no further rate request messages will be sent for an extended period after such a message is sent, so it is important that the rate request message with value -1 is successfully delivered, and that the value -1 at the SG is not subsequently overwritten as a result of a protocol error.

The second message is processed and acknowledged promptly by the SG, and the rate parameter in the ASPCAR Ack matches the ASP's local copy of the current rate, so T(ack) is stopped. The ASP assumes that the rate in force at the SG is -1. Then the first message is processed (out of sequence) by the SG, and the SG's rate controller is set to rate 1. A second ASPCAR Ack is sent with rate parameter 1, but this ASPCAR Ack is lost in transit to the ASP, perhaps by tail drop from an ASP queue. The SCTP SACK for the second ASPCAR Ack is sent by the ASP to the SG and no retransmission for the second ASPCAR Ack occurs at the SCTP layer. Hence the ASP remains unaware that the SG has "lost" the ASP's most recent rate request (in this example, an instruction to admit all calls).

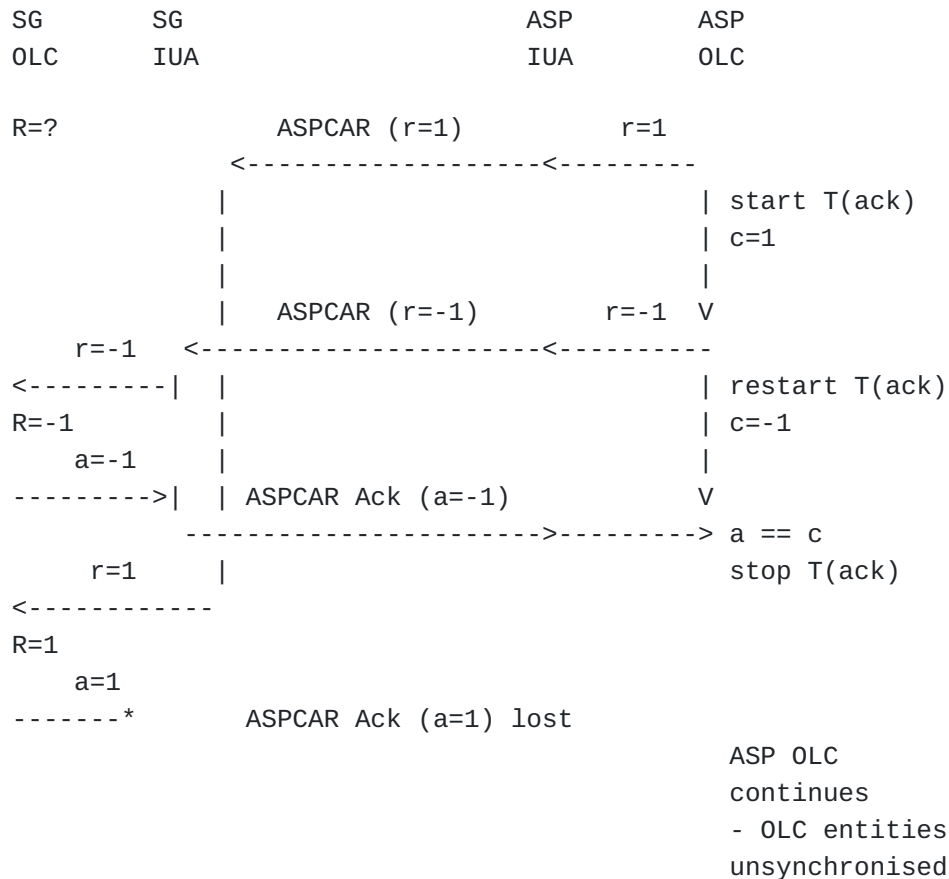


Figure 10: Message sequence for sequencing error and lost acknowledgement

8.9. Message sequence for updated rate and lost acknowledgement

[TOC](#)

[Figure 11 \(Message sequence for updated rate and lost acknowledgement\)](#) shows a case where an acknowledgement is lost in circumstances similar to those of [Figure 10 \(Message sequence for sequencing error and lost acknowledgement\)](#) above, but where the system is constrained to preserve message order. Because ordering is preserved, the acknowledgement mechanism is able to recover from the error. The diagram shows the case where the SG OLC is slow to respond, such that the acknowledgement for the first message is sent after the second message has been processed. However because ordering is preserved, either the SG OLC rate control will be set to the value requested in the later message, or the ASP will not receive an acknowledgement for this value, or both (the case illustrated here). It is not possible for

the ASP to receive an acknowledgement stating a rate equal to the local copy of the current rate, whilst the SG operates a different rate. In the diagram the ASP requests first rate 1, and then rate -1, in quick succession. The SG acknowledges rate 1, but the acknowledgement is slow to arrive, arriving after the second rate request has been sent. Hence the acknowledgement rate does not match the ASP's local copy of the current rate, and is discarded. Because the acknowledgement for the second rate request is lost, the ASP does not receive an acknowledgement matching its current rate, and T(ack) expires. This causes a resend, which is successful in this example.

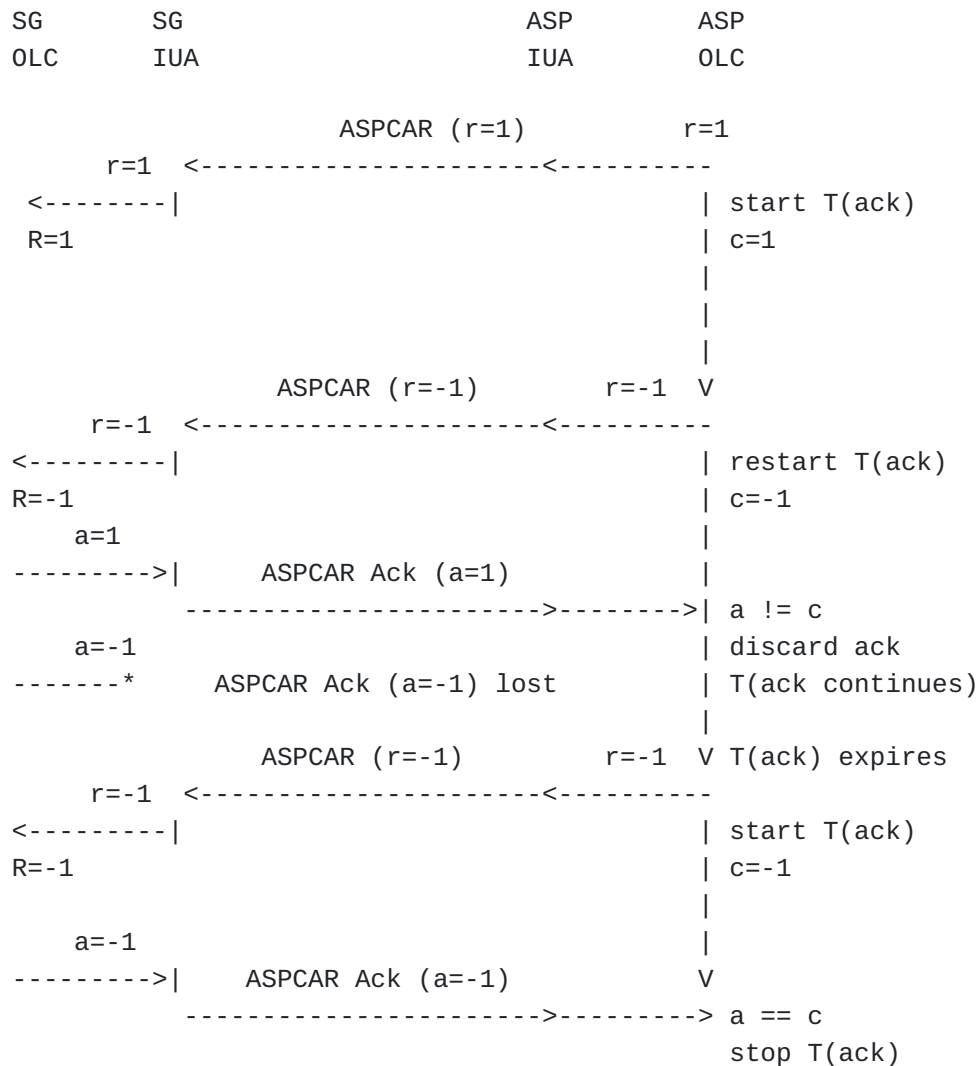


Figure 11: Message sequence for updated rate and lost acknowledgement

9. Contributors

[TOC](#)

The authors gratefully acknowledge key contributions from Juan Harrison.

10. References

[TOC](#)

10.1. Normative References

[TOC](#)

[RFC2119]	Bradner, S., " Key words for use in RFCs to Indicate Requirement Levels ," RFC 2119, BCP 14, March 1997.
[RFC3788]	Loughney, J., " Security Considerations for Signaling Transport (SIGTRAN) Protocols ," RFC 3788, June 2004.
[RFC4233]	Morneault, K., " Integrated Services Digital Network (ISDN) Q.921-User Adaptation Layer ," RFC 4233, January 2006.
[ascii]	ANSI, "Coded Character Set 7-Bit American Standard Code for Information Interchange," ANSI X3.4-1986, 1986.

10.2. Informative References

[TOC](#)

[ETSI_NR]	ETSI Standard, "NGN Overload Control Architecture; Part 4: Adaptative Control for the MGC," ES 283 039-4, April 2007.
[H.248]	ITU-T Recommendation, "Gateway control protocol: Version 3," ITU-T H.248.1, September 2005.
[RFC2719]	Ong, L., " Framework Architecture for Signaling Transport ," RFC 2719, October 1999.
[RFC4129]	Mukundan, R., " Digital Private Network Signaling System (DPNSS)/ Digital Access Signaling System 2 (DASS 2) Extensions to the IUA Protocol ," RFC 4129, August 2005.

Authors' Addresses

[TOC](#)

	Nick Stewart
	BT
	Aquarius 4M2 3
	Adastral Park
	Martlesham Heath

	Ipswich, Suffolk IP5 3RE
	United Kingdom
Phone:	+44 1473 649579
Email:	nick.m.stewart@bt.com
	Geoff Hunt
	BT
	Orion 2 PP3
	Adastral Park
	Martlesham Heath
	Ipswich, Suffolk IP5 3RE
	United Kingdom
Phone:	+44 1473 651704
Email:	geoff.hunt@bt.com
	Dal Chohan
	Fujitsu Telecommunications Europe Limited
	Birmingham Business Park
	Solihull Parkway
	Birmingham, West Midlands B37 7YU
	United Kingdom
Phone:	+44 121 717 6177
Email:	d.chohan@ftel.co.uk