

Diameter Maintenance and Extensions (DIME)	V. Fajardo, Ed.	
Internet-Draft	T. Asveren	
Intended status: Informational	Sonus Network	
Expires: May 28, 2009	H. Tschofenig	
	Nokia Siemens Networks	
	G. McGregor	
	Alcatel-Lucent	
	J. Loughney	
	Nokia Research Center	
	November 24, 2008	

[TOC](#)

## Diameter Applications Design Guidelines draft-ietf-dime-app-design-guide-08.txt

### Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of 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/1id-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 May 28, 2009.

### Abstract

The Diameter Base protocol provides updated rules on how to extend Diameter by modifying and/or deriving from existing applications or creating entirely new applications. This is a companion document to the Diameter Base protocol that further explains and clarifies these rules. It is meant as a guidelines document and therefore it does not add, remove or change existing rules.

---

## Table of Contents

- [1.](#) Introduction
- [2.](#) Terminology
- [3.](#) Overview
- [4.](#) Adding a new command
- [5.](#) Deleting a Command
- [6.](#) Reusing Existing Commands
  - [6.1.](#) Adding AVPs to a Command
  - [6.2.](#) Deleting AVPs from a Command
- [7.](#) Reusing Existing AVPs
- [8.](#) Rules for new Applications
  - [8.1.](#) Use of Application-Id in a Message
  - [8.2.](#) Application Specific Session State Machine
- [9.](#) End-to-End Applications Capabilities Exchange
- [10.](#) Diameter Accounting Support
- [11.](#) Generic Diameter Extensions
- [12.](#) IANA Considerations
- [13.](#) Security Considerations
- [14.](#) Contributors
- [15.](#) Acknowledgments
- [16.](#) References
  - [16.1.](#) Normative References
  - [16.2.](#) Informative References
- [§](#) Authors' Addresses
- [§](#) Intellectual Property and Copyright Statements

---

## 1. Introduction

[TOC](#)

The Diameter Base protocol document defines rules on how one would extend Diameter (see Section 1.2 of [\[1\] \(Fajardo, V., Arkko, J., Loughney, J., and G. Zorn, "Diameter Base Protocol," April 2010.\)](#)). In the context of this document, extending Diameter means one of the following:

1. A new functionality is being added to an existing Diameter application without defining a new application.
2. A new Diameter application is being defined by extending an existing application.
3. A completely new application is being defined that has nothing in common with existing applications.
4. A new generic functionality is being defined that can be reused across different applications.

All of these choices are design decisions that can be done by any combination of reusing existing or defining new commands, AVPs or AVP values. Protocol designers do, however, not have total freedom when making their design. A number of rules defined in [1] (Fajardo, V., Arkko, J., Loughney, J., and G. Zorn, "Diameter Base Protocol," April 2010.) place constraints on when an extension demands a new Diameter Application to be defined or a new Command Code to be registered. The objective of this document is the following:

- \*Clarify updated Diameter extensibility rules in the Diameter Base Protocol.

- \*Clarify usage of certain Diameter functionalities which are not explicitly described in the Diameter Base specification.

- \*Discuss design choices and provide guidelines when defining applications.

- \*Present tradeoffs of design choices.

---

## 2. Terminology

[TOC](#)

This document reuses the terminology used in [1] (Fajardo, V., Arkko, J., Loughney, J., and G. Zorn, "Diameter Base Protocol," April 2010.).

---

## 3. Overview

[TOC](#)

As it is currently interpreted and practiced, the Diameter Base protocol is a two-layer protocol. The lower layer is mainly responsible for managing connections between neighboring peers and for message routing. The upper layer is where the Diameter applications reside. This model is in line with a Diameter node having an application layer and a peer-to-peer delivery layer. The Diameter Base protocol document completely defines the architecture and behavior of the message delivery layer and then provides the framework for designing Diameter applications on the application layer. This framework includes definitions of application sessions and accounting support (see Section 8 and 9 of [1] (Fajardo, V., Arkko, J., Loughney, J., and G. Zorn, "Diameter Base Protocol," April 2010.)). The remainder of this document also treats a Diameter node as a single instance of a Diameter message delivery layer and one or more Diameter applications using it.

Extending Diameter can mean the reuse of commands, AVPs and AVP values in any combination for the purpose of inheriting the features of an existing Diameter application. This section discusses the rules on how such reuse can be done.

When reusing existing applications, the requirements of the new applications are typically not completely unique and hence a lot can be re-used from existing specifications. Therefore, there is a greater likelihood of ambiguity on how much of the existing application can be reused and what would be the implications for both the new and existing application. To broadly categorize, the rules for reusing existing applications can fall into two categories, as listed below.

**Minor Extension:** This, for example, is the case when optional AVPs are being defined. In general, this includes everything that is not covered by the next category. The standardization effort will be fairly small.

**Major Extension:** Such an extension requires the definition of a new Diameter application. The rules outlined in Section 1.2 of [\[1\] \(Fajardo, V., Arkko, J., Loughney, J., and G. Zorn, "Diameter Base Protocol," April 2010.\)](#) indicate when an extension requires a new Command Code to be registered and when new Diameter applications have to be defined. Typically, these types of extensions require a longer and more careful effort depending on the degree of re-use. Therefore, the amount of time and effort to complete the specification should also be considered by the designer.

The subsequent sections provide discussions about the specific Diameter Base protocol rules.

---

#### 4. Adding a new command

[TOC](#)

Adding a new command is considered a major extension and requires a new Diameter application to be defined. Adding a new command to an application means either defining a completely new command or importing an existing command from another application whereby the new application inherits some or all of the functionality of the application where the command came from. In the former case, the decision to create a new application is straightforward since this is typically a result of adding a new functionality that does not exist yet. For the latter, the decision will depend on whether importing the command in a new application is more suitable than simply using the existing application as it is in conjunction with any other application. Therefore, a case by case study of each application requirement should be applied.

In general, it is difficult to come to a hard guideline, and so a case by case study of each application requirement should be applied. Before adding or importing a command, application designers should consider the following:

\*Can the new functionality be fulfilled by creating a new application independent from the existing applications? In this case, both old and new application can work independent of, but cooperating with each other.

\*Can the existing application be reused without major extensions that requires the definition of a new application, e.g. new functionality introduced by the creation of new optional AVPs.

\*Care should be taken to avoid a liberal method of importing existing commands that results in a monolithic and hard to manage application which supports many different functionalities.

---

## 5. Deleting a Command

[TOC](#)

Although this process is not typical, removing a command to an application requires a new Diameter application to be defined. It is unusual to delete an existing command from an application for the sake of deleting it or the functionality it represents. This normally indicates of a flawed design. An exception might be if the intent of the deletion is to create a newer version of the same application which is somehow simpler than the previous version.

---

## 6. Reusing Existing Commands

[TOC](#)

This section discusses rules in adding and/or deleting AVPs from an existing command of an existing application. The cases described in this section may not necessarily result in the creation of new applications.

---

### 6.1. Adding AVPs to a Command

[TOC](#)

Based on the rules in [\[1\] \(Fajardo, V., Arkko, J., Loughney, J., and G. Zorn, "Diameter Base Protocol," April 2010.\)](#), AVPs that are added to an existing command can be categorized into:

\*Mandatory to understand AVPs. As defined in [\[1\] \(Fajardo, V., Arkko, J., Loughney, J., and G. Zorn, "Diameter Base Protocol," April 2010.\)](#), these are AVPs with the M-bit flag set, which means that a Diameter node receiving are required to understand not only their values but their semantics. Failure to do so will cause an message handling error. This is regardless of whether these AVPs are required or optional as specified by the commands ABNF.

\*Optional AVPs. [TBD]

The rules are strict in the case where the AVPs to be added are mandatory. A mandatory AVP cannot be added to or deleted from an existing command with defining a new Diameter application. [\[1\] \(Fajardo, V., Arkko, J., Loughney, J., and G. Zorn, "Diameter Base Protocol," April 2010.\)](#) states that doing so would require the definition of a new application. This falls into the "Major Extensions" category. Despite the clarity of the rule, ambiguity still arises when trying to decide whether a new AVP being added should be mandatory to begin with. The follow are a few common questions that application designers should contemplate when trying to decide:

\*Is it required for the receiving side to be able to process and understand the AVP and its content (rather than just writing it's content into to a file)?

\*Do the AVPs change the state machine of the application ?

\*Would the presence of the new AVPs (or the newly specified value contained in an existing AVP) lead to a different number of roundtrips, effectively changing the state machine of the application?

\*Would the AVP be used to differentiate between old and new versions of the same application whereby the two versions are not backward compatible?

\*Will it have duality in meaning, i.e., be used to carry application related information as well as be used to indicate that the message is for a new application?

When one of the above questions can be answered with 'yes' then the M-bit has to be set. If application designers are contemplating on the use of optional AVPs instead, then the following are some of the pitfalls that should be avoided:

\*Use of optional AVPs with intersecting meaning. One AVP has partially the same usage and meaning as another AVP. The presence

of both can lead to confusion.

\*An optional AVPs with dual purpose, i.e. to carry applications data as well as to indicate support for one or more features. This has a tendency to introduce interpretation issues.

\*Adding one or more optional AVPs and indicating (usually within descriptive text for the command) that at least one of them has to be present in the command. This essentially circumventing the ABNF and is equivalent to adding a mandatory AVPs to the command.

These practices generally result in interoperability issues and should be avoided as much as possible.

---

## 6.2. Deleting AVPs from a Command

[TOC](#)

When deleting an AVP from a Command the following cases need to be differentiated:

\*An AVP that is indicated as {} in the ABNF in the Command (with or without the M-bit set). In this case the new Command Code and subsequently a new Diameter application has to be specified.

\*An AVP that is indicated as [] in the ABNF in the Command (with the M-bit set). No new Command Code has to be specified but the definition of a new Diameter application is required.

\*An AVP that is indicated as [] in the ABNF in the Command (without the M-bit set). In this case the AVP can be deleted without consequences.

If possible application designers should attempt the reuse the Command ABNF without modification and simply ignore (but not delete) any optional AVP that will not be used. This is to maintain compatibility with existing applications that will not know about the new functionality as well as maintain the integrity of existing dictionaries.

---

## 7. Reusing Existing AVPs

[TOC](#)

This section discusses rules in adding, deleting or modifying the specified values of an AVP.

When reusing AVPs in a new application, the AVP flag setting, such as the mandatory flag ('M'-bit), has to be re-evaluated for a new Diameter application and, if necessary, even for every Command within the application. In general, for AVPs defined outside of the base protocol, its mandatory characteristics are tied to its role within an application and Command.

---

## 8. Rules for new Applications

[TOC](#)

The general theme of Diameter extensibility is to reuse commands, AVPs and AVP values as much as possible. However, some of the extensibility rules described in the previous section also apply to scenarios where a designer is trying to define a completely new Diameter application.

This section discusses the case where new applications have requirements that cannot be filled by existing applications and would require definition of completely new commands, AVPs and/or AVP values. Typically, there is little ambiguity about the decision to create these types of applications. Some examples are the interfaces defined for the IP Multimedia Subsystem of 3GPP, i.e. Cx/Dx ([\[2\]](#) (3GPP, "IMS Cx and Dx interfaces : signalling flows and message contents," .)) and [\[3\]](#) (3GPP, "IMS Cx and Dx interfaces based on the Diameter protocol; Protocol details," .)), Sh ([\[4\]](#) (3GPP, "IMS Sh interface : signalling flows and message content," .)) and [\[5\]](#) (3GPP, "IMS Sh interface based on the Diameter protocol; Protocol details," .)) etc.

Application designers should also follow the theme of Diameter extensibility which in this case means to import existing AVPs and AVP values for any newly defined commands. In certain cases where accounting will be used, the models described in [Section 10 \(Diameter Accounting Support\)](#) should also be considered. Though some decisions may be clear, designers should also consider certain aspects of defining a new application. Some of these aspects are described in following sections.

---

### 8.1. Use of Application-Id in a Message

[TOC](#)

When designing new applications, designers should specify that the application ID carried in all session level messages must be the application ID of the application using those messages. This includes the session level messages defined in base protocol, i.e., RAR/RAA, STR/STA, ASR/ASA and possibly ACR/ACA in the coupled accounting model, see [Section 10 \(Diameter Accounting Support\)](#). Existing specifications may not adhere to this rule for historical or other reasons. However, this scheme should be followed to avoid possible routing problems for these messages.



In general, when a new application has been allocated with a new application id and it also reuses existing commands with or without modifications (Sec 4.1), it must use the newly allocated application id in the header and in all relevant application id AVPs (Auth-Application-Id or Acct-Application-Id) present in the commands message body.

Additionally, application designs using Vendor-Specific-Application-Id AVP should not use the Vendor-Id AVP to further dissect or differentiate the vendor-specification application id. Diameter routing is not based on the Vendor-Id. As such, the Vendor-ID should not be used as an additional input for routing or delivery of messages. In general, the Vendor-Id AVP is an informational AVP only and kept for backward compatibility reasons.

---

## 8.2. Application Specific Session State Machine

[TOC](#)

Section 8 of [\[1\] \(Fajardo, V., Arkko, J., Loughney, J., and G. Zorn, "Diameter Base Protocol," April 2010.\)](#) provides session state machines for authentication, authorization and accounting (AAA) services. When a new application is being defined that cannot clearly be categorized into any of these services it is recommended that the application itself define its own session state machine. The existing session state machines defined by [\[1\] \(Fajardo, V., Arkko, J., Loughney, J., and G. Zorn, "Diameter Base Protocol," April 2010.\)](#) is not intended for general use beyond AAA services, therefore any behavior not covered by that category would not fit well. Support for server initiated request is a clear example where an application specific session state machine would be needed, for example, the Rw interface for ITU-T push model.

---

## 9. End-to-End Applications Capabilities Exchange

[TOC](#)

It is also possible that applications can use optional AVPs to exchange application specific capabilities and features. These AVPs are exchanged on an end-to-end basis. Examples of this can be found in [\[6\] \(Korhonen, J., Bournelle, J., Tschofenig, H., Perkins, C., and K. Chowdhury, "Diameter Mobile IPv6: Support for Network Access Server to Diameter Server Interaction," January 2009.\)](#) and [\[7\] \(Korhonen, J., Tschofenig, H., Arumaithurai, M., Jones, M., and A. Lior, "Traffic Classification and Quality of Service Attributes for Diameter," December 2009.\)](#).

The end-to-end capabilities AVPs can aid in the following cases:

\*Formalizing the way new functionality is added to existing applications by announcing support for it.

\*Applications that do not understand these AVP can discard it upon receipt. In such case, senders of the AVP can also safely assume the receiving end-point does not support any functionality carried by the AVP if it is not present in subsequent responses.

\*Useful in cases where deployment choices are offered and the generic design can be made available for a number of applications.

Note that this list is not meant to be comprehensive.

---

## 10. Diameter Accounting Support

[TOC](#)

Accounting can be treated as an auxiliary application which is used in support of other applications. In most cases, accounting support is required when defining new applications. This document provides two(2) possible models for using accounting:

### **Split Accounting Model**

In this model, the accounting messages will use the Diameter base accounting application ID (value of 3). The design implication for this is that the accounting is treated as an independent application, especially during Diameter routing. This means that accounting commands emanating from an application may be routed separately from the rest of the other application messages. This may also imply that the messages generally end up in a central accounting server. A split accounting model is a good design choice when:

\*The application itself will not define its own unique accounting commands.

\*The overall system architecture permits the use of centralized accounting for one or more Diameter applications.

Centralizing accounting may have advantages but there are also drawbacks. The model assumes that the accounting server can somehow differentiate received accounting messages. Since the received accounting messages can be for any application and/or service, the accounting server has to have a method to uniquely match accounting messages with applications and/or services being accounted for. This may mean defining new AVPs, checking the presence, absence or contents of existing AVPs or checking the contents of the accounting records itself. But in general, there is no clean and generic scheme for sorting these messages. Therefore, the use of this model is

recommended only when all received accounting messages can be clearly identified and sorted. For most cases, the use of Coupled Accounting Model is recommended.

#### **Coupled Accounting Model**

In this model, the accounting messages will use the application ID of the application using the accounting service. The design implication for this is that the accounting messages are tightly coupled with the application itself; meaning that accounting messages will be routed like any other application messages. It would then be the responsibility of the application server (application entity receiving the ACR message) to send the accounting records carried by the accounting messages to the proper accounting server. The application server is also responsible for formulating a proper response (ACA). A coupled accounting model is a good design choice when:

\*The system architecture or deployment will not provide an accounting server that supports Diameter.

\*The system architecture or deployment requires that the accounting service for the specific application should be handled by the application itself.

\*The application server is provisioned to use a different protocol to access the accounting server; e.g., via LDAP, SOAP etc. This includes attempting to support older accounting systems that are not Diameter aware.

In all cases above, there will generally be no direct Diameter access to the accounting server.

These models provide a basis for using accounting messages. Application designers may obviously deviate from these models provided that the factors being addressed here have also been taken into account. Though it is not recommended, examples of other methods might be defining a new set of commands to carry application specific accounting records.

---

## **11. Generic Diameter Extensions**

[TOC](#)

Generic Diameter extensions are AVPs, commands or applications that are designed to support other Diameter applications. They are auxiliary applications meant to improve or enhance the Diameter protocol itself or Diameter applications/functionality. Some examples include the extensions to support auditing and redundancy (see [\[12\] \(Calhoun, P., "Diameter Resource Management Extensions,"](#)

[March 2001.](#))), improvements in duplicate detection scheme (see [\[13\]](#) (Asveren, T., "Diameter Duplicate Detection Cons.," August 2006.)), and piggybacking of QoS attributes (see [\[7\]](#) (Korhonen, J., Tschofenig, H., Arumaithurai, M., Jones, M., and A. Lior, "Traffic Classification and Quality of Service Attributes for Diameter," December 2009.)).

Since generic extensions can cover many aspects of Diameter and Diameter applications, it is not possible to enumerate all the probable scenarios in this document. However, some of the most common considerations are as follows:

- \*Backward compatibility: Dealing with existing applications that do not understand the new extension. Designers also have to make sure that new extensions do not break expected message delivery layer behavior.

- \*Forward compatibility: Making sure that the design will not introduce undue restrictions for future applications. Future applications attempting to support this feature should not have to go through great lengths to implement any new extensions.

- \*Tradeoffs in signaling: Designers may have to choose between the use of optional AVPs piggybacked onto existing commands versus defining new commands and applications. Optional AVPs are simpler to implement and may not need changes to existing applications; However, the drawback is that the timing of sending extension data will be tied to when the application would be sending a message. This has consequences if the application and the extensions have different timing requirements. The use of commands and applications solves this issue but the tradeoff is the additional complexity of defining and deploying a new application. It is left up to the designer to find a good balance among these tradeoffs based on the requirements of the extension.

In practice, it is often the case that the generic extensions use optional AVPs because it's simple and not intrusive to the application that would carry it. Peers that do not support the generic extensions need not understand nor recognize these optional AVPs. However, it is recommended that the authors of the extension specify the context or usage of the optional AVPs. As an example, in the case that the AVP can be used only by a specific set of applications then the specification must enumerate these applications and the scenarios when the optional AVPs will be used. In the case where the optional AVPs can be carried by any application, it should be sufficient to specify such a use case and perhaps provide specific examples of applications using them.

In most cases, these optional AVPs piggybacked by applications would be defined as a Grouped AVP and it would encapsulate all the functionality of the generic extension. In practice, it is not uncommon that the Grouped AVP will encapsulate an existing AVP that has previously been defined as mandatory ('M'-bit set) e.g., 3GPP IMS Cx / Dx interfaces ([\[2\]](#) (3GPP, "IMS Cx and Dx interfaces :

[signalling flows and message contents," .\)](#) and [\[3\] \(3GPP, "IMS Cx and Dx interfaces based on the Diameter protocol; Protocol details," .\)\)](#).

---

## 12. IANA Considerations

[TOC](#)

This document does not require actions by IANA.

---

## 13. Security Considerations

[TOC](#)

This document does provides guidelines and considerations for extending Diameter and Diameter applications. It does not define nor address security related protocols or schemes.

---

## 14. Contributors

[TOC](#)

The content of this document was influenced by a design team created to revisit the Diameter extensibility rules. The team consisting of the members listed below was formed in February 2008 and finished its work in June 2008.

\*Avi Lior

\*Glen Zorn

\*Jari Arkko

\*Lionel Morand

\*Mark Jones

\*Victor Fajardo

\*Tolga Asveren

\*Jouni Korhonen

\*Glenn McGregor

\*Hannes Tschofenig

\*Dave Frascone

---

## 15. Acknowledgments

[TOC](#)

We greatly appreciate the insight provided by Diameter implementers who have highlighted the issues and concerns being addressed by this document.

---

## 16. References

[TOC](#)

### 16.1. Normative References

[TOC](#)

[1]	Fajardo, V., Arkko, J., Loughney, J., and G. Zorn, " <a href="#">Diameter Base Protocol</a> ," draft-ietf-dime-rfc3588bis-20 (work in progress), April 2010 ( <a href="#">TXT</a> ).
[2]	3GPP, "IMS Cx and Dx interfaces : signalling flows and message contents," 3GPP TS 29.228 Version 7.0.0 2006.
[3]	3GPP, "IMS Cx and Dx interfaces based on the Diameter protocol; Protocol details," 3GPP TS 29.229 Version 7.0.0 2006.
[4]	3GPP, "IMS Sh interface : signalling flows and message content," 3GPP TS 29.328 Version 6.8.0 2005.
[5]	3GPP, "IMS Sh interface based on the Diameter protocol; Protocol details," 3GPP TS 29.329 Version 6.6.0 2005.
[6]	Korhonen, J., Bournelle, J., Tschofenig, H., Perkins, C., and K. Chowdhury, " <a href="#">Diameter Mobile IPv6: Support for Network Access Server to Diameter Server Interaction</a> ," draft-ietf-dime-mip6-integrated-12 (work in progress), January 2009 ( <a href="#">TXT</a> ).
[7]	Korhonen, J., Tschofenig, H., Arumaithurai, M., Jones, M., and A. Lior, " <a href="#">Traffic Classification and Quality of Service Attributes for Diameter</a> ," draft-ietf-dime-qos-attributes-15 (work in progress), December 2009 ( <a href="#">TXT</a> ).
[8]	Hakala, H., Mattila, L., Koskinen, J-P., Stura, M., and J. Loughney, " <a href="#">Diameter Credit-Control Application</a> ," RFC 4006, August 2005 ( <a href="#">TXT</a> ).
[9]	Calhoun, P., Loughney, J., Guttman, E., Zorn, G., and J. Arkko, " <a href="#">Diameter Base Protocol</a> ," RFC 3588, September 2003 ( <a href="#">TXT</a> ).
[10]	Korhonen, J., Tschofenig, H., Bournelle, J., Giaretta, G., and M. Nakhjiri, " <a href="#">Diameter Mobile IPv6: Support for Home Agent to Diameter Server Interaction</a> ," draft-ietf-dime-mip6-split-17 (work in progress), April 2009 ( <a href="#">TXT</a> ).
[11]	Sun, D., McCann, P., Tschofenig, H., ZOU, T., Doria, A., and G. Zorn, " <a href="#">Diameter Quality of Service Application</a> ," draft-ietf-dime-diameter-qos-15 (work in progress), March 2010 ( <a href="#">TXT</a> ).

---

## 16.2. Informative References

[TOC](#)

[12]	Calhoun, P., " <a href="#">Diameter Resource Management Extensions</a> ," draft-calhoun-diameter-res-mgmt-08.txt (work in progress), March 2001 ( <a href="#">TXT</a> ).
[13]	Asveren, T., " <a href="#">Diameter Duplicate Detection Cons.</a> ," draft-asveren-dime-dupcons-00 (work in progress), August 2006 ( <a href="#">TXT</a> ).

---

## Authors' Addresses

[TOC](#)

	Victor Fajardo (editor)
	Toshiba America Research Inc.
	One Telcordia Drive, #1S222
	Piscataway, NJ 08854
	USA
	Email: <a href="mailto:vfajardo@tari.toshiba.com">vfajardo@tari.toshiba.com</a>
	Tolga Asveren
	Sonus Network
	4400 Route 9 South
	Freehold, NJ 07728
	USA
	Email: <a href="mailto:tasveren@sonusnet.com">tasveren@sonusnet.com</a>
	Hannes Tschofenig
	Nokia Siemens Networks
	Linnoitustie 6
	Espoo 02600
	Finland
	Phone: +358 (50) 4871445
	Email: <a href="mailto:Hannes.Tschofenig@gmx.net">Hannes.Tschofenig@gmx.net</a>
	URI: <a href="http://www.tschofenig.priv.at">http://www.tschofenig.priv.at</a>
	Glenn McGregor
	Alcatel-Lucent
	3461 Robin Ln Ste 1
	Cameron Park, CA 95682
	USA
	Email: <a href="mailto:glenn@aaa.lucent.com">glenn@aaa.lucent.com</a>
	John Loughney
	Nokia Research Center
	955 Page Mill Road
	Palo Alto, CA 94304
	US
	Phone: 1-650-283-8068
	Email: <a href="mailto:john.loughney@nokia.com">john.loughney@nokia.com</a>

---

Copyright © The IETF Trust (2008).

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, THE IETF TRUST 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](mailto:ietf-ipr@ietf.org).