Network Working Group
Request for Comments: 2924
Category: Informational

N. Brownlee
The University of Auckland
A. Blount
MetraTech Corp.

September 2000

Accounting Attributes and Record Formats

Status of this Memo

Copyright Notice

Abstract

This document summarises Internet Engineering Task Force (IETF) and International Telecommunication Union (ITU-T) documents related to Accounting. A classification scheme for the Accounting Attributes in the summarised documents is presented. Exchange formats for Accounting data records are discussed, as are advantages and disadvantages of integrated versus separate record formats and transport protocols. This document discusses service definition independence, extensibility, and versioning. Compound service definition capabilities are described.

Table of Contents

1. Introduction

This document summarises IETF and ITU-T documents related to Accounting. For those documents which describe Accounting Attributes (i.e. quantities which can be measured and reported), an Attribute Summary is given. Although several of the documents describe Attributes which are similar, no attempt is made to identify those which are the same in several documents. An extensible classification scheme for AAA Accounting Attributes is proposed; it is a superset of the attributes in all the documents summarised.

Many existing accounting record formats and protocols [RAD-ACT] [TIPHON] are of limited use due to their single-service descriptive facilities and lack of extensibility. While some record formats and protocols support extensible attributes [RAD-ACT], none provide identification, type checking, or versioning support for defined groupings of attributes (service definitions). This document makes a case for well-defined services.

Advantages and disadvantages of integrated versus separate record formats and transport protocols are discussed. This document discusses service definition independence, extensibility, and versioning. Compound service definition capabilities are described.

2. Terminology and Notation

The following terms are used throughout the document.

Accounting Server
: A network element that accepts Usage Events from Service Elements. It acts as an interface to back-end rating, billing, and operations support systems.

Attribute-Value Pair (AVP)
: A representation for a Usage Attribute consisting of the name of the Attribute and a value.

Property
: A component of a Usage Event. A Usage Event describing a phone call, for instance, might have a "duration" Property.

Service
: A type of task that is performed by a Service Element for a Service Consumer.

Service Consumer
: Client of a Service Element. End-user of a network service.

Service Definition
: A specification for a particular service. It is composed of a name or other identifier, versioning information, and a collection of Properties.

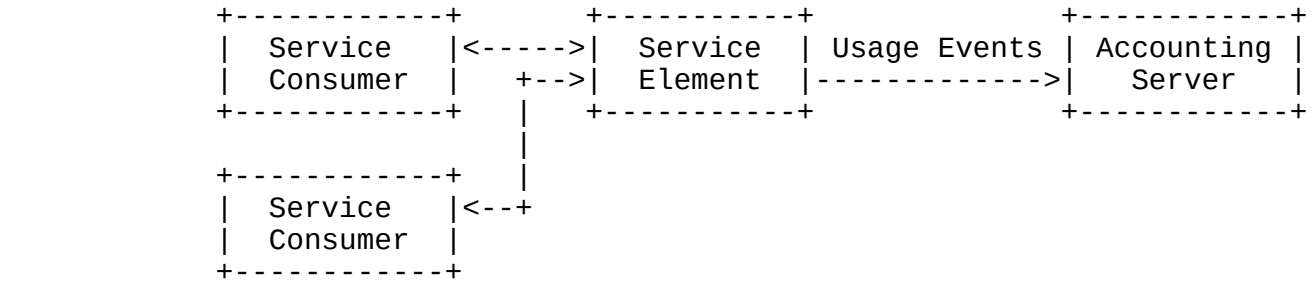
Service Element
: A network element that provides a service to Service Consumers. Examples include RAS devices, voice and fax gateways, conference bridges.

Usage Attribute
A component of a Usage Event that describes some metric of service usage.

Usage Event
The description of an instance of service usage.

3. Architecture Model

Service Elements provide Services to Service Consumers. Before, while, and/or after services are provided, the Service Element reports Usage Events to an Accounting Server. Alternately, the Accounting Server may query the Service Element for Usage Events. Usage events are sent singly or in bulk.



Accounting Servers may forward Usage Events to other systems, possibly in other administrative domains. These transfers are not addressed by this document.

4. IETF Documents

In March 1999 there were at least 19 Internet Drafts and 8 RFCs concerned with Accounting. These are summarised (by working group) in the following sections.

4.1. RADIUS

The RADIUS protocol [RAD-PROT] carries authentication, authorization and configuration information between a Network Access Server (NAS) and an authentication server. Requests and responses carried by the protocol are expressed in terms of RADIUS attributes such as User-Name, Service-Type, and so on. These attributes provide the information needed by a RADIUS server to authenticate users and to establish authorized network service for them.

The protocol was extended to carry accounting information between a NAS and a shared accounting server. This was achieved by defining a set of RADIUS accounting attributes [RAD-ACT].

RADIUS packets have a short header containing the RADIUS packet type and authenticator (sixteen octets) and length, followed by a sequence of (Type, Length, Value) triples, one for each attribute.

RADIUS is very widely used, and a number of significant new extensions to it have been proposed. For example [RAD-EXT] discusses extensions to implement the Extensible Authentication Protocol (EAP) and the Apple Remote Access Protocol (ARAP). [RAD-TACC] discusses extensions to permit RADIUS to interwork effectively with tunnels using protocols such as PPTP and L2TP.

4.1.1. RADIUS Attributes

Each RADIUS attribute is identified by an 8-bit number, referred to as the RADIUS Type field. Up-to-date values of this field are specified in the most recent Assigned Numbers RFC [ASG-NBR], but the current list is as follows:

RADIUS Attributes [RAD-PROT]

1 User-Name
2 User-Password
3 CHAP-Password
4 NAS-IP-Address
5 NAS-Port
6 Service-Type
7 Framed-Protocol
8 Framed-IP-Address
9 Framed-IP-Netmask
10 Framed-Routing
11 Filter-Id
12 Framed-MTU
13 Framed-Compression
14 Login-IP-Host
15 Login-Service
16 Login-TCP-Port
17 (unassigned)
18 Reply-Message
19 Callback-Number
20 Callback-Id
21 (unassigned)
22 Framed-Route
23 Framed-IPX-Network
24 State
25 Class
26 Vendor-Specific
27 Session-Timeout
28 Idle-Timeout
36 Login-LAT-Group
37 Framed-AppleTalk-Link
38 Framed-AppleTalk-Network
39 Framed-AppleTalk-Zone
60 CHAP-Challenge
61 NAS-Port-Type
62 Port-Limit
63 Login-LAT-Port

RADIUS Accounting Attributes [RAD-ACT]

40 Acct-Status-Type
41 Acct-Delay-Time
42 Acct-Input-Octets
43 Acct-Output-Octets
44 Acct-Session-Id
45 Acct-Authentic
46 Acct-Session-Time
47 Acct-Input-Packets
48 Acct-Output-Packets
49 Acct-Terminate-Cause
50 Acct-Multi-Session-Id
51 Acct-Link-Count

RADIUS Extension Attributes [RAD-EXT]

52 Acct-Input-Gigawords

29 Termination-Action
30 Called-Station-Id
31 Calling-Station-Id
32 NAS-Identifier
33 Proxy-State
34 Login-LAT-Service
35 Login-LAT-Node
53 Acct-Output-Gigawords
54 Unused
55 Event-Timestamp

70 ARAP-Password
71 ARAP-Features
72 ARAP-Zone-Access
73 ARAP-Security
74 ARAP-Security-Data
75 Password-Retry
76 Prompt
77 Connect-Info
78 Configuration-Token
79 EAP-Message
80 Message-Authenticator

84 ARAP-Challenge-Response
85 Acct-Interim-Interval
87 NAS-Port-Id
88 Framed-Pool

RADIUS Tunneling Attributes
[RAD-TACC]

64 Tunnel-Type
65 Tunnel-Medium-Type
66 Tunnel-Client-Endpoint
67 Tunnel-Server-Endpoint
68 Acct-Tunnel-Connection
69 Tunnel-Password

81 Tunnel-Private-Group-ID
82 Tunnel-Assignment-ID
83 Tunnel-Preference

90 Tunnel-Client-Auth-ID
91 Tunnel-Server-Auth-ID

4.2. DIAMETER

The DIAMETER framework [DIAM-FRAM] defines a policy protocol used by clients to perform Policy, AAA and Resource Control. This allows a single server to handle policies for many services. The DIAMETER protocol consists of a header followed by objects. Each object is encapsulated in a header known as an Attribute-Value Pair (AVP).

DIAMETER defines a base protocol that specifies the header formats, security extensions and requirements as well as a small number of mandatory commands and AVPs. A new service can extend DIAMETER by extending the base protocol to support new functionality.

One key differentiator with DIAMETER is its inherent support for Inter-Server communication. Although this can be achieved in a variety of ways, the most useful feature is the ability to "proxy" messages across a set of DIAMETER servers (known as a proxy chain).

The DIAMETER Accounting Extension document [DIAM-ACT] extends DIAMETER by defining a protocol for securely transferring accounting records over the DIAMETER base protocol. This includes the case where accounting records may be passed through one or more intermediate proxies, in accordance with the 'referral broker' model.

The DIAMETER accounting protocol [DIAM-ACT] defines DIAMETER records for transferring an ADIF record (see below). It introduces five new attributes (480..485) which specify the way in which accounting information is to be delivered between DIAMETER servers.

4.2.1. DIAMETER Attributes

DIAMETER AVPs are identified by a 16-bit number defined in [DIAM-AUTH]. Since most of the AVPs found in that document were copied from the RADIUS protocol [RAD-PROT], it is possible to have both RADIUS and DIAMETER servers read the same dictionary and users files.

The backward compatibility that DIAMETER offers is intended to facilitate deployment. To this end, DIAMETER inherits the RADIUS attributes, and adds only a few of its own.

In the list below attribute numbers which are used for RADIUS attributes but not for DIAMETER are indicated with a star (*). RADIUS attributes used by DIAMETER are not listed again here.

The DIAMETER attributes are:

```
   4      (unassigned, *)
  17      (unassigned)
  21      (unassigned)
  24      (unassigned, *)
  25      (unassigned, *)
  27      (unassigned, *)
  32      (unassigned, *)
  33      (unassigned, *)
 280      Filter-Rule
 281      Framed-Password-Policy
```

480	Accounting-Record-Type
481	ADIF-Record
482	Accounting-Interim-Interval
483	Accounting-Delivery-Max-Batch
484	Accounting-Delivery-Max-Delay
485	Accounting-Record-Number
600	SIP-Sequence
601	SIP-Call-ID
602	SIP-To
603	SIP-From

4.3. ROAMOPS

[ROAM-IMPL] reviews the design and functionality of existing roaming implementations. "Roaming capability" may be loosely defined as the ability to use any one of multiple Internet service providers (ISPs), while maintaining a formal customer-vendor relationship with only one. One requirement for successful roaming is the provision of effective accounting.

[ROAM-ADIF] proposes a standard accounting record format, the Accounting Data Interchange Format (ADIF), which is designed to compactly represent accounting data in a protocol-independent manner. As a result, ADIF may be used to represent accounting data from any protocol using attribute value pairs (AVPs) or variable bindings.

ADIF does not define accounting attributes of its own. Instead, it gives examples of accounting records using the RADIUS accounting attributes.

4.4. RTFM

The RTFM Architecture [RTFM-ARC] provides a general method of measuring network traffic flows between "metered traffic groups". Each RTFM flow has a set of "address" attributes, which define the traffic groups at each of the flow’s end-points.

As well as address attributes, each flow has traffic-related attributes, e.g. times of first and last packets, counts for packets and bytes in each direction.

RTFM flow measurements are made by RTFM meters [RTFM-MIB] and collected by RTFM meter readers using SNMP. The MIB uses a "DataPackage" convention, which specifies the attribute values to be read from a flow table row. The meter returns the values for each

required attribute within a BER-encoded sequence. This means there is only one object identifier for the whole sequence, greatly reducing the number of bytes required to retrieve the data.

4.4.1. RTFM Attributes

RTFM attributes are identified by a 16-bit attribute number.

The RTFM Attributes are:

0	Null		
1	Flow Subscript	Integer	Flow table info
4	Source Interface	Integer	Source Address
5	Source Adjacent Type	Integer	
6	Source Adjacent Address	String	
7	Source Adjacent Mask	String	
8	Source Peer Type	Integer	
9	Source Peer Address	String	
10	Source Peer Mask	String	
11	Source Trans Type	Integer	
12	Source Trans Address	String	
13	Source Trans Mask	String	
14	Destination Interface	Integer	Destination Address
15	Destination Adjacent Type	Integer	
16	Destination Adjacent Address	String	
17	Destination AdjacentMask	String	
18	Destination PeerType	Integer	
19	Destination PeerAddress	String	
20	Destination PeerMask	String	
21	Destination TransType	Integer	
22	Destination TransAddress	String	
23	Destination TransMask	String	
26	Rule Set Number	Integer	Meter attribute
27	Forward Bytes	Integer	Source-to-Dest counters
28	Forward Packets	Integer	
29	Reverse Bytes	Integer	Dest-to-Source counters
30	Reverse Packets	Integer	
31	First Time	Timestamp	Activity times
32	Last Active Time	Timestamp	
33	Source Subscriber ID	String	Session attributes
34	Destination Subscriber ID	String	
35	Session ID	String	

36	Source Class	Integer	"Computed" attributes
37	Destination Class	Integer	
38	Flow Class	Integer	
39	Source Kind	Integer	
40	Destination Kind	Integer	
41	Flow Kind	Integer	
50	MatchingStoD	Integer	PME variable
51	v1	Integer	Meter Variables
52	v2	Integer	
53	v3	Integer	
54	v4	Integer	
55	v5	Integer	

65-127 "Extended" attributes
(to be defined by the RTFM working group)

4.5. ISDN MIB

The ISDN MIB [ISDN-MIB] defines a minimal set of managed objects for SNMP-based management of ISDN terminal interfaces. It does not explicitly define anything related to accounting, however it does define isdnBearerChargedUnits as

> The number of charged units for the current or last connection. For incoming calls or if charging information is not supplied by the switch, the value of this object is zero.

This allows for an ISDN switch to convert its traffic flow data (such as Call Connect Time) into charging data.

4.5.1. ISDN Attributes

The relevant object in the MIB is the ISDN bearer table, which has entries in the following form:

```
IsdnBearerEntry ::=
    SEQUENCE {
        isdnBearerChannelType          INTEGER,
        isdnBearerOperStatus           INTEGER,
        isdnBearerChannelNumber        INTEGER,
        isdnBearerPeerAddress          DisplayString,
        isdnBearerPeerSubAddress       DisplayString,
        isdnBearerCallOrigin           INTEGER,
        isdnBearerInfoType             INTEGER,
        isdnBearerMultirate            TruthValue,
        isdnBearerCallSetupTime        TimeStamp,
```

```
        isdnBearerCallConnectTime       TimeStamp,
        isdnBearerChargedUnits          Gauge32
        }
```

4.6. AToMMIB

The "ATM Accounting Information MIB" document [ATM-ACT] describes a large set of accounting objects for ATM connections. An administrator may select objects from this set using a selector of the form (subtree, list) where "subtree" specifies an object identifier from the AToMMIB. For each subtree there is a table holding values for each ATM connection. The required connections are indicated by setting bits in "list", which is an octet string. For example, the set containing the number of received cells for the first eight ATM connections would be selected by (atmAcctngReceivedCells, 0xFF).

The Connection-Oriented Accounting MIB document [ATM-COLL] defines a MIB providing managed objects used for controlling the collection and storage of accounting information for connection-oriented networks such as ATM. The accounting data is collected into files for later retrieval via a file transfer protocol. Records within an accounting file are stored as BER strings [ASN1, BER].

4.6.1. AToMMIB Attributes

Accounting data objects within the AToMMBIB are identified by the last integer in their object identifiers.

The ATM accounting data objects are:

```
 1   atmAcctngConnectionType
 2   atmAcctngCastType
 3   atmAcctngIfName
 4   atmAcctngIfAlias
 5   atmAcctngVpi
 6   atmAcctngVci
 7   atmAcctngCallingParty
 8   atmAcctngCalledParty
 9   atmAcctngCallReference
10   atmAcctngStartTime
11   atmAcctngCollectionTime
12   atmAcctngCollectMode
13   atmAcctngReleaseCause
14   atmAcctngServiceCategory
15   atmAcctngTransmittedCells
16   atmAcctngTransmittedClp0Cells
17   atmAcctngReceivedCells
```

```
18   atmAcctngReceivedClp0Cells
19   atmAcctngTransmitTrafficDescriptorType
20   atmAcctngTransmitTrafficDescriptorParam1
21   atmAcctngTransmitTrafficDescriptorParam2
22   atmAcctngTransmitTrafficDescriptorParam3
23   atmAcctngTransmitTrafficDescriptorParam4
24   atmAcctngTransmitTrafficDescriptorParam5
25   atmAcctngReceiveTrafficDescriptorType
26   atmAcctngReceiveTrafficDescriptorParam1
27   atmAcctngReceiveTrafficDescriptorParam2
28   atmAcctngReceiveTrafficDescriptorParam3
29   atmAcctngReceiveTrafficDescriptorParam4
30   atmAcctngReceiveTrafficDescriptorParam5
31   atmAcctngCallingPartySubAddress
32   atmAcctngCalledPartySubAddress
33   atmAcctngRecordCrc16
```

4.7. QoS: RSVP and DIFFSERV

As we move towards providing more than simple "best effort" connectivity, there has been a tremendous surge of interest in (and work on) protocols to provide managed Quality of Service for Internet sessions. This is of particular interest for the provision of "Integrated Services", i.e. the transport of audio, video, real-time, and classical data traffic within a single network infrastructure.

Two approaches to this have emerged so far:

- the Integrated Services architecture (intserv) [IIS-ARC], with its accompanying signaling protocol, RSVP [RSVP-ARC], and RSVP's Common Open Policy Service protocol, COPS [RAP-COPS]

- the Differentiated Services architecture (diffserv) [DSRV-ARC]

RSVP is a signaling protocol that applications may use to request resources from the network. The network responds by explicitly admitting or rejecting RSVP requests. Certain applications that have quantifiable resource requirements express these requirements using intserv parameters [IIS-SPEC].

Diffserv networks classify packets into one of a small number of aggregated flows or "classes", based on the diffserv codepoint (DSCP) in the packet's IP header. At each diffserv router, packets are subjected to a "per-hop behavior" (PHB), which is invoked by the DSCP. Since RSVP is purely a requirements signalling protocol it can also be used to request connections from a diffserv network [RS-DS-OP].

4.7.1. RSVP and DIFFSERV Attributes

A set of parameters for specifying a requested Quality of Service are given in [IIS-SPEC]. These have been turned into accounting attributes within RTFM [RTFM-NEWA] and within the RSVP MIB [RSVP-MIB].

The RTFM QoS attributes are:

98	QoSService
99	QoSStyle
100	QoSRate
101	QoSSlackTerm
102	QoSTokenBucketRate
103	QoSTokenBucketSize
104	QoSPeakDataRate
105	QoSMinPolicedUnit
106	QoSMaxPolicedUnit

The RSVP MIB contains a large number of objects, arranged within the following sections:

General Objects
Session Statistics Table
Session Sender Table
Reservation Requests Received Table
Reservation Requests Forwarded Table
RSVP Interface Attributes Table
RSVP Neighbor Table

The Session tables contain information such as the numbers of senders and receivers for each session, while the Reservation Requests tables contain details of requests handled by the RSVP router. There are too many objects to list here, but many of them could be used for accounting. In particular, RSVP Requests contain the specification of the service parameters requested by a user; these, together with the actual usage data for the connection make up an accounting record for that usage.

5. ITU-T Documents

5.1. Q.825: Call Detail Recording

ITU-T Recommendation Q.825 specifies how CDRs (Call Detail Records) are produced and managed in Network Elements for POTS, ISDN and IN (Intelligent Networks).

Uses of Call Detail information for various purposes are discussed.

Each call produces one or more records describing events that occurred during the life of a call. Data may be produced in real time (single CDRs), near real-time (blocks of CDRs), or as batch files of CDRs.

The information model for Call Detail Recording is formally described in terms of an Entity-Relationship model, and an object model specified in terms of GDMO templates (Guidelines for the Definition of Managed Objects). Note that this model includes the ways in which CDRs are transported from the (NE) Network Element where they are generated to the OS (Operations System) where they are used.

5.2. Q.825 Attributes

The following attributes are defined. The explanations given are very brief summaries only, see [Q-825] for the complete text.

1 accessDelivery
Indicates that the call was delivered to the called subscriber

2 accountCodeInput
Account code (for billing), supplied by subscriber.

78 additionalParticipantInfo
(No details given)

5 b-PartyCategory
Subscriber category for called subscriber.

4 bearerService
Bearer capability information (only for ISDN calls).

13 cDRPurpose
Reason for triggering this Call Data Record.

70 callDetailDataId
Unique identifier for the CallDetailData object.

79 callDuration
Duration of call

6 callIdentificationNumber
Identification number for call; all records produced for this call have the same callIdenfificationNumber.

73 callStatus
Identifies whether the call was answered or not.

9 calledPartyNumber
Telephone number of the called subscriber (may be a "diverted-to" or "translated" number.

7 callingPartyCategory
Calling subscriber category.

8 callingPartyNumber
Telephone number of the calling party.

10 callingPartyNumberNotScreened
An additional, user-provided (not screened) number to the calling party.

11 callingPartyType
Calling subscriber type.

74 carrierId
Carrier ID to which the call is sent.

12 cause
Cause and location value for the termination of the call.

14 chargedDirectoryNumber
Charged directory number (where the charged participant element can’t indicate the number).

16 chargedParticipant
Participant to be charged for the usage.

15 chargingInformation
Charging information generated by a Network Element which is capable of charging.

17 configurationMask
Time consumption, e.g. from B-answer to termination time, between partial call records, etc.

18 conversationTime
Time consumption from B-answer to end of call.

19 creationTriggerList
List of trigger values which will create Call Detail data objects.

75 dPC
Destination point code (for analysis purposes).

20 dataValidity
Indicates that the NE is having problems, contents of the generated Call Detail record is not reliable.

23 durationTimeACM
Time consumption from seizure until received ACM.

21 durationTimeB-Answer
Time consumption from seizure until B-answer.

22 durationTimeNoB-Answer
Time from seizure to termination when no B-answer was received.

25 exchangeInfo
Identity of exchange where Call Detail record was generated.

26 fallbackBearerService
Fallback bearer capability information for a call.

27 glare
Indicates if a glare condition was encountered.

31 iNServiceInformationList
Contains information about the use of IN (Intelligent Network) services.

32 iNSpecificInformation
Contains information about the use of one IN service.

33 iSUPPreferred
Indicate whether an ISUP preference was requested.

28 immediateNotificationForUsageMetering
Indicates that the Call Detail records requires immediate data transfer to the Operations System.

34 maxBlockSize
Maximum number of Call Detail records in a block.

35 maxTimeInterval
Maximum latency allowable for near-real-time Call Detail data delivery.

36 networkManagementControls
Indicates which Traffic Management Control has affected the call.

37 networkProviderId
Indicates the Network Provider for whom the CDR is generated.

76 oPC
Originating point code for a failed call (for analysis purposes).

38 operatorSpecific1AdditionalNumber
40 operatorSpecific2AdditionalNumber
42 operatorSpecific3AdditionalNumber
Operator-defined additional participant information.

39 operatorSpecific1Number
41 operatorSpecific2Number
43 operatorSpecific3Number
Operator-defined participant information.

44 originalCalledNumber
Telephone number of the original called party.

45 partialGeneration
Included if the CDR (Call Detail record) output is partial. Such CDRs have a field indicating their partial record number.

77 participantInfo
(No details given).

46 percentageToBeBilled
Percentage to be billed when normal billing rules are not to be followed.

47 periodicTrigger
Defines the intervals at which the CDR file should be created.

48 personalUserId
Internationally unique personal User Identity (for UPT calls).

49 physicalLineCode
Identifies the call subscriber’s physical line.

50 progress
Describes an event which occurred during the life of a call.

51 queueInfo
Used to record usage of queueing resources with IN calls.

52 receivedDigits
The digits dialed by the subscriber. (Normally only included for customer care purposes).

53 recordExtensions
Information elements added by network operators and/or manufacturers in addition to the standard ones above.

6. Other Documents

6.1. TIPHON: ETSI TS 101 321

TIPHON [TIPHON] is an XML-based protocol, carried by HTTP, which handles accounting and authorization requests and responses.

The following are elements selected from TIPHON's DTD that are used for accounting.

<!ELEMENT Currency (#PCDATA)> <!ELEMENT Amount (#PCDATA)>
Identifies a numeric value. Expressed using the period (.) as a decimal separator with no punctuation as the thousands separator.

<!ELEMENT CallId (#PCDATA)>
Contains a call's H.323 CallID value, and is thus used to uniquely identify individual calls.

<!ELEMENT Currency (#PCDATA)>
Defines the financial currency in use for the parent element.

<!ELEMENT DestinationInfo type (e164 | h323 | url | email |
transport | international |
national | network | subscriber |
abbreviated | e164prefix)
Gives the primary identification of the destination for a call.

<!ELEMENT Increment (#PCDATA)>
Indicates the number of units being accounted.

<!ELEMENT Service EMPTY>
Indicates a type of service being priced, authorized, or reported. An empty Service element indicates basic Internet telephony service, which is the only service type defined by V1.4.2 of the specification. The specification notes that "Later revisions of this standard are expected to specify more enhanced service definitions to represent quality of service, availability, payment methods, etc."

```
<!ELEMENT DestinationInfo type ( e164 | h323 | url | email |
                                 transport | international |
                                 national | network | subscriber |
                                 abbreviated | e164prefix)
    Gives the primary identification of the source of a call.


<!ELEMENT Timestamp (#PCDATA)>
    A restricted form of [ISO-DATE] that indicates the time at which
    the component was generated.

<!ELEMENT TransactionId (#PCDATA)>
    Contains an integer, decimal valued identifier assigned to a
    specific authorized transaction.

<!ELEMENT Unit (#PCDATA)>
    Indicates the units by which pricing is measured or usage
    recorded.  It shall contain one of the following values:
        s      seconds
        p      packets (datagrams)
        byte   bytes

<!Element UsageDetail ( Service, Amount, Increment, Unit ) >
    Collects information describing the usage of a service.
```

6.2. MSIX

MSIX [MSIX-SPEC] is an XML-based protocol transported by HTTP that is used to make accounting service definitions and transmit service usage information. As its service definitions are parameterized and dynamic, it makes no definition of services or attributes itself, but allows implementors to make their own. It specifies only the base data types that attributes may take: STRING, UNISTRING, INT32, FLOAT, DOUBLE, BOOLEAN, TIMESTAMP.

7. Accounting File and Record Formats

7.1. ASN.1 Records

7.1.1. RTFM and AToMMIB

RTFM and AToMMIB use ASN.1 Basic Encoding Rules (BER) to encode lists of attributes into accounting records. RTFM uses SNMP to retrieve such records as BER strings, thus avoiding having to have an object identifier for every object.

AToMMIB carries this a stage further by defining an accounting file format in ASN.1 and making it available for retrieval by a file transfer protocol, thereby providing a more efficient alternative to simply retrieving the records using SNMP.

7.1.2. Q.825

A Q.825 Call Record is an ASN.1 SET containing a specified group of the Q.825 attributes. Call records would presumably be encoded as BER strings before being collected for later processing.

7.2. Binary Records

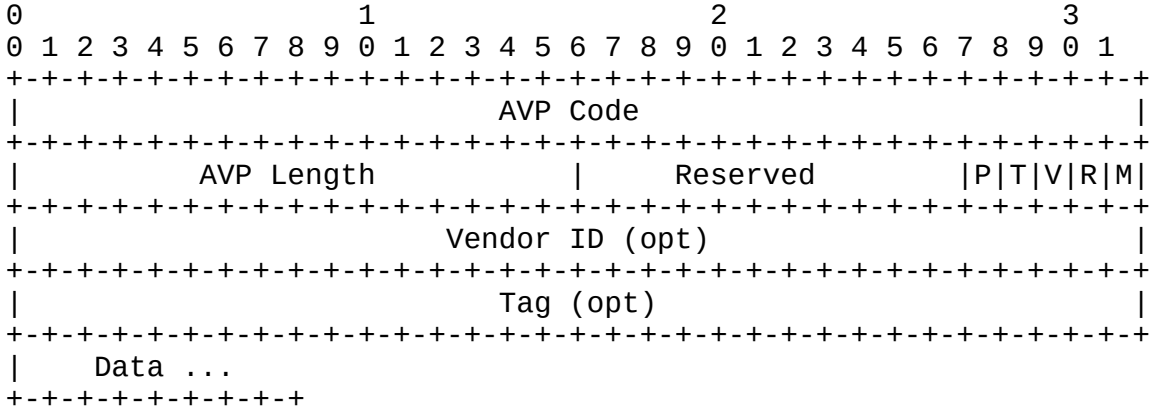
7.2.1. RADIUS

Radius packets carry a sequence of attributes and their values, as (Type, Length, Value) triples. The format of the value field is one of four data types.

string	0-253 octets
address	32 bit value, most significant octet first.
integer	32 bit value, most significant octet first.
time	32 bit value, most significant octet first -- seconds since 00:00:00 GMT, January 1, 1970. The standard Attributes do not use this data type but it is presented here for possible use within Vendor-Specific attributes.

7.2.2. DIAMETER

Each DIAMETER message consists of multiple AVP's that are 32-bit aligned, with the following format:



Code
The AVP Code identifies the attribute uniquely. If the Vendor-Specific bit is set, the AVP Code is allocated from the vendor's private address space.

The first 256 AVP numbers are reserved for backward compatibility with RADIUS and are to be interpreted as per RADIUS [RAD-PROT]. AVP numbers 256 and above are used for DIAMETER, which are allocated by IANA.

AVP Length
A 16-bit field contains the total object length in bytes. Must always be a multiple of 4, and at least 8.

AVP Flags

P	Protected bit
T	Tag bit
V	Vendor-ID bit
R	Reserved (MUST be set to 0)
M	Mandatory bit

7.3. Text Records

7.3.1. ROAMOPS

ADIF (Accounting Data Interchange Format [ROAM-ADIF]) presents a general, text-based format for accounting data files, described in a straightforward BNF grammar. Its file header contains a field indicating the default protocol from which accounting attributes are drawn. If an attribute from another protocol is to be used, it is preceded by its protocol name, for example rtfm//27 would be RTFM's "forward bytes" attribute. Comments in an ADIF file begin with a cross-hatch.

Example: An ADIF file encoding RADIUS accounting data

```
version: 1
device: server3
description: Accounting Server 3
date: 02 Mar 1999 12:19:01 -0500
defaultProtocol: radius

rdate: 02 Mar 1999 12:20:17 -0500
#NAS-IP-Address
4: 204.45.34.12
#NAS-Port
5: 12
#NAS-Port-Type
```

```
61: 2
#User-Name
1: fred@bigco.com
#Acct-Status-Type
40: 2
#Acct-Delay-Time
41: 14
#Acct-Input-Octets
42: 234732
#Acct-Output-Octets
43: 15439
#Acct-Session-Id
44: 185
#Acct-Authentic
45: 1
#Acct-Session-Time
46: 1238
#Acct-Input-Packets
47: 153
#Acct-Output-Packets
48: 148
#Acct-Terminate-Cause
49: 11
#Acct-Multi-Session-Id
50: 73
#Acct-Link-Count
51: 2
```

8. AAA Requirements

8.1. A Well-Defined Set of Attributes

AAA needs a well-defined set of attributes whose values are to be carried in records to or from accounting servers.

Most of the existing sets of documents described above include a set of attributes, identified by small integers. It is likely that these sets overlap, i.e. that some of them have attributes which represent the same quantity using different names in different sets. This suggests it might be possible to produce a single combined set of "universal" accounting attributes, but such a "universal" set does not seem worthwhile.

The ADIF approach of specifying a default protocol (from which attributes are assumed to come) and identifying any exceptions seems much more practical. We therefore propose that AAA should use the

ADIF convention (or something like it) to identify attributes, together with all the sets of attributes covered by the [ASG-NBR] document.

8.2. A Simple Interchange Format

AAA needs a simple interchange file format, to be used for accounting data. Several schemes for packaging and transporting such data have been described above.

The SNMP-based ones fit well within the context of an SNMP-based network management system. RTFM and AToMMIB provide ways to reduce the SNMP overhead for collecting data, and AToMMIB defines a complete file format. Both provide good ways to collect accounting data.

As an interchange format, however, ASN.1-based schemes suffer from being rather complex binary structures. This means that one requires suitable tools to work with them, as compared to plain-text files where one can use existing text-based utilities.

The binary schemes such as RADIUS and DIAMETER have simpler structures, but they too need purpose-built tools. For general use they would need to be extended to allow them to use attributes from other protocols.

From the point of view of being easy for humans to understand, ADIF seems very promising. Of course any processing program would need a suitable ADIF input parser, but using plain-text files makes them much easier to understand.

TIPHON's record format is specified by an XML DTD. While XML representations have the advantages of being well-known, they are limited by XML's inability to specify type or other validity checking for information within the tags. This situation will likely be improved by the XML Schema [XML-SCHM] efforts that are underway, but a stable reference is not yet available.

9. Issues

It is generally agreed that there is a need for a standard record format and transport protocol for communication between Service Elements and Accounting Servers.

There is less agreement on the following issues:

- Separate or integral record format and transport protocol
- Standard set of base data types
- Service definitions: part of the protocol or separately defined

o Service definition namespace management

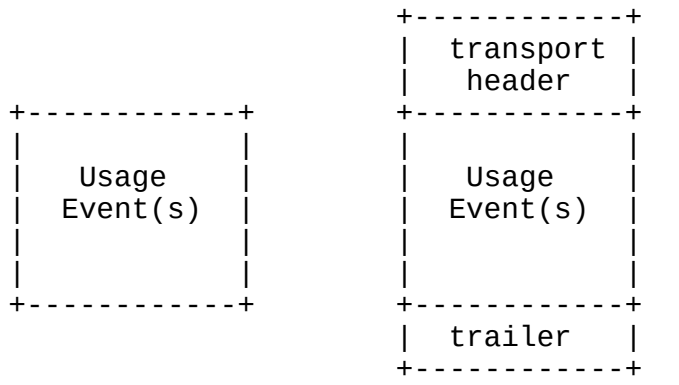
The following sections address these issues.

9.1. Record Format vs. Protocol

All known Internet-centric billing protocols to date have an integral record format. That is, the collection of Properties that describe a Usage Event are specified as an integral part of the protocol, typically as a part of a "submit" message that is used to transmit a Usage Event from a Service Entity to an Accounting Server.

It may be advantageous to define a record format that is independent of the transport protocol. Such a record format should support both representation of individual records and records in bulk, as Usage Events are often aggregated and transmitted in bulk.

A separate record format is useful for record archiving and temporary file storage. Multiple transport protocols may be defined without affecting the record format. The task of auditing is made easier if a standard file format is defined. If a canonical format is used, bulk records may be hashed with MD5 [MD5] or a similar function, for reliability and security purposes.



If the protocol is written such that it can transmit Usage Events in the record format, no record rewriting for transport is required.

9.2. Tagged, Typed Data

Record formats and protocols use a combination of data locality and explicit tagging to identify data elements. Mail [RFC822], for instance, defines a header block composed of several Attribute-Value Pairs, followed by a message body. Each header field is explicitly

tagged, but the order of the AVPs is undefined. The message body is not tagged (except with an additional preceding blank line), and is found through its position in the message, which must be after all header fields.

Some record formats make no use of tags--data elements are identified only by their position within a record structure. While this practice provides for the least amount of record space overhead, it is difficult to later modify the record format by adding or removing elements, as all record readers will have to be altered to handle the change. Tagged data allows old readers to detect unexpected tags and to detect if required data are missing. If the overhead of carrying explicit tags can be borne, it is advantageous to use explicitly tagged data elements where possible.

An AVP approach has proven useful in accounting. RADIUS [RADIUS] uses numeric data type identifiers. ETSI's TIPHON [TIPHON] uses XML markup.

For an AAA accounting record format, the authors suggest that each Property be named by a textual or numeric identifier and carry a value and a data type indicator, which governs interpretation of the value. It may also be useful for each Property to carry a units of measure identifier. The TIPHON specification takes this approach. TS 101 321 also carries an Increment field, which denominates the Property's Unit of Measure field. Whether this additional convenience is necessary is a matter for discussion.

It is not strictly necessary for each data record to carry data type, units of measure, or increments identifiers. If this information is recorded in a record schema document that is referenced by each data record, each record may be validated against the schema without the overhead of carrying type information.

9.2.1. Standard Type Definitions

It is useful to define a standard set of primitive data types to be used by the record format and protocol. Looking at the prior art, DIAMETER supports Data (arbitrary octets), String (UTF-8), Address (32 or 128 bit), Integer32, Integer64, Time (32 bits, seconds since 1970), and Complex. MSIX [MSIX-SPEC] supports String, Unistring, Int32, Float, Double, Boolean, and Timestamp. SMIv2 [SMI-V2] offers ASN.1 types INTEGER, OCTET STRING, and OBJECT IDENTIFIER, and the application-defined types Integer32, IpAddress, Counter32, Gauge32, Unsigned32, TimeTicks, Opaque, and Counter64.

An appropriate set would likely include booleans, 32 and 64 bit signed integers, 32 and 64 bit floats, arbitrary octets, UTF-8 and UTF-16 strings, and ISO 8601:1988 [ISO-DATE] timestamps. Fixed-precision numbers capable of representing currency amounts (with precision specified on both sides of the decimal point) have proven useful in accounting record formats, as they are immune to the precision problems that are encountered when one attempts to represent fixed-point amounts with floating point numbers.

It may be worthwhile to consider the datatypes that are being specified by the W3C's "XML Schema Part 2: Datatypes" [XML-DATA] document. That document specifies a rich set of base types, along with a mechanism to specify derivations that further constrain the base types.

9.3. Transaction Identifiers

Each Usage Event requires its own unique identifier.

It is expedient to allow Service Elements to create their own unique identifiers. In this manner, Usage Events can be created and archived without the involvement of an Accounting Server or other central authority.

A number of methods for creating unique identifiers are well known. One popular identifier is an amalgamation of a monotonically increasing sequence number, a large random value, a network element identifier, and a timestamp. Another possible source of entropy is a hash value of all or part of the record itself.

RFC 822 [MAIL], RFC 1036 [NEWS], and RFC 2445 [ICAL-CORE] give guidance on the creation of good unique identifiers.

9.4. Service Definitions

A critical differentiator in accounting record formats and protocols is their capability to account for arbitrary service usage. To date, no accounting record format or protocol that can handle arbitrary service definitions has achieved broad acceptance on the Internet.

This section analyzes the issues in service definition and makes a case for a record format and protocol with the capability to carry Usage Events for rich, independently-defined services.

9.4.1. Service Independence

It is informative to survey a number of popular Internet protocols and document encodings and examine their capacities for extension. These protocols can be categorized into two broad categories--"fully specified" protocols that have little provision for extension and "framework" protocols that are incomplete, but provide a basis for future extension when coupled with application documents.

Examples of fully-specified protocols are NTP [NTP], NNTP [NNTP], RADIUS Accounting [RAD-ACT], and HTML [HTML].

Aside from leaving some field values "reserved for future use", all of Network Time Protocol's fields are fixed-width and completely defined. This is appropriate for a simple protocol that solves a simple problem.

Network News Transfer Protocol [NEWS-PROT] specifies that further commands may be added, and requests that non-standard implementations use the "X-" experimental prefix so as to not conflict with future additions. The content of news is 7-bit data, with the high-order bit cleared to 0. Nothing further about the content is defined. There is no in-protocol facility for automating decoding of content type.

We pay particular attention to RADIUS Accounting [RAD-ACT]. Perhaps the second most frequently heard complaint (after security shortcomings) about RADIUS Accounting is its preassigned and fixed set of "Types". These are coded as a range of octets from 40 to 51 and are as follows:

40	Acct-Status-Type
41	Acct-Delay-Time
42	Acct-Input-Octets
43	Acct-Output-Octets
44	Acct-Session-Id
45	Acct-Authentic
46	Acct-Session-Time
47	Acct-Input-Packets
48	Acct-Output-Packets
49	Acct-Terminate-Cause
50	Acct-Multi-Session-Id
51	Acct-Link-Count

These identifiers were designed to account for packet-based network access service. They are ill-suited for describing other services. While extension documents have specified additional types, the base

protocol limits the type identifier to a single octet, limiting the total number of types to 256.

HTML/2.0 [HTML] is mostly a fully-specified protocol, but with W3C's HTML/4.0, HTML is becoming more of a framework protocol. HTML/2.0 specified a fixed set of markups, with no provision for addition (without protocol revision).

Examples of "framework" protocols and document encodings are HTTP, XML, and SNMP.

HTTP/1.1 [HTTP] is somewhat similar to NNTP in that it is designed to transport arbitrary content. It is different in that it supports description of that content through its Content-Type, Content-Encoding, Accept-Encoding, and Transfer-Encoding header fields. New types of content can be designated and carried by HTTP/1.1 without modification to the HTTP protocol.

XML [XML] is a preeminent general-purpose framework encoding. DTD publishing is left to users. There is no standard registry of DTDs.

SNMP presents a successful example of a framework protocol. SNMP's authors envisioned SNMP as a general management protocol, and allow extension through the use of private MIBs. SNMP's ASN.1 MIBs are defined, published, and standardized without the necessity to modify the SNMP standard itself. From "An Overview of SNMP" [SNMP-OVER]:

> It can easily be argued that SNMP has become prominent mainly from its ability to augment the standard set of MIB objects with new values specific for certain applications and devices. Hence, new functionality can continuously be added to SNMP, since a standard method has been defined to incorporate that functionality into SNMP devices and network managers.

Most accounting protocols are fully-specified, with either a completely defined service or set of services (RADIUS Accounting) or with one or more services defined and provision for "extension" services to be added to the protocol later (TIPHON). While the latter is preferable, it may be preferable to take a more SNMP-like approach, where the accounting record format and protocol provide only a framework for service definition, and leave the task of service definition (and standardization) to separate efforts. In this manner, the accounting protocol itself would not have to be modified to handle new services.

9.4.2. Versioned Service Definitions

Versioning is a naming and compatibility issue. Version identifiers are useful in service definition because they enable service definitions to be upgraded without a possibly awkward name change. They also enable possible compatibility between different versions of the same service.

An example could be the service definition of a phone call. Version 1 might define Properties for the start time, duration, and called and calling party numbers. Later, version 2 is defined, which augments the former service definition with a byte count. An Accounting Server, aware only of Version 1, may accept Version 2 records, discarding the additional information (forward compatibility). Alternately, if an Accounting Server is made aware of version 2, it could optionally still accept version 1 records from Service Elements, provided the Accounting Sever does not require the additional information to properly account for service usage (backward compatibility).

9.4.3. Relationships Among Usage Events

Accounting record formats and protocols to date do not sufficiently addressed "compound" service description.

A compound service is a service that is described as a composition of other services. A conference call, for example, may be described as a number of point-to-point calls to a conference bridge. It is important to account for the individual calls, rather than just summing up an aggregate, both for auditing purposes and to enable differential rating. If these calls are to be reported to the Accounting Server individually, the Usage Events require a shared identifier that can be used by the Accounting Server and other back-end systems to group the records together.

In order for a Service Element to report compound events over time as a succession of individual Usage Events, the accounting protocol requires a facility to communicate that the compound event has started and stopped. The "start" message can be implicit--the transmission of the first Usage Event will suffice. An additional semaphore is required to tell the Accounting Server that the compound service is complete and may be further processed. This is necessary to prevent the Accounting Server from prematurely processing compound events that overlap the end of a billing period.

RADIUS Accounting has some provision for this sort of accounting with its "Acct-Multi-Session-Id" field. Unfortunately, RADIUS Accounting's other shortcomings preclude it from being used in general purpose service usage description.

9.4.4. Service Namespace Management

"Framework" protocols, as previously mentioned, do not define complete schema for their payload. For interoperability to be achieved, it must be possible for:

(1) content definers to specify definitions without conflicting with the names of other definitions

(2) protocol users to find and use content definitions

Condition (1) can be readily managed through IANA assignment or by using an existing namespace differentiator (for example, DNS).

Condition (2) is harder, and places considerable burden on the implementors. Their clients and servers must be able, statically or dynamically, to find and validate definitions, and manage versioning issues.

As previously mentioned, the XML specification provides no facility for DTD discovery or namespace management. XML specifies only a document format, and as such does not need to specify support for more "protocol" oriented problems.

For an accounting record format and protocol, an approach closer to SNMP's is useful. SNMP uses an ISO-managed dotted-decimal namespace. An IANA-managed registry of service types is a possibility. Another possibility, used by MSIX [MSIX-SPEC], is for Service Element creators to identify their services by concatenation of a new service name with existing unique identifier, such as a domain name.

A standard record format for service definitions would make it possible for Service Element creators to directly supply accounting system managers with the required definitions, via the network or other means.

10. Encodings

It may be useful to define more than one record encoding.

A "verbose" XML encoding is easily implemented and records can be syntactically verified with existing tools. "Human-readable" protocols tend to have an edge on "bitfield" protocols where ease of

implementation is paramount and the application can tolerate any additional processing required to generate, parse, and transport the records.

A alternative "compressed" encoding that makes minimal use of storage and processing may be useful in many contexts.

There are disadvantages to supporting multiple encodings. Optionally-supported multiple encodings mandate the requirement for capabilities exchange between Service Element and Accounting Server. Also, implementations can tend to "drift apart", with one encoding better-supported than another. Unless all encodings are mandatory, implementors may find they are unable to interoperate because they picked the wrong encoding.

11. Security Considerations

This document summarises many existing IETF and ITU documents; please refer to the original documents for security considerations for their particular protocols.

It must be possible for the accounting protocol to be carried by a secure transport. A canonical record format is useful so that regeneration of secure record hashes is possible.

When dealing with accounting data files, one must take care that their integrity and privacy are preserved. This document, however, is only concerned with the format of such files.

12. References

[ACC-BKG] Mills, C., Hirsch, G. and G. Ruth, "Internet Accounting Background", RFC 1272, November 1991.

[ASG-NBR] Reynolds, J. and J. Postel, "Assigned Numbers", STD 2, RFC 1700, October 1994.

[ASN1] Information processing systems - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1), International Organization for Standardization, International Standard 8824, December 1987.

[ATM-ACT] McCloghrie, K., Heinanen, J., Greene, W. and A. Prasad, "Accounting Information for ATM Networks", RFC 2512, February 1999.

[ATM-COLL] McCloghrie, K., Heinanen, J., Greene, W. and A. Prasad, " Managed Objects for Controlling the Collection and Storage of Accounting Information for Connection-Oriented Networks", RFC 2513, February 1999.

[BER] Information processing systems - Open Systems Interconnection - Specification of Basic Encoding Rules for Abstract Notation One (ASN.1), International Organization for Standardization, International Standard 8825, December 1987.

[DIAM-ACT] Arkko, J., Calhoun, P.R., Patel, P. and Zorn, G., "DIAMETER Accounting Extension", Work in Progress.

[DIAM-AUTH] Calhoun, P.R. and Bulley, W., "DIAMETER User Authentication Extensions", Work in Progress.

[DIAM-FRAM] Calhoun, P.R., Zorn, G. and Pan, P., "DIAMETER Framework Document", Work in Progress.

[DSRV-ARC] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z. and W. Weiss, "An Architecture for Differentiated Services", RFC 2475, December 1998.

[HTML] Berners-Lee, T. and D. Connolly, "Hypertext Markup Language - 2.0", RFC 1866, November 1995.

[HTTP] Fielding, R., Gettys, J., Mogul, J. Frystyk, H. and T. Berners-Lee, "Hypertext Transfer Protocol--HTTP/1.1", RFC 2068, January 1997.

[ICAL-CORE] Dawson, F. and D. Stenerson, "Internet Calendaring and Scheduling Core Object Specification", RFC 2445, November 1998.

[IIS-ARC] Braden, R., Clark, D. and S. Shenker, "Integrated Services in the Internet Architecture: an Overview", RFC 1633, June 1994.

[IIS-SPEC] Shenker, S., Partridge, C. and R. Guerin, "Specification of Guaranteed Quality of Service", RFC 2212, September 1997.

[ISDN-MIB] Roeck, G., "ISDN Management Information Base using SMIv2", RFC 2127, March 1997.

[ISO-DATE] "Data elements and interchange formats -- Information interchange -- Representation of dates and times", ISO 8601:1988.

[MAIL] Crocker, D., "STANDARD FOR THE FORMAT OF ARPA INTERNET TEXT MESSAGES", STD 11, RFC 822, August 1982.

[MD5] Rivest, R., "The MD5 Message-Digest Algorithm", RFC 1321, April 1992.

[MSIX-SPEC] Blount, A. and D. Young, "Metered Service Information Exchange 1.2", Work in Progress.

[NEWS-MSGS] Horton, M. and R. Adams, "Standard for Interchange of USENET Messages", RFC 1036, December 1987.

[NEWS-PROT] Kantor, B. and P. Lapsley, "Network News Transfer Protocol", RFC 977, February 1986.

[NTP] Mills, D., "Network Time Protocol (NTP)", RFC 958, September 1985.

[Q-825] "Specification of TMN applications at the Q3 interface: Call detail recording", ITU-T Recommendation Q.825, 1998.

[RAD-ACT] Rigney, C., "RADIUS Accounting", RFC 2866, June 2000.

[RAD-EXT] Rigney, C., Willats, W. and Calhoun, P., "RADIUS Extensions", RFC 2869, June 2000.

[RAD-PROT] Rigney, C., Willens, S., Rubens, A., and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)", RFC 2865, June 2000.

[RAD-TACC] Zorn, G., Mitton, D. and A. Aboba, "RADIUS Accounting Modifications for Tunnel Protocol Support", RFC 2867, June 2000.

[RAP-COPS] Boyle, J., Cohen, R., Durham, D., Herzog, S., Rajan, R. and A. Sastry, "The COPS (Common Open Policy Service) Protocol", RFC 2748, January 2000.

[ROAM-ADIF] Aboba, B. and D. Lidyard, "The Accounting Data Interchange Format (ADIF)", Work in Progress.

[ROAM-IMPL] Aboba, B., Lu, J., Alsop, J., Ding, J. and W. Wang, "Review of Roaming Implementations", RFC 2194, September 1997.

[RS-DS-OP] Bernet, Y., Yavatkar, R., Ford, P., Baker, F., Zhang, L., Speer, M., Braden, R., Davie, B., Wroclawski, J. and E. Felstaine, "A Framework For Integrated Services Operation Over Diffserv Networks", Work in Progress.

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

[RSVP-MIB] Baker, F., Krawczyk, J. and A. Sastry, "RSVP Management Information Base using SMIv2", RFC 2206, September 1997.

[RTFM-ARC] Brownlee, N., Mills, C. and G. Ruth, "Traffic Flow Measurement: Architecture", RFC 2722, October 1999.

[RTFM-MIB] Brownlee, N., "Traffic Flow Measurement: Meter MIB", Measurement: Architecture", RFC 2720, October 1999.

[RTFM-NEWA] Handelman, S., Brownlee, N., Ruth, G. and S. Stibler, "New Attributes for Traffic Flow Measurement", RFC 2724, October 1999.

[SIP-PROT] Handley, M., Schulzrinne, H., Schooler, E. and J. Rosenberg, "SIP: Session Initiation Protocol", RFC 2543, March 1999.

[SMI-V2] McCloghrie, K., Perkins, D. and J. Schoenwaelder, "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.

[SNMP-OVER] "AN OVERVIEW OF SNMP V2.0", Diversified Data Resources, Inc., http://www.ddri.com, 1999.

[TIPHON] "Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON); Inter-domain pricing, authorization, and usage exchange", TS 101 321 V1.4.2, December 1998.

[XML] Bray, T., J. Paoli, and C. Sperberg-McQueen, "Extensible Markup Language (XML) 1.0", W3C Recommendation, February 1998.

[XML-DATA] "XML Schema Part 2: Datatypes", W3C Working Draft 07 April 2000, April 2000.

[XML-SCHM] "XML Schema Part 1: Structures", W3C Working Draft 7 April 2000, April 2000.

13. Authors' Addresses

Nevil Brownlee
Information Technology Systems & Services
The University of Auckland

Phone: +64 9 373 7599 x8941
EMail: n.brownlee@auckland.ac.nz

Alan Blount
MetraTech Corp.
330 Bear Hill Road
Waltham, MA 02451

EMail: blount@alum.mit.edu

14. Full Copyright Statement

Acknowledgement

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