Network Working Group
INTERNET-DRAFT
Category: Informational
<draft-blount-acct-service-00.txt>
15 September 1999

Alan Blount
MetraTech Corp.

Accounting Protocol and Record Format Features

Status of this Memo

Abstract

This document discusses features of Internet-centric accounting record formats and protocols. It advocates the separation of record format and transport protocol. It argues for the facility to create service definitions without modification to the base record format or protocol. "Rich" service definition capabilities, with the ability to describe compound events, are described.

Table of Contents

1. Introduction

There is no Internet Standard as of yet for general service usage accounting. RADIUS Accounting [10] perhaps comes closest, but is very limited in its applicability due to its "single service" descriptive facility and its limited extensibility to other services. This draft describes features that would be useful in a yet-to-be-defined accounting record format and protocol.

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 and billing systems.

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.

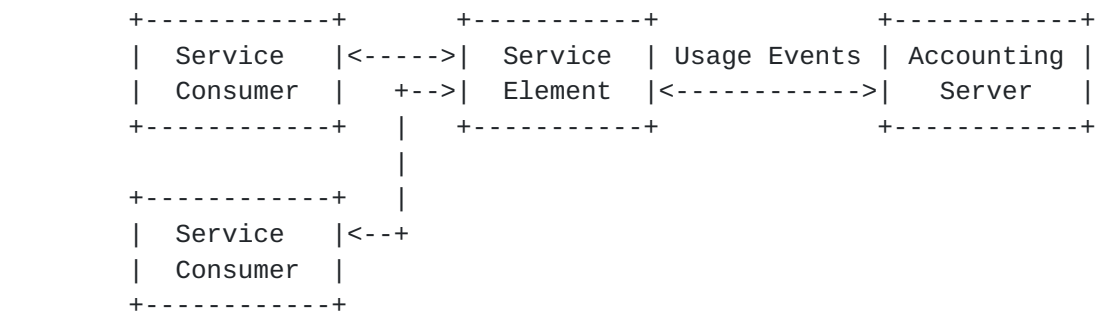
Service Element
: A network element that provides a service to Service Consumers. Examples include RAS servers, voice and fax gateways, conference bridges.

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.



4. Features

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
- Service definition namespace management

The following sections address these issues.

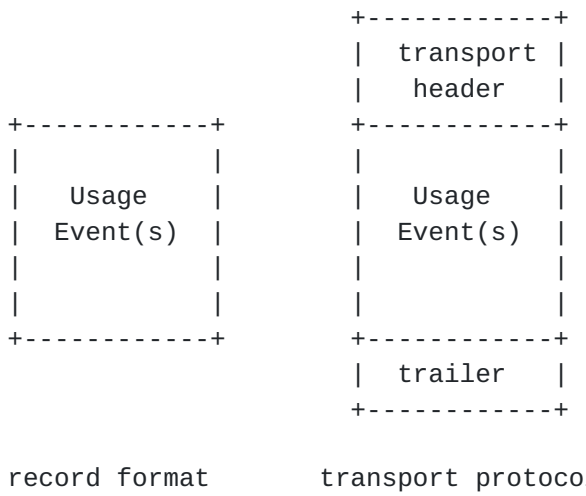
4.1 Record Format vs. Protocol

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 worthwhile 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 temporary file storage and for archival purposes. 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 [3] 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.

4.2. Tagged, Typed Data

Each Property is named by a textual identifier and carries 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. ETSI's TIPHON [11] 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.

4.2.1 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, and Time (32 bits, seconds since 1970). MSIX [6] supports String, Unistring, Int32, Float, Double, Boolean, and Timestamp. SNMP offers Text, Counter, Gauge, Integer, EnumVal, Time (elapsed only), IPAddr, PhysAd, and String.

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 [4] timestamps.

4.3 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.

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 [7], RFC 1036 [8], and RFC 2445 [9] give guidance on the creation of good unique identifiers.

4.4 Service Definitions

A critical differentiator in accounting record formats and protocols is their capability to account for arbitrary service usage. To date, no 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.

4.4.1 Service Independence

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

Examples of "complete" protocols are NTP [12], NNTP [13], RADIUS Accounting (RFC 2139), and HTML [14].

Network Time Protocol (RFC 958) defines everything. Aside from leaving some field values "reserved for future use," all the fields are fixed-width and completely defined. This is appropriate for a simple protocol that solves a simple problem. The notion of "time" is unlikely to be extended (we hope).

Network News Transfer Protocol (RFC 977) 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 (RFC 2139). Perhaps the second biggest 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.

HTML 2.0 (RFC 1866) was mostly a "complete" protocol, but with W3C's HTML 4.0, it is becoming more of a framework. HTML/2.0 specified a fixed set of markups, with no provision for addition (without protocol revision).

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

HTTP/1.1 (RFC 2068 [2]) 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 [1] is the king-hell general-purpose "framework" encoding. DTD publishing is left to users. There is no standard registry of DTDs.

SNMP also 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 or even involve the SNMP standard itself. From "An Overview of SNMP" [5]:

> 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.

"complete" protocols, with either a completely defined 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 would be better to have a more SNMP-like approach where the accounting record format and protocol merely provide a framework for service definition, and leave the task of service definition (and optionally standardization) to separate efforts. In this manner, the accounting protocol itself

would not have to be modified to handle new services.

4.4.2 Versioned Service Definitions

Versioning is a naming and compatibility issue. Versioning is useful

in service definition because it enables service definitions to be upgraded without a possibly awkward name change. It also enables 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 with a byte count. An Accounting Server, aware of only Version 1, may accept Version 2 records, discarding the additional information (forwards compatibility). Alternately, if an Accounting Server is made aware of version 2, it could optionally still accept version 1 records from Service Elements, provided it does not require the additional information to properly account for service usage (backwards compatibility).

4.4.3 Relationships Among Usage Events

A case that accounting record formats and protocols to date have failed to sufficiently address is that of "compound" service description.

A compound service is a service that is described as a composition of other services. A conference call, for example, is well 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 backend systems to group the records together.

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.

4.4.4 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.

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, XML provides no facility for DTD discovery or namespace management. XML, however, 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 [6], is for Service Element creators to identify their services by concatination 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 by other means.

4.5 Transactional Capabilities

Transactional capabilities, at least in the "ACID" sense of the word "transactional", do not seem to be a major accounting protocol requirement. The creation and transmission of usage events is a task closer to logging than to transactional database access, as there is little need for rollback.

5. Encodings

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

A "verbose" XML encoding is useful in that it 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.

Multiple encodings mandate the requirement for capabilities exchange between Service Element and Accounting Server.

6. Security Considerations

Security is of paramount concern in accounting. It must be possible for the accounting protocol to be implemented atop a secure transport. A canonical record format is useful so that regeneration of secure record hashes is possible.

7. References

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

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

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

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

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

[6] Blount, A., D. Young, "Metered Service Information Exchange 1.2", Work in Progress, <draft-blount-acct-msix-00.txt>, July 1999.

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

[8] Horton, M., R. Adams, "Standard for Interchange of USENET Messages", RFC 1036, December 1987.

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

[10] Rigney, C., "RADIUS Accounting", RFC 2059, January 1997.

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

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

[13] Kantor, B., P. Lapsley, "Network News Transfer Protocol", RFC 977, February 1986.

[14] Berners-Lee, T., D. Connolly, "Hypertext Markup Language - 2.0",

RFC 1866, November 1995.

8. Authors' Addresses

Alan Blount
MetraTech Corp.
411 Waverley Oaks Road
Waltham, MA 02452
Email: blount@alum.mit.edu