ipdr.org

Network Data Management – Usage (NDM-U) For IP-Based Services

Version 2.0

October 23, 2000

Preface

Contacts

For general questions regarding this document and referrals to technical experts for detailed questions, please contact:

Chief Editor
Steven A. Cotton
Cotton Management Consulting
scotton@compuserve.com

BSS Group Editor
Pat Walls
GTE TSI
pwalls@tsi.gte.com

Protocol Group Editor
Ken Sarno
NARUS, Inc.
kensarno@narus.com

Acknowledgements

The following member companies contributed materially to the creation of this release of the document:

Charter Members
ACE*COMM
Amdocs
AT&T
Convergys Corporation
Daleen Technologies
HP
Lucent Technologies
Narus, Inc.
Sprint PCS
XACCT Technologies

Supporting Members
American Management Systems
Comptel Plc
Computer Generation, Inc.
MetraTech Corp.
Openet Telecom LTD
Telcordia Technologies

Associate Members
Nexus Telecom AG
RateIntegration

Abstract

In order to develop the IPDR record format it is required that a framework be in place that formally classifies the “IP network and service elements” and “support systems.” Next, the relationship between such subsystems needs to be in place to determine the flow of information between components. Finally, the requirements of each subsystem must be determined in order to specify the type of “IP resource and service usage information” that will be exchanged.

This document proposes a reference model to satisfy the above requirements. Chapter 2 proposes a general IP support system framework. This Chapter is intended to identify key components found in production IP networks, present the relationship between them, and define basic terminology. The Chapter focuses on the network and service element layer (NSE), and illustrates the concepts with example ISP network infrastructures. Chapter 3 details the business requirements that the structures and systems defined in Chapter 2 must satisfy. Next, Chapter 4 describes the information flow requirements between the NSE layer and the various support systems identified. Chapter 5 concludes with a proposed structure of the IPDR record for various IP-based services, and motivates this structure using the business requirements identified in Chapter 3 and the technical requirements identified in Chapter 4. Chapter 6 is the template for documenting service requirements definitions in Chapter 3.

The following summarizes the essential content of each revision of this document, as of that release:

1.0 – This revision of the document represents the state of the work at a point determined by the working groups to be useful for broader review and validation. Many issues have been identified for further work and are not fully addressed in this issue. Additionally, it is anticipated that domain experts will contribute via liaison relationships currently being established. The services represented in this version may be substantially altered once these contributions are considered, even if at the expense of backward compatibility. Future revisions are expected to make every attempt to preserve investments made by service providers and solution vendors by considering backward and forward compatibility whenever it is practical.

1.1 – This revision was editorial, reflecting comments received from public review and experiences of the first prototype implementations. No significant changes to the 1.0 content occurred.

2.0 – This revision also introduces a major upgrade of the syntax notation of the protocol, namely XML Schema versus XML 1.0. This upgrade has been introduced to allow the protocol to specify strong typing of the usage attributes, thus conforming to the business requirements for data integrity. In addition, the dynamic operation of IDPR document transport has been specified, using the consensus choice for best conforming to business requirements, Simple Object Access Protocol (SOAP). Finally, the usage attributes for each of the services defined in the Business Requirements chapter are now formally specified, using the XML Schema definition supplied in the Protocol chapter.

Change History

1.0 WDA	12/14/1999 Working Draft Revision A - Circulated to Membership for Comment
1.0.WDB	1/17/2000 Working Draft Revision B - Circulated to Membership for Comment
1.0 RDA	3/6/2000 Review Draft Revision A – Circulated to Steering Committee for Comment
1.0	3/20/2000 Steering Committee Approval
1.0.1	6/5/2000 Interim Issue to Correct XML Example
1.1 RDA	6/12/2000 Review Draft of 1.1
1.1	6/26/2000
2.0 RDA	September 15, 2000 Review Draft A of 2.0
2.0 RDB	September 22, 2000 Review Draft B of 2.0
2.0 AD	October 1, 2000 Approval Draft to be Balloted at St. Louis Meeting
2.0	October 23, 2000 Official Release

Table of Contents

1. Introduction

1.1. Purpose

This document is intended to specify technical information that is sufficient for practical implementations of interchange of usage data among service elements participating in the delivery of IP-based services, either within a single enterprise or across multiple enterprises.

The IPDR organization intends to submit this specification to selected accredited organizations for consideration as an approved standard.

1.2. Scope

This document is limited to the discussion of issues as defined by the mission statement of IPDR.org, namely:

> *The IPDR Organization (the "Organization") is organized and operates as a non-stock not for profit organization for the following purposes:*
>
> *(a) To develop, agree upon and publish a non-proprietary, open specification for the representation and encapsulation of Internet Protocol (IP)-based events for use by business, operations and decision support systems. Such events include, but are not limited to, IP-based network services, application services and e-commerce transactions;*
> *(b) To develop, agree upon and publish a non-proprietary, open specification for the representation and encapsulation of IP-based network and service elements provisioning events;*
> *(c) To promote work accomplished and uniform specifications to the industry and submit approved published specifications to the appropriate standards bodies for acceptance in the public domain; and*
> *To have and exercise all powers necessary or convenient to affect any or all of the purposes for which the Organization is organized.*

1.3. Compatibility

Future revisions are expected to make every attempt to preserve investments made by service providers and solution vendors by considering backward and forward compatibility whenever it is practical.

1.4. Timeline

At the recent quarterly meeting in St. Louis, IPDR.org outlined its ongoing work plan. The next release of the NDM-U specification is scheduled for February 2001. The Protocol Working Group will focus on ensuring the scalability of the IPDR standard. To that end, they will focus on developing traffic models and metrics for measuring IPDR processing. The Business Requirements Working Group will continue development of IPDR specifications for new services such as VoIP, VoD, eCommerce/mCommerce. It will also work on methods of converting existing legacy specifications to new IPDR specifications.

1.5. References

[1] Telecom Operations Map - Evaluations Version 1.1, GB910, *TeleManagement Forum (TMF - http://www.tmforum.org)*, April 1999.

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

[3] XML Schema Part 2: Data Types, W3C Working Draft 7 April 2000.

[4] Simple Object Access Protocol (SOAP) 1.1, W3C Note 08 May 2000.

[5] IPDR Organization Timeline.

[6] IPDR Organization FAQ.

1.6. Overview

This specification is divided into four major chapters:

- IPDR Reference Model - a definition of the abstract and operational relationships between entities involved in the generation, recording, storage, transport, and processing of usage attributes.
- Business Requirements - a definition of business requirements to be addressed by the protocol specification and specific scenarios for the major process flows anticipated in actual application.
- Protocol - the notation, data unit syntax, and dynamic procedures involved in the operation of the interfaces specified in the reference model.
- Structures - the specific usage and document attributes and collections of such attributes associated with services for which accounting occurs.

The Protocol and Structures chapters represent the specific design produced through analysis of the Business Requirements chapter, consistent with the Reference Model chapter.

1.7. Terminology and Glossary

Terminology

Term	Definition
Accounting	The process of collecting and analyzing **service** and **resource usage** metrics for the purposes of capacity and trend analysis, cost allocation, auditing, and billing, etc. Accounting management requires that resource consumption be measured, rated, assigned, and communicated between appropriate business entities.
Mediation	In view of network reference model, Mediation refers to the combination of the logical entities IPDR recorder, IPDR transmitter, and IPDR store.
Resource	A quantifiable asset employed by a **Service Provider**, or on behalf of a **Service Provider** by another Service Provider, to fulfill a request of a **Service Consumer**. (Examples include: files, communications, goods, etc).
Roaming	Service usage initiated by a service consumer and provided by a service provider other than the one with which the service consumer have business relationship.
Service	Network and/or application operation that provides the **Service Consumer** with the requested **resource.**
Service Consumer	The beneficiary (human or system) of a **service**.
Service Element	Any element that is responsible for fulfilling a **Service Consumer** request. (Examples include: network equipment and system processes)
Service Provider	An enterprise that provides communications-based **services**.
Session	A set of related service usages; service usages may or may not be time based in the unit of measurement.
Usage	Consumption of **resources** and **services** by a **Service Consumer**.
Usage Attribute	A parameter whose value indicates some aspect of **usage** of a given **service** and/or **resource.**
Usage Entry[1]	A **Service**-specific trigger resulting in the generation by a **Service Element** of a set of **Usage Attribute** values related to **Usage** specific to a given **Service Consumer**

[1] Because of legacy issues, a Usage Entry from a given Service Element will not initially conform to an IPDR specification or, in some cases, may never conform. To be considered a Usage Entry the information presented or made available by inference from the Service Element must minimally contain attributes from some of the general attribute categories.

Glossary:

ANI	- Automatic Number Identification
ASP	- Application Service Provider
BSS	- Business Support Systems
CRM	- Customer Relationship Management
DSS	- Decision Support Systems
DTD	- Document Type Definition
DSL	- Digital Subscriber Line
EP	- End Point
ESN	- Electronic Serial Number
FoIP	- Fax over IP
GK	- Gate Keeper
GPRS	- General Packet Radio Service
IETF	- Internet Engineering Task Force
IMSI	- International Mobile Subscriber Identity
IP	- Internet Protocol
IS	- IPDR Store
ISDN	- Integrated Services Digital Network
ISP	- Internet Service Provider
IT	- IPDR Transmitter
NDM	- Network Data Management
NSE	- Network Service Element
OSS	- Operations Support System
PLMN	- Public Land Mobile Network
PSTN	- Public Switched Telephone Network
QoS	- Quality of Service
RADIUS	– Remote Access Dial-In Usage Server
RAS	- Remote Access Server
SC	- Service Consumer
SE	- Service Element
SMS	- Short Message Service
SP	- Service Provider
TMF	- TeleManagement Forum
TOM	- Telecommunications Operations Map
UA	- Usage Aggregators
UC	- Usage Collectors
VoIP	- Voice over IP
VPN	- Virtual Private Network
WAP	- Wireless Application Protocol
xDSL	- Digital Subscriber Line of type x
XML	- eXtensible Markup Language

2. IPDR Reference Model

The IPDR organization has adopted the Telecommunication Management Forum's (TMF) telecommunications operation map (TOM) for the purposes of motivating the functional role and interfaces of the IPDR specifications relative to operations support systems (OSS). We have chosen the TOM because it is a well-known, industry-accepted organizational model of telecommunications support systems used by carriers and service providers today. The TMF Model is useful as a model of typical systems, and as motivation for design decisions. However, the TMF Model itself is not part of IPDR, and the data structures and interfaces of IPDR may be used in systems that vary substantially from the TMF Model. See [1] for more details.

2.1. IPDR and the TMF Model

The TOM, shown in Figure 1, identifies the core operation support processes found in a production carrier business operation. The systems that implement the customer care, services development/operations and network/systems management processes each provide a well-defined set of services that enable a carrier to successfully deploy and manage telecommunications services. As the model shows, these systems are organized in a layered fashion. Thus, each component builds on the services provided at a lower layer (and possibly adjacent components) to deliver the required functionality.

The IPDR organization's charter is to facilitate the integration of IP-based network elements into billing, reporting and assurance systems. In particular, one key goal is to define a common usage record format and exchange protocol to facilitate the flow of usage information from IP network elements managers to support systems. In the TOM, the network data management (NDM) component (defined as part of the network and systems management processes) defines the device-independent collection mechanism for such purposes. As such, the work of this specification falls into the definition of the Network Data Management component.

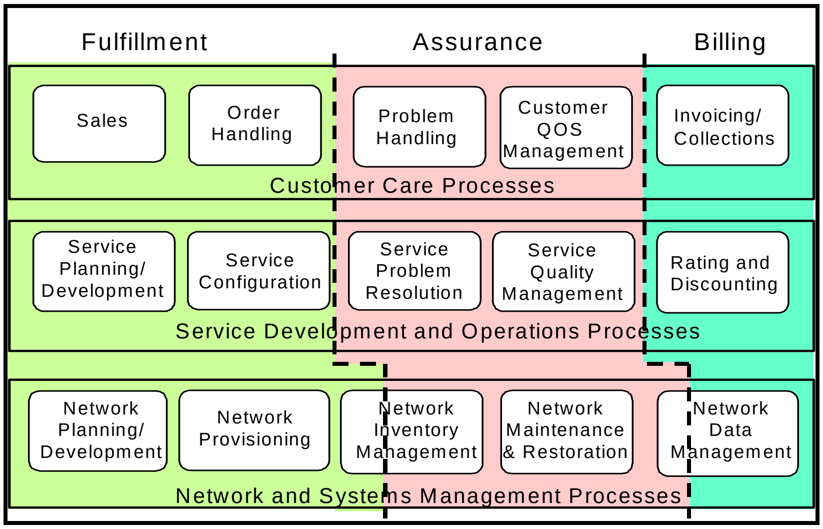

Fulfillment
Assurance
Billing
Sales
Order Handling
Problem Handling
Customer QOS Management
Invoicing/ Collections
Customer Care Processes
Service Planning/ Development
Service Configuration
Service Problem Resolution
Service Quality Management
Rating and Discounting
Service Development and Operations Processes
Network Planning/ Development
Network Provisioning
Network Inventory Management
Network Maintenance & Restoration
Network Data Management
Network and Systems Management Processes

Figure 1 – Telecommunications Operations Map (TOM)

2.2. IPDR NDM-U High-level Model

Within the scope of the NDM-U module, the TOM shows interfaces to billing (i.e., rating and discounting) and customer care systems (i.e., service quality management and problem resolution). Likewise, the figure shows that the NDM-U component must interface directly with the network and service element manager to accomplish their various services.

The IPDR NDM-U reference model, shown in Figure 2[2], expands on the NDM-U definition by dividing the module into layers; namely: (1) the network and service element layer, (2) the mediation layer, and (3) the business support systems layer. Each layer is discussed below:

- Network and service element layer (NSE): The NSE layer consists of all the network and service elements required to provide an IP-based service to a given customer. For example, routers, access devices and transmission facilities together provide basic connectivity; firewalls might provide a security service; email, file and print servers provide application services; gateways provide a translation service from circuit to packet voice; and more. In addition to physical devices, the systems that configure and manage such devices are considered part of the NSE layer (note, that this functionality is identified as adjacent component within the "Network and Systems Management" layer in the TOM model). Examples here include a bandwidth management system, H.323 gatekeeper, RADIUS, authentication server or network management platform.

- Mediation layer: As shown in Figure 2, mediation systems sit between the network elements/infrastructure and the business support systems. Typically, a mediation system provides a single interface to BSS systems that provides all network usage data as well as a single service elements provisioning. In terms of usage collection, the goal of the mediation system is to capture all usage information required by the BSS systems, and export it within the temporal requirements. Thus, the mediation system must, in some way, determine the devices at the service element layer and interface with that infrastructure to extract the relevant usage information. The second mediation goal is to pass provisioning information from the BSS, to the network elements – again, within the temporal constraints.

- Business support systems (BSS) layer: The BSS layer consists of the systems deployed by a Service Provider or provider to support IP business operations. This layer corresponds to the "Systems Development and Operation Processes" in the TOM model. Some examples include billing (i.e., rating and discounting), customer care/relationship management, decision support, and market analysis and fraud detection. The BSS layer is the highest layer in the model. Thus, the BSS usage collection and provisioning requirements drive the mediation system and ultimately the services provided at the service element layer.

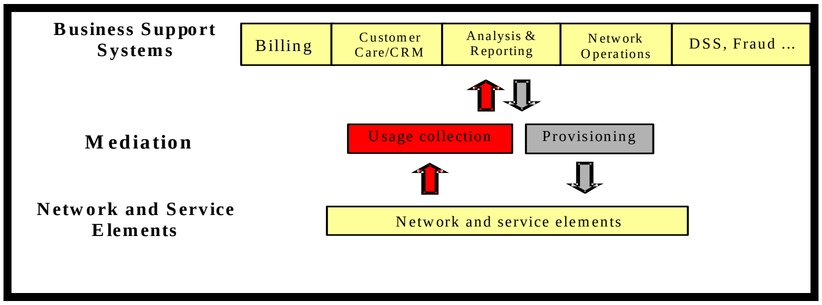


Figure 2 – IPDR NDM-U High-Level Model

The IPDR NDM-U model shown in Figure 2 gives a layered perspective of the components and interfaces designed to meet the NDM-U specification. The mission statement given in the introduction limits the organization's scope to the usage collection path (shown flowing upward in Figure 2). Thus, provisioning or the internal design of any of the identified components is not considered in this document.

[2] Note that both usage and provisioning flows are depicted in this figure. Only the usage component is discussed further in this issue.

The usage collection process represents a flow of usage data from the network and service elements to the BSS processes. Figure 3 illustrates the usage data path from network elements (e.g., gateways, remote access servers (RAS), routers, and bandwidth managers) to a mediation device. Note that this example assumes the interface between the mediation device and network elements is based on a proprietary access protocol, record format and API. The mediation system aggregates, normalizes and correlates the usage data as required, and exports the data to the billing, decision support, or other business support systems.

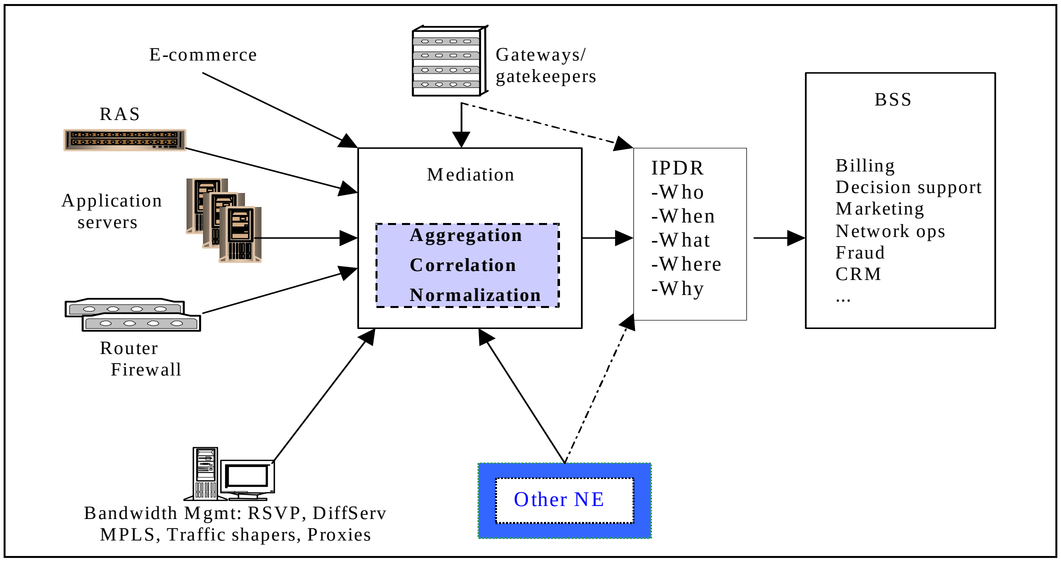


Figure 1 – IPDR Record Flow

The IPDR NDM-U plays several roles in this data transfer. First, the IPDR record provides flexible structure that is sufficiently powerful to describe the usage attributes collected by the mediation system, and required by the BSS system. Second, the IPDR NDM-U provides a set of interfaces that facilitate the exchange of IPDR records between mediation systems and BSS systems, or between IP network elements and BSS systems. Finally, the IPDR specification provides a set of interfaces that facilitate the intermediate storage of IPDR records between IPDR-enabled components.

2.3 IPDR Record Structure

The IPDR record structure must be capable of characterizing any type of usage that might be collected from an IP-based network or application service. As figure 4 shows, there are 5 components common to all IPDR records. Broadly, these components are the “who, what, where, when and why” values that describe a particular usage event. Each are discussed below:

- {Who} (Responsible for the usage)
  User ID (in some form, if available)

- {When}
  End Time or Event Time

- {What}
  Service
  Usage measures / quantities
  Ex: Bytes, packets, flows, hits, transactions, time duration...
  QoS measures
  State information
  Event code (logon, logoff, threshold exceeded)
  Other information about state transition or current state (Start Time)[3]

- {Where}
  Traceability / Context
  Source Identifier
  Destination Identifier
  Service Element identifier (originator)

- {Why}
  Event trigger type – (i.e., why is the network and service element reporting this data?)

In addition to the “5Ws” defined above, each record may include reference pointers to other IPDR records that either capture related usage information, or contain usage information that was used to create the given record.

[3] Note that "always on" services may be measured via periodic emission of IPDRs, recording usage since the last interval boundary.

2.4 IPDR Interfaces

In addition to the IPDR record structure, the IPDR specification defines a set of interfaces for exchanging IPDRs between IPDR-enabled devices or systems. As will be specified in the Protocol chapter, IPDRs are packaged in protocol data units (PDUs) know as IPDR Documents (IPDRDocs). These PDUs are the entities for which protocol transactions and tracking are done. All future references to IPDR documents imply these formal protocol entities. Figure 5 shows the key interfaces and elements found within the NDM-U module, represented in an abstract form. Note that this model does not constrain implementations to be physically packaged as portrayed, nor to present all of the interfaces to other systems.

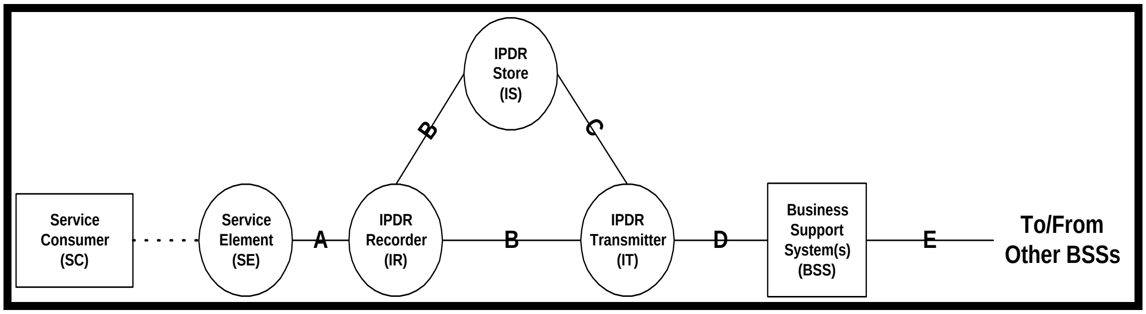


Figure 4 – Basic Network Model

Each element and the roles played by the various nodes in the model are given below:

- Service Consumer (SC) - requests and receives services (typically end user on end system)
- Service Element (SE) - provides access to services and resources requested, authenticates SC, authorizes access, performs accounting measurement for resources provided, provides services requested by SC, and performs accounting measurement for services provided
- IPDR Recorder (IR) - packages usage information into IPDRs, optionally packages IPDRs into IPDR documents and presents the documents to an IS or an IT
- IPDR Transmitter (IT) - delivers IPDR documents to a BSS
- IPDR Store (IS) - retains IPDR documents from an IR in a non-volatile medium, presents such documents to an IT upon request
- Business Support System (BSS) - receives information contained in IPDR documents from an IT, processes the information contained in IPDRs for use in the commercial activities of a Service Provider, presents information for transmittal to other BSSs

The following describes the various interfaces in the model:

- A - for delivery of usage information from SEs to IRs[4]
- B - for delivery of IPDR documents from IRs to ISs and ITs
- C - for delivery of IPDR documents from ISs to ITs
- D - for delivery of IPDR documents from ITs to BSSs
- E - for delivery of IDPR documents from BSS to BSS

[4] Note that the A interface is not specified in detail in this document, but rather minimally constrained as to the basic behavior and content that is necessary to allow the recording of an IPDR compliant with BSS requirements.

3. Business Requirements

3.1. Introduction

For each IP service, this chapter provides high-level requirements, general and specific usage attributes, and use cases for BSS applications needs. It also provides the framework for specifying new IP services not yet covered in this chapter to meet the extensibility needs of the future.
Section 3.4 provides the general overview of the network model from previous chapters and it's applicability to the BSS applications needs. The focus for specifying requirements is given to D interface though there may be implied requirements to other interfaces. Section 3.5 provides general requirements and general usage attributes applicable to any IP service. Note that any new IP service added in the future may impact the general requirements and usage attributes. Section 3.6 provides IP services covered and yet to be covered in this chapter. Section 3.6 provides the use cases for the services covered in this chapter

3.2. Notation

Use cases are used in this chapter to describe the environment of business requirements. To standardize the extension of those use cases a use case template is developed and included in appendix A.

3.3. Assumptions

- Home Service Provider handles all business needs (via BSS applications) of the Service Consumer.
- Service Consumer may or may not be within the Home Service Provider's service area.
- BSS to BSS interfaces (E Interface) is outside the scope of this version of the NDM-U.
- Some applications of NDM-U will result in large numbers of IPDRs being generated, requiring economical storage, transport, and processing implementations. Several requirements stated below are intended to address this assumption. However, no quantitative requirements regarding performance (end-to-end delay, transfer rate, etc.) or efficiency (message size, compression ratio, etc.) will be stated in this document. The mechanisms designed in later chapters of this document, which satisfy the general requirements in this area, should give implementers adequate tools to make cost versus technology tradeoffs, justified in light of the business problem being solved. Product vendors designing implementations of this specification are assumed to be aware of the overall marketplace requirements for such systems and service providers selecting one or more of these implementations will be expected to require those vendors to demonstrate competitive features in the area of performance.

3.4. Network View

The interfaces described in the abstract network model for an IPDR system have been projected into a generic Network View as shown in the table below. This is done to allow various operations scenarios, some of which will involve complex relationships between SPs. The use cases will be expressed in terms of this Operations Model, thus allowing the specific service usage to be tied to the behavior at the specified interfaces in the Protocol chapter.

The Operations Model is segmented into five domains:

- Service Consumer - there are no IPDR NDM-U interfaces specified relating to this domain. The interface with SP1 is with an unspecified access interface, possibly over one or more categories of media (PSTN, cable network, PLMN, xDSL, ISDN, fixed wireless, mobile data, or a combination of one or more). The Service Consumer is not necessarily associated with SP1 from a BSS perspective in this model (i.e., a roaming scenario is possible).

- Service Provider 1 (SP1: Access/Transport/Application Services Provider) - this SP plays the unique role of providing access services, at a minimum. This SP may also provide transport and/or application services in given scenarios.
- Service Provider 2 (SP2: Transport/Application Services Provider) - this SP plays the role of transport services between SP1 and SP3, at a minimum. They may also provide application services in given scenarios.
- Service Provider 3 (SP3: Application Services Provider) - this SP provides only application services, relying on one or more access/network SPs for establishment of sessions with an SC.
- Service Provider 4 (SP4: BSS Services Provider) - this SP acts as an intermediate BSS on behalf of one or more other SPs. Such applications as service bureaus, clearing houses, rating bureaus, fraud bureaus, pre-paid authorization centers and other intermediate IPDR processing applications are examples of the role of this SP.

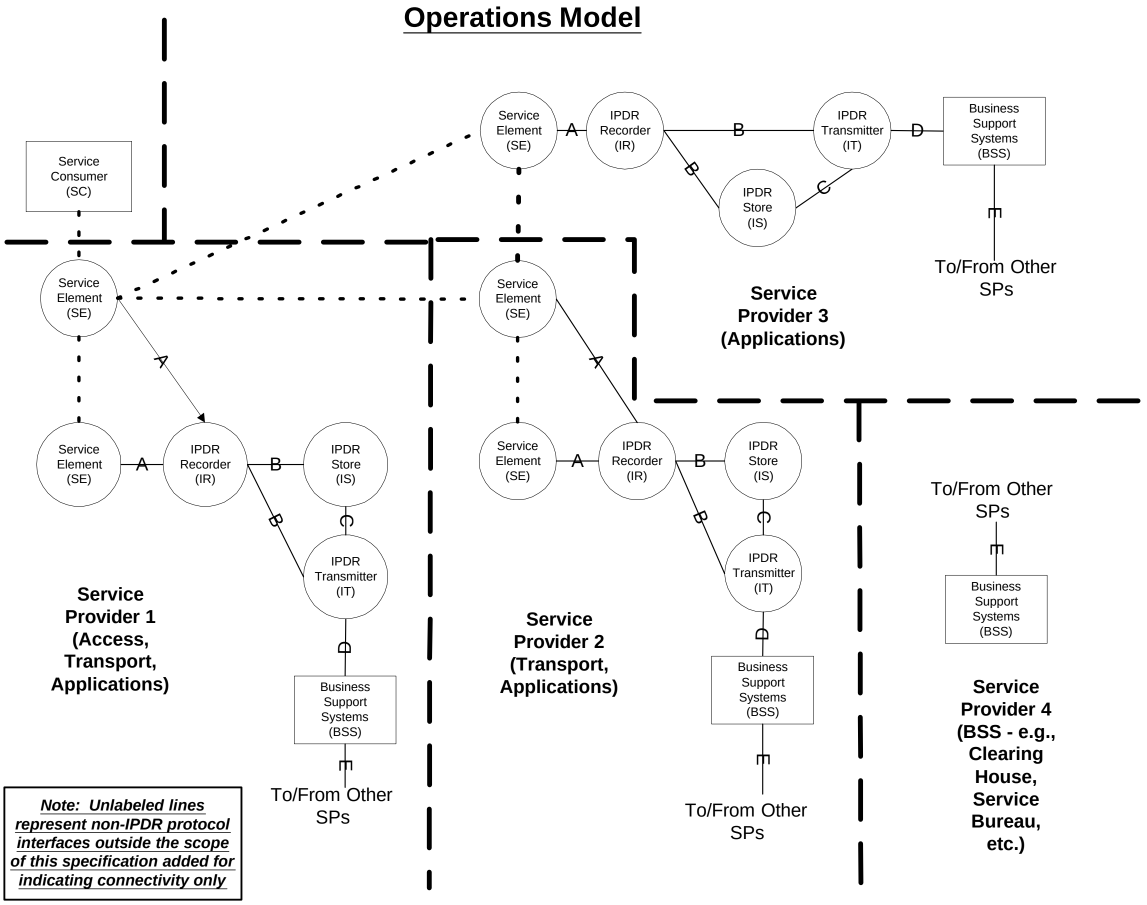

Operations Model
Service Consumer (SC)
Service Element (SE)
IPDR Recorder (IR)
IPDR Transmitter (IT)
IPDR Store (IS)
Business Support Systems (BSS)
A
B
C
D
E
To/From Other SPs
Service Provider 1 (Access, Transport, Applications)
Service Provider 2 (Transport, Applications)
Service Provider 3 (Applications)
Service Provider 4 (BSS - e.g., Clearing House, Service Bureau, etc.)
Note: Unlabeled lines represent non-IPDR protocol interfaces outside the scope of this specification added for indicating connectivity only

Figure 5 – Network view

3.5. General Requirements for the Operational Model

This section lists requirements that are service independent. That is, requirements that are not captured in the specific service use cases covered in section 3.8.

3.5.1. Mediation Requirements

The general requirements for mediation are, in almost all cases, service independent. Depending on the business model mediation tasks could span a wide variety of actions. However, in general terms mediation tasks include the collection, generation, aggregation and reconciliation of IPDRs across Service Elements, geographical areas and time.

1. Mediation shall support both polling and pushing for data transfer, so that the data transfer can be initiated either by the collector or spontaneously by the Mediation.
2. Mediation shall support data transfer for both individual events and batches of events.
3. Mediation shall support retrieval of IPDR documents.
4. Service elements shall be uniquely identified within the scope of each terminating IPDR recorder.
5. IPDR shall have a unique event identifier within service elements. If IPDRs are related and the relation is visible to the IPDR recorder (aggregator) then, a reference to the related record (base IPDR) shall contain this unique identifier.
6. IPDR shall enable the interim recording across multiple service elements and time. That is, enabling event information to exist in multiple records, over several IPDR documents.
7. Mediation shall support uniquely identifying IPDR documents for the purpose of gap and duplicate detection.

3.5.2. Format

This set of requirements pertains to the IPDR format.

1. The IPDR format shall be extensible permitting the addition of any set of services and service specific usage attributes.
2. The IPDR format shall be able to self-describe its usage attributes.

3. The IPDR format shall capture sufficient information to identify an IPDR service consumer.
4. The IPDR format shall provide specified data types, so that various systems can interpret the data properly.

3.5.3. Application Protocol

1. The IPDR protocol shall support encryption of IPDR documents.

2. The IPDR shall support efficient encoding.

3. The IPDR shall use open protocols and description languages.

4. IPDR protocol/format shall separate the record format and exchange protocol.

5. IPDR protocol shall support transfer capabilities negotiation.

6. IPDR protocol shall support both individual and batch transfers of data

7. IPDR protocol shall support resynchronization to a particular point in the order of delivery of IPDR documents.

3.5.4. Usage Attributes

1. The IPDR format specification shall indicate, for all usage attributes, if the information is required, optional or conditional.

2. The IPDR format specification shall indicate usage attributes data type.

3. Where appropriate, a data type of value/unit shall be specified to denote the unit of measure of an associated attribute value.

3.5.5. Settlement

1. The IPDR protocol and format shall support roaming.

2. The IPDR protocol and format shall support mobile service consumer.

3.6. Listing of Services

3.6.1. Services Covered

For describing the context environment of the business requirements listed in this chapter a set of services are analyzed then, for each service a multiple of use cases are depicted. The list of services considered in this chapter is a representative and not a comprehensive list. This list will be augmented through contributions by other relevant standard bodies and through the progress of IPDR organization work.

Services considered by the BR working group in this version of the draft are:

1. Authentication and Authorization Services (AA)
2. Internet Access (including wireless)
3. Content/Service (including wireless)
4. Push Delivery (including wireless)
5. Wholesale Requirements

Services considered by the BR working group in previous version of the draft are:

1. Application Services (ASP)
2. Voice over IP (VoIP)
3. E-mail Services

3.6.2. Services for Future Consideration

Since the list of services considered in this version is not a comprehensive list, and recognizing the importance of other services, we are including a list of services to be considered in future releases of these specifications.

1. Video on Demand (VoD)
2. Virtual Private Networks (VPN)
3. Multi-party conferencing (video/voice)
4. E-commerce/M-commerce
5. Unified Messaging
6. Video conferencing over IP
7. IP television

3.7. Services Considered by other Organizations (References)

It is recognized that the specification of services requires expertise and experience in the providing or equipping such services. The IPDR encourages domain experts and service providers to submit specifications of services whose usage would be recorded by an IPDR recorder. The form of such submissions should conform to the templates and guidelines described in this chapter.

3.8. Use Cases

The following service definitions document the requirements specified in accordance with the general requirements stated above. The attribute lists are presented in a tabular form to aid readability, but the formal specification of the records is in Chapter 5. Any inconsistency with the tabular attribute listings from Chapter 3 shall be resolved by giving the Chapter 5 specifications "normative" precedence.

3.8.1 Application Service Provider

3.8.1.1. Service Definition

Application Service is the use of applications supplied by providers outside the service consumers business.

3.8.1.2. Service Requirements

1. IPDR must provide unique and clear identification of parties participating in the activity.
2. IPDR must provide information about the amount and type of resources used.

3.8.1.3. Service Usage Attribute List

Category	**Usage Attribute Name**	**Data Type**	**Presence**	**Possible Values**	**Remarks**
What	Feature	String	Conditional	"Sort" as a feature of a "spreadsheet" app, etc.	Specific feature of service
What	Type	String	Conditional	"Front office service", "front office/word processor service", "front office/spreadsheet service", etc.	Type of application that is invoked.
When	AppRequestTime	Datetime	Required	ISO 8601 time	May be different from AppStartTime. This will allow measuring response time.
When	AppStartTime	Datetime	Required	ISO 8601 time	Time when application starts
Where	LoginLocation	String	Required		
Where	ProviderLocation	String	Required		Will support providers that host applications at different locations
Who	ProviderName	String	Required		Actual provider of the service

Category	Usage Attribute Name	Data Type	Presence	Possible Values	Remarks
Who	UserLoginName	String	Required		Identifies a unique user in the system. Real time mapping of dynamically allocated IP addresses might be necessary

3.8.1.4. Front Office Service Use Case

Front office service provides ASP user with access to common office applications.

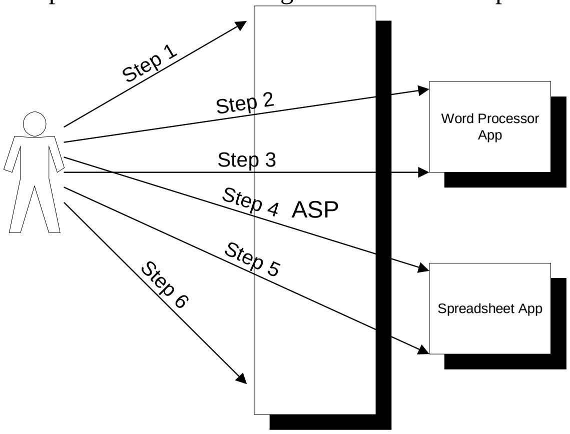
3.8.1.4.1. Basic Flow

1. User logs into an Application Service Provider (ASP).
2. After authentication, the user invokes a word processor application (front office service).
3. In the middle of using the word processor, the user invokes a spreadsheet application.
4. While still using the spreadsheet, application the user closes the word processor application.
5. The user closes the spreadsheet application.
6. The user logs out.

This could be considered as a single session with multiple events. Events are:

Start of session (at login)
Start of word processor
Start of Spreadsheet
End of word processor
End of spreadsheet
End of session

Another alternative is considering the word processor and the spreadsheet to be different services. As such, two separate records tracking identifiers will be produced one for each service.



3.8.1.4.2. Basic Flow Requirements

These are the basic flow requirements for the front office use case:

1. IPDR must provide information about the time that the event occurred.
2. IPDR must provide a correlation between the start and end events for the application.

3.8.1.4.3. Basic Flow Usage Attribute List

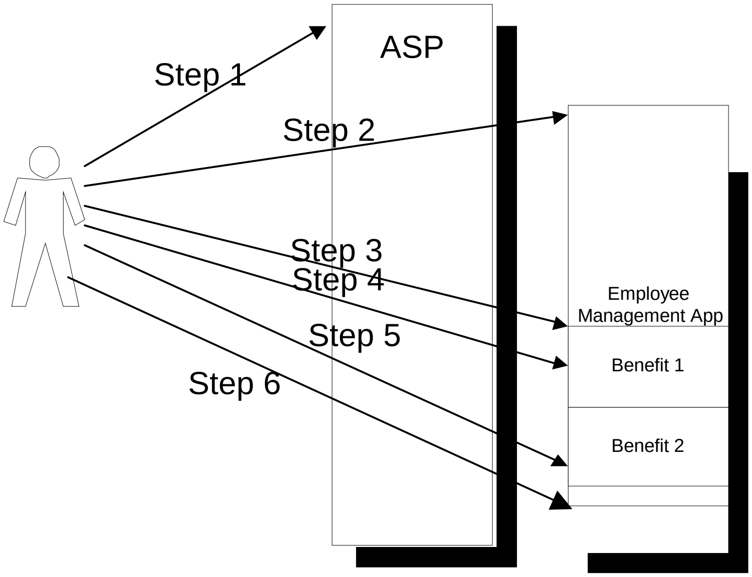
Category	**Usage Attribute Name**	**Data Type**	**Presence**	**Possible Values**	**Remarks**
What	AppActiveTime	Integer	Conditional	Milliseconds	Total elapsed active time for each process in the session. Active time is a measure of time when the CPU usage exceeds a certain percentage. At least one of the conditional usage attributes must be present.
What	AppLoadedTime	Integer	Conditional	Milliseconds	Total elapsed loaded time for each process in the session
What	NumberOfApps	Integer	Conditional		Number of apps invoked during the session time.
What	SessionDuration	Integer	Conditional	Seconds	Duration of the session.

3.8.1.5. Back Office Service Use Case

Back office service provides ASP user with access to applications specific to back-office operations.

3.8.1.5.1 Basic Flow

1. User logs into an Application Service Provider (ASP).
2. After authentication, the user invokes an employee management application.
3. The user checks his benefits info.
4. The user cancels one of the benefits.
5. The user modifies another benefit.
6. The user logs off.



3.8.1.5.2 Basic Flow Requirements

1. IPDR must provide unique and clear identification of parties participating in the activity.
2. IPDR must provide information about the type of service used and operations performed.
3. IPDR must provide service consumer activities to including number of requests and amount obtained for each service feature.

3.8.1.5.3 Basic Flow Usage Attribute List

Service Definitions

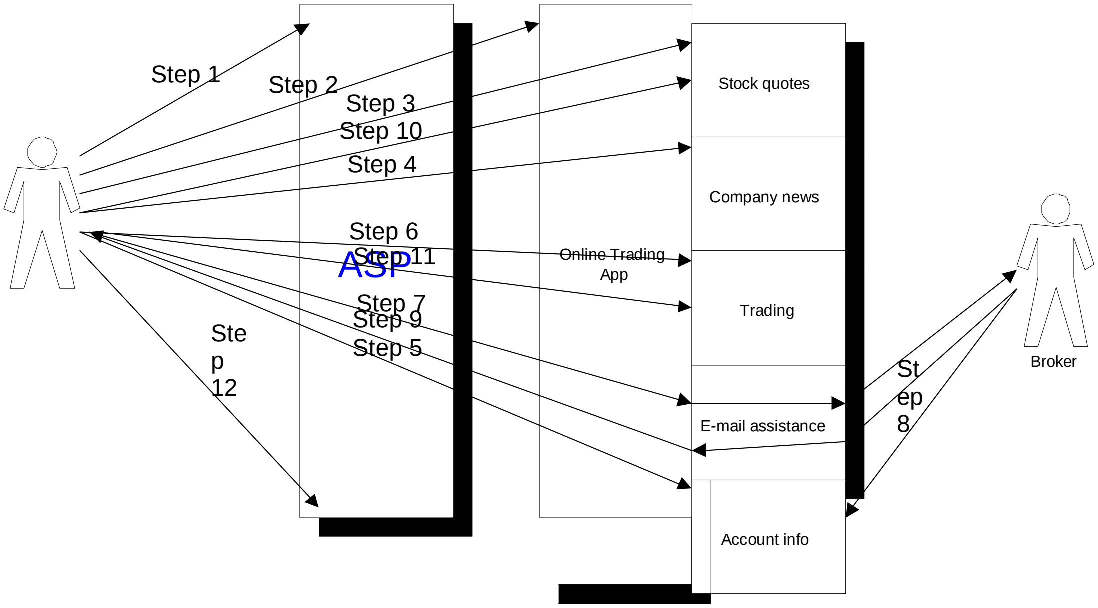
Category	**Usage Attribute Name**	**Data Type**	**Presence**	**Possible Values**	**Remarks**
What	BytesTransferred	Integer	Conditional		Number of bytes transferred on request basis. Should not include any inline images or ad view.
What	NumberOfTransactionsRequested	Integer	Conditional		Number of transactions requested by the user during the session.
What	NumberOfTransactionsCompleted	Integer	Conditional		
What	RequestDuration	Integer	Conditional	Seconds	Time between two consecutive requests
What	VisitTime	Integer	Required	Seconds	The duration that covers a series of consecutive requests to the ASP site, bounded by the first and last requests made by user

3.8.1.6. Online Trading Service Use Case

Online trading service provider provides ASP customer with an opportunity to view the current stock quotes and trade stocks online. It includes access to latest company news, comprehensive account information services, and e-mail assistance from a brokerage firm. It is assumed that the brokerage firm owns the brokerage application.

3.8.1.6.1. Basic Flow

1. User logs into an Application Service Provider (ASP).
2. After authentication user invokes online trading application using his account number.
3. The user checks current stock quotes
4. The user browses latest company news.
5. The user checks his account status.
6. The user attempts to place a stock trading request, encounters a problem with his account.
7. The user sends e-mail to his broker through the application, waits for response.
8. Broker modifies a parameter on user’s account.
9. The user receives a response from the broker.
10. The user checks stock quotes again.
11. The user places a stock-trading request, this time successfully.
12. The user logs out.



1. Basic Flow Requirements IPDR must provide unique and clear identification of parties participating in the activity, and their locations.
2. IPDR must provide information about the amount and type of resources used.
3. IPDR must provide a list of trading requests executed during the session.

3.8.1.6.2. Basic Flow Usage Attribute List

3.8.1.6.2.1. Service Consumer

Category	**Usage Attribute Name**	**Data Type**	**Presence**	**Possible Values**	**Remarks**
Who	UserAccountNumber	String	Required		May be derived from the user login

3.8.1.6.2.2. Service Definitions

Category	**Usage Attribute Name**	**Data Type**	**Presence**	**Possible Values**	**Remarks**
What	NumberOfEmailAssistanceRequests	Integer	Conditional		Total number of e-mail assistance requests per session
What	NumberOfNewsRequests	Integer	Conditional		Total number of company news requests per session
What	NumberOfQuotes	Integer	Conditional		Total number of stock quotes obtained during the session
What	SessionDuration	Integer	Conditional	Seconds	Duration of the session, including both browsing and trading
What	TradeRequests	String	Conditional	“MSFT, buy, 100”, etc.	List of trade requests

3.8.2. Voice over IP (VoIP)

3.8.2.1. Service Definition

VoIP is voice communications between two or more parties over a partial / complete Internet-based connection. The “call” is initiated by a calling party and received by recipient(s). The “call” participants include service elements, gatekeepers, and endpoints (end-users).

The transmission path of the call is realized at VoIP switch by a VoIP gatekeeper (GK) and at each customer location by an Endpoint (VoIP/PSTN). At customer locations, the user speaks and listens into a device that carries the voice data.

The intent is for SEs to generate and transfer to a BSS IPDR records which represent each voice call transparently between all SEs involved in the VoIP call.

3.8.2.2. Service Requirements

1. An IPDR must contain the identifiers of all call participants (call initiator & call recipients).
2. An IPDR must contain the time that the call was initiated and completed.
3. An IPDR must contain call progress codes for each call.
4. The state (as defined in the use case) that describes the phases that each SE goes through in a call must be contained by an IPDR.
5. All times contained in IPDR must be within one 1ms granularity and must have an accuracy of +-100ms.

3.8.2.3. IP to IP

This use case covers a VoIP scenario where the participating parties use completely Internet-based connections.

3.8.2.3.1. Basic Flow

1. A VoIP caller (IP based EP1) signals a GK1 for a call activation and passes in a callee (IP based call recipient – EP2) id / phone number.
2. The GK1 who owns the call may contact (via dir lookups) another GK to complete the call if the callee is not a subscriber in the GK1’s domain.
3. The GK with the callee as a subscriber acknowledges the request for service activation and proceeds to ring/signal the callee.
4. The callee answers the call and enters into the CallConnectedState. The EP’s deliver their voice content for a finite amount of time (call duration), and then disconnect the call (call complete).

Alternatively, an error may occur, and the call is disconnected. In duration of the call, various QoS changes may occur in real-time.

3.8.2.3.2. Basic Flow Requirements

1. Requires at least two EP’s for a call to be complete.
2. All EP’s must be “On-Net” and all GKs must be “On-Net”
3. An EP needs to be identifiable via either an ID or a phone number.
4. GKs (SP’s) must maintain a directory of subscribers, and each subscriber is assigned a unique ID within his domain Each GK must maintain a directory of other VoIP GKs/GWs.
5. Each GK has a universally unique identifier

3.8.2.3.3. Basic Flow Usage Attribute List

Category	Usage Attribute Name	Data Type	Presence	Possible Values	Remarks
Who	Subscriber_ID	String	Required		Unique within a service provider network Tied to a Service Consumer or a Service Element requesting a service
Who	IMSI_ingress	Integer	Conditional		Optional International Mobile Subscriber Identity Optional Required if calling party is using a cellular phone
Who	ESN_ingress	Integer	Conditional		Optional Electronic Serial Number which uniquely identifies each cellular phone Required if calling party is using a cellular phone
Who	Service_Consumer_Type		Optional	EU for end user, NE for network element, NK for partner network	
Who	PIN	String	Conditional		Optional Unique within a service provider network Tied to a Service Consumer or a Service Element requesting a service
When	Start_Access_Time	Date time	Optional		ISO 8601 time when a Service Consumer starts using a Network Element
When	Start_Time	Date time	Required		ISO 8601 time when a Service Consumer starts using a Service Element
When	End_Time	Date time	Required		
What	Call_Duration	Integer	Required	Seconds	This is exclusive of all set-up procedures
What	Total_Time	Integer	Required	Seconds	
What	Type		Conditional	A is for administrative (e.g. authentication and authorization), I is for IVR, N	

Category	Usage Attribute Name	Data Type	Presence	Possible Values	Remarks
				for no answer, V for voice, F for fax, D for data, VF for voice and fax combination, VD for voice and data combination	
What	Feature	String	Conditional	R for roaming, H for home	
What	Codec	String	Optional	G711Alaw, G711Mulaw, G723Low, G723High, G726, G727, G728, G729A, P for proprietary	
What	Modem	String	Optional		Optional Required if a modem is involved
What	Supplementary_Service	String	Optional		This field needs to be extensible to accommodate any number and any type of new service that could be used in conjunction with point-to-point IP telephony (e.g. call waiting, three-way calling, call forwarding, etc.)
What	Disconnect_Reason	String	Required		
What	Extended_Reason_Code	String	Optional		
What	Proprietary_Error_Code	Integer	Optional		Vendor-specific error code
What	Units_Consumed	Integer	Optional		
What	Average_latency	Integer	Required	Milliseconds	Incoming from the IP network. Measured from the preceding node in the call path
What	Inbound_Byte_Count	Integer	Optional		
What	Outbound_Byte_Count	Integer	Optional		
What	Fax_Page_Count	Integer	Conditional		
What	Packet_Loss_Percentage	Integer	Conditional		
What	Out_of_Sequen	Integer	Optional		

Category	**Usage Attribute Name**	**Data Type**	**Presence**	**Possible Values**	**Remarks**
	ce_Packets				
What	Correct_Sequence_Packets	Integer	Optional		
Where	ANI	String	Optional		
Where	ii_Digits	String	Optional		
Where	DNIS	String	Optional		
Where	Destination_Phone_Number	String	Conditional		Digit string entered by the calling party Different from DNIS if two-stage dialing
Where	Outpulsed_Digits	Integer	Optional		Digit string given to the switch on the egress side
Where	IP_Address_Ingress_Device	String	Optional		Required if using a gateway Null if using DHCP.
Where	IP_Address_Egress_Device	String	Optional		Required if using a gateway Null if using DHCP.
Where	Port_Number	String	Optional		
Where	IMSI_egress	String	Conditional		International Mobile Subscriber Identity Required if called party is using a cellular phone
Where	ESN_egress	String	Conditional		Electronic Serial Number which uniquely identifies each cellular phone Required if calling party is using a cellular phone
Where	Home_Location_ID_ingress	String	Conditional	An MSCID or IP address of an HLR	Required if calling party is using a cellular phone
Where	Home_location_ID_egress	String	Conditional	An MSCID or IP address of an HLR	Required if called party is using a cellular phone

3.8.3. E-mail Service

3.8.3.1. Service Definition

E-mail service is a service provided by Internet service provider that includes receiving and sending of messages. It also includes storage of incoming messages.

3.8.3.2. Service Requirements

Not applicable

3.8.3.3. Service Usage Attribute List

Not applicable

3.8.3.4. Basic E-mail

User subscribes to an e-mail service with a service provider. User is advised during the subscription process to configure her/his mail client to delete messages from the server on download. This process will eliminate storing of the e-mail messages on the service provider machines. In the case where download of messages is not practical, an example of which is using portable devices, the service provider would store a limited number of messages or use a limited amount of storage space. When the number of new messages or the amount of storage space exceeds a certain limit three alternative actions can be taken (1) the message will be rejected and a notification will be sent back, (2) the oldest message will be purged and replaced with the new one (FIFO), or (3) a combination of both 1 and 2 where, if read messages exist they are purged and replaced other wise the new message is rejected.

3.8.3.4.1. Basic Flow

1. Message arrives at a service provider's mail server.
2. After determining the recipient of the message, the mail server will check the number of messages, or storage space, in the respective inbox to determine availability.
3. If the number of messages, or storage space, is below a certain limit, the message is deposited into the inbox.
4. Otherwise, the mail server will reject the message sending a notification to the sender and keeping a note to the owner.
5. The owner logs into the system using her/his mail client. Client is configured to delete messages on download.
6. As messages are downloaded or read, they may be deleted from the server.

3.8.3.4.2. Basic Flow Requirements

These are the basic flow requirements for the e-mail use case:

1. Mediation shall support recording IPDRs that are not service consumer initiated.

3.8.3.4.3. Basic Flow Usage Attribute List

3.8.3.4.3.1. Service Consumer

Category	Usage Attribute Name	Data Type	Presence	Possible Values	Remarks
Who	UserLoginName	String	Required		This should identify a unique user in

Category	**Usage Attribute Name**	**Data Type**	**Presence**	**Possible Values**	**Remarks**
					the system. Real time mapping of dynamically allocated IP addresses might be necessary.
Where	UserLoginLocation	String	Required		This could also be included under the where section. However, it is noted here to recognize the difference between the location of the service provider and the service consumer.

3.8.3.4.3.2. Service Provider

Category	**Usage Attribute Name**	**Data Type**	**Presence**	**Possible Values**	**Remarks**
Where	ProviderName	String	Required		This is the actual provider of the service.
Where	ProviderLocation	String	Required		This is the location of the service provider.

3.8.3.4.3.3. Service Definitions

Category	**Usage Attribute Name**	**Data Type**	**Presence**	**Possible Values**	**Remarks**
What	Service	String	Required		This use case supports only one service type, e-mail, however, a generic type field is need to cover multiple services offered by the same provider.
What	Feature	String	Required		Again, this is included to keep the general notion of having multiple features for each service type. An example could be monitoring e-mail based on messages as a different feature than monitoring based on storage space.
What	ProcessingTime	Integer	Conditional		This is the total time used by the server to process an e-mail.
What	Storage	Integer	Conditional		Is based on the total storage at the time of generating a usage record.
What	StorageDuration	Date time	Conditional		The duration of time messages were stored on the providers servers.
What	BytesTransferred	Integer	Conditional		This is the total bytes transferred either during e-mail arrival or during e-mail download.
What	EventTime	Date time	Conditional		The event took place at this time. Events are generated when a message arrives and when a message is read.

3.8.4. Authentication and Authorization Services

The services described in this section are defined by IETF in a set of RFCs (for RADIUS) and in over 10 Internet Drafts (for DIAMETER and other protocols); the Internet Drafts are still under discussion in the AAA Working Group of IETF (AAA stands for Authentication, Authorization and Accounting) .

Other examples of AAA processes, such as the authentication within the framework of ITU-T's H.323 standard for VoIP, Video over IP, have not directly been used in the creation of this document.

The characteristic of authentication and/or an authorization service ("AA services") is that of enabling other services such as dial-in service.

3.8.4.1. Service Definition

The AAA keywords have been defined by IETF as follows:

- Authentication: The act of verifying the identity of an entity (subject).
- Authorization: The act of determining whether a requesting entity (subject) will be allowed access to a resource (object).
- Accounting: The act of collecting information on resource usage for the purpose of trend analysis, auditing, billing, or cost allocation.

An authentication service and/or an authorization service is a service provided by an IP Network Service Provider to control access to network resources. Apart from authentication and/or authorization the AA services may include configuration of the service the user wants to use and accounting for that service.

The authentication or the authorization process itself may include generation of accounting records.

The following parties are involved in this service:

- The User, i.e. the Service Consumer,
- the Network Access Server (NAS), and
- the AAA Server, or Shared (Authentication, Authorization and Accounting) Server containing a single database of users, which allows for authentication and/or authorization (by verifying user ID and password), for accounting as well as for configuration information detailing the type of service to deliver to the user (for example, SLIP, PPP, telnet, rlogin, VoIP etc.).

3.8.4.2. Service Requirements

- The IPDR must provide unique identification of all parties involved in the service, i.e. the User, the NAS and the AAA Server.
- If there is more than one administrative domain, some of the parties may have proxies; the IPDR must provide unique identification of those as well.
- The IPDR must contain all attributes needed for AA services for mobile wire line and mobile wireless users.
- The set of attributes must reflect a state-of-art AA service protocol such as DIAMETER.

3.8.4.3. Service Usage Attribute List

Category	**Usage Attribute Name**	**Data Type**	**Presence**	**Possible Values**	**Remarks**
What	Type of AAService	String	Required	"Radius", "Diameter", "H.225", "Undefined" etc.	Type of authentication service that is invoked.
When	AARequestTime	Datetime	Required	ISO 8601 time	May be different from AAAcknowledgeTime. This will allow measuring response time.
When	AAAcknowledgeTime	Datetime	Required	ISO 8601 time	Time when authentication and/or authorization granted at NAS
Where	NAS ID	String	Required		Identification or Location of Network Access Server (NAS)
Where	AAA Server ID	String	Required		Identification or Location of Shared (Authentication/Authorization and Accounting) Server
Who	ProviderName	String	Required		Actual provider of the AA service
Who	UserLoginName (= User ID)	String	Required		Identifies a unique user in the system. Real time mapping of dynamically allocated IP addresses might be necessary

This attribute list should be completed by attributes defined in RADIUS and DIAMETER documents; see Internet Draft "Accounting Attributes and Record Formats" by Nevil Brownlee and Alan Blount, 16 June 2000; draft-ietf-aaa-accounting-attribute-04.txt.

In the case of RADIUS the usage is detailed by the following table.

Category	**Usage Attribute Name**	**Data Type**	**Presence**	**Possible Values**	**Remarks**
What	Acct-Status-Type		Required	"Start", "Stop", "Acct-on", "Acct-off"	RADIUS Type Field 40
What	Acct-Input-Volume (octets or packets)		Required		RADIUS Type Field 42 or 47
What	Acct-Output-Volume (octets or packets)		Required		RADIUS Type Field 43 or 48
What	Acct-Session-Id		Required		RADIUS Type Field 44
What	Acct-Session-Time		Required		RADIUS Type Field 46
Why	Acct-Terminate-Cause		Optional		RADIUS Type Field 49

3.8.4.4 Basic Flow (RADIUS Type)

The description in this section is based on the RADIUS protocol (Remote Authentication Dial-in Service; see RFC 2138, RFC 2139 and related documents).

Basic Authentication/Authorization

The RADIUS protocol provides a basic AA(A) service.

In this simplest case, the user is in the same administrative domain as the NAS and the AAA Server.

The user subscribes to an AA service with his/her IP Network Service Provider, and typically gets a user ID and a password.

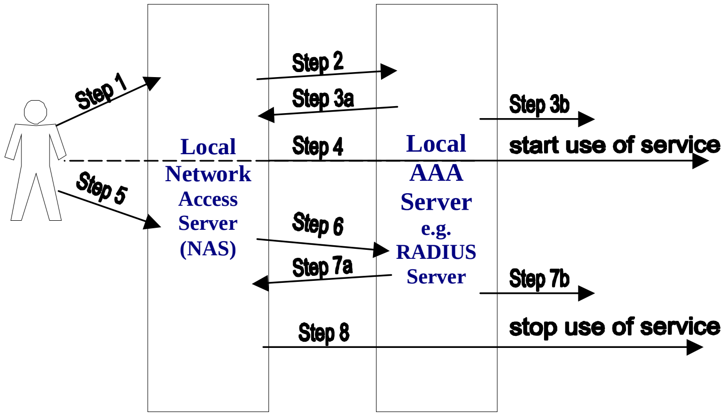
At the start of service delivery the NAS/AAA Server pair may generate an Accounting Start Record.

At the end of service delivery the NAS/AAA Server pair may generate an Accounting End Record.

Client/Server Model: A Network Access Server (NAS) operates as a client of the AAA Server. The client is responsible for passing user information to the designated AAA Server, and then for acting on the response that is returned.

Basic Flow on Start of Service

1) The User logs in and requests a service at the NAS.
2) The authentication request (as sent by the NAS) arrives at the AAA Server
3) The AAA Server acknowledges if the User ID is valid and the password is correct and the user is entitled to use the requested service.
    a) On receiving the acknowledge message the NAS starts the service initially requested by the user.
    b) The AAA Server generates a Accounting Start Record.
4) Start use of service.



Attributes for Basic Flow on Start of Service

See table above.

Basic Flow on End of Service

5. The User requests termination of the service at the NAS.
6. The termination request arrives at the AAA Server
   a) The AAA Server acknowledges receipt of the termination request, and
   b) generates an Accounting End Record containing various parameter describing the effective use of the service (end time, duration, volume etc.).
7. NAS stops the service as requested by the user.

Attributes for Basic Flow on End of Service

See Table 1 above. In addition to the attributes listed in Table 1, there will be attributes describing the billable usage specific to the type of service consumed. Refer to Table 2 above for an example of the additional specific attributes for describing billable usage for a dial-in service over the IP network using the RADIUS protocol.

3.8.4.5 Basic Flow (DIAMETER Type)

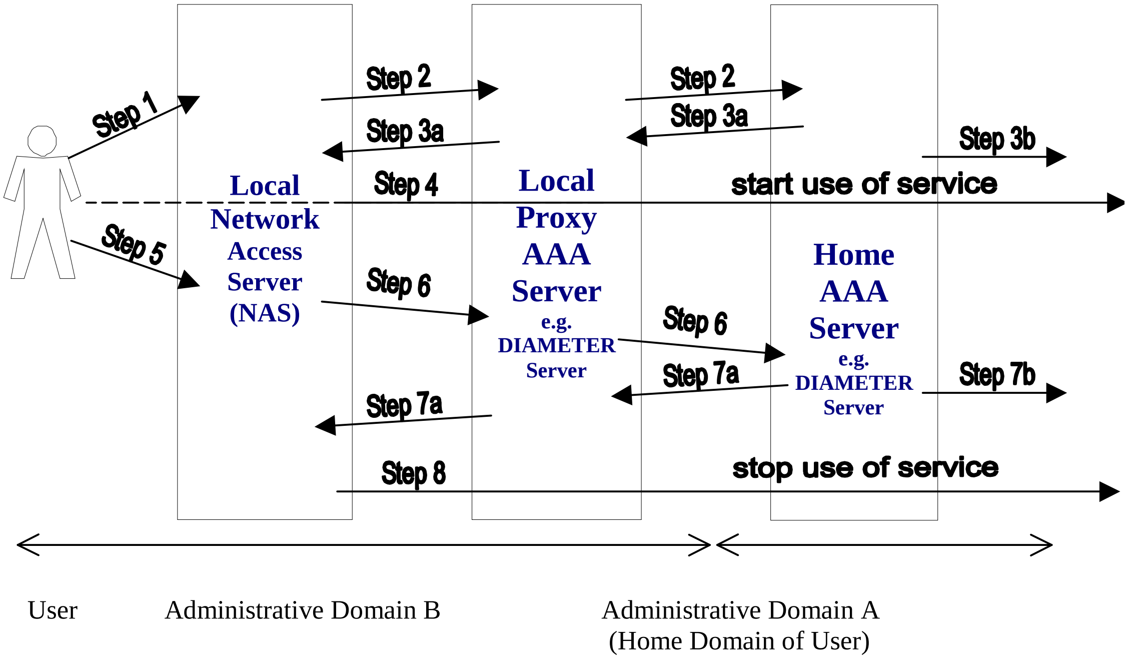
An extension of the RADIUS protocol, the DIAMETER protocol has been proposed and is now discussed in the IETF working groups (see the considerable number of pertinent Internet Drafts dated July 2000 and earlier). DIAMETER is the attempt to improve on most of the shortcomings of the RADIUS protocol. Furthermore, an additional objective is to accommodate roaming aspects of mobile wire line and wireless IP services.

Alternate Authentication/Authorization

The DIAMETER protocol provides an alternate flow for an AA(A) service.

The user is either in the same administrative domain as the NAS and the AAA Server or may be in another administrative domain (when roaming).

To provide AA services in the case of mobile IP and more than one administrative domain AAA Servers may have proxies in others than their original administrative domains.



DIAMETER Peer Model: A Network Access Server (NAS) operates as a peer of the AAA Server. Among AAA Servers, there is also a peer-to-peer relationship in DIAMETER.

Attributes for Alternate Flow

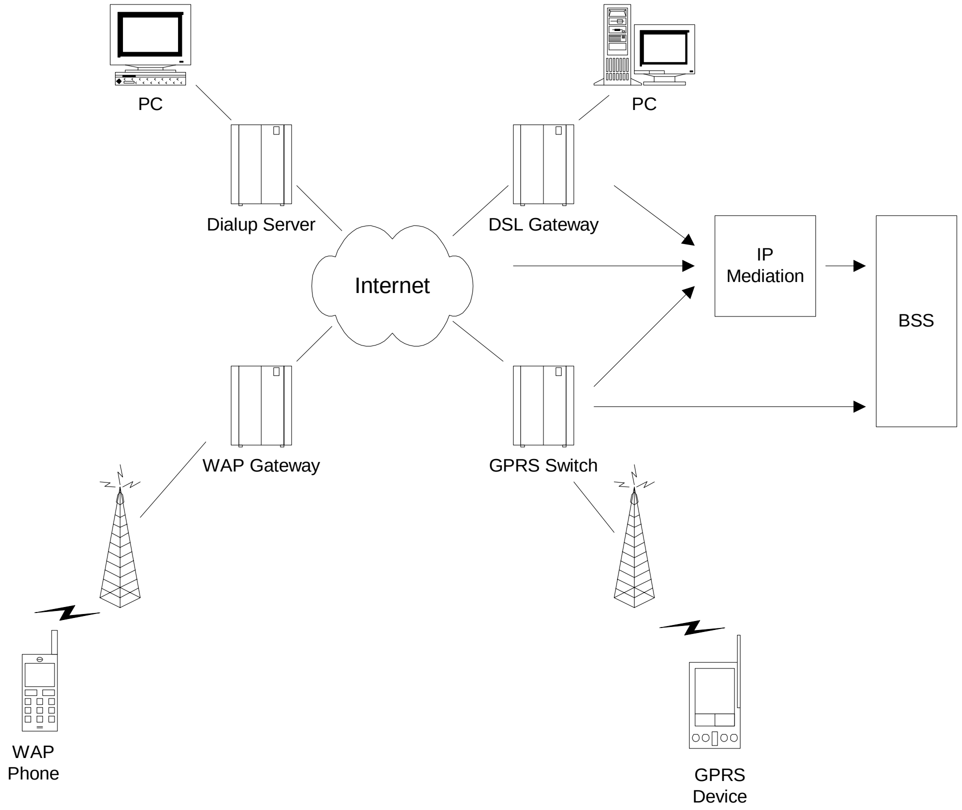
See Internet Draft "Accounting Attributes and Record Formats" by Nevil Brownlee and Alan Blount, 16 June 2000; draft-ietf-aaa-accounting-attribute-04.txt; see the section on DIAMETER Attributes.

3.8.5 Internet Access

Currently, multiple billing models (for example, Sprint's WAP or Palm Net) are used for wirless internet access:

- Flat rate ($20 per month)
- By Volume ($5 per megabyte)
- By Time ($0.2 per minute)
- Free

The goal of the Internet Access IPDR definition is to provide a means to capture the information for current billing models, as well as providing the possibility for future models (for example time combined with QoS, or download volume combined with bandwidth). The Internet Access IPDR is defined in a generic fashion so that it can be used for existing internet access methods, such as fixed-wire dialup or DSL.



3.8.5.1 Service Definition

Internet access is the service of providing access to the internet using one of the following methods or protocols: Modem Dialup, WAP, GPRS, DSL, ISDN, etc

3.8.5.2 Service Requirements

- The Internet Access IPDR must provide all the information necessary so that a user can be billed by flat rate, volume, time or a combination of the above.
- The Internet Access IPDR has to cover mobile as well as stationary internet access services.

3.8.5.3 Usage Attributes List

Category	Usage Attribute Name	Data Type	Presence	Possible Values	Remarks
What	transportProtocol	String	Required	WAP, TCP, PPPoE	Transport protocol that was used for internet access.
What	connectionType	String	Optional	Fixed Wire Dialup, DSL, …	Connection type that was used for internet access.
What	upBandwidth	Value/ Unit	Optional	28 Kb, …	Upstream bandwidth provided.
What	downBandwidth	Value/ Unit	Optional	56 Kb, …	Downstream bandwidth provided.
What	upVolume	Value/ Unit	Optional	257 KB	Volume that was uploaded.
What	downVolume	Value/ Unit	Optional	5 MB	Volume that was downloaded.
What	qosRequested	Number	Optional	0..255	Requested QoS corresponding to the SLA or dynamic QoS request
What	qosDelivered	Number	Optional	0..255	Pre-calculated indicator representing the delivered / negotiated QoS. Physical attributes such as latency or error rates are weighted and combined into one value.
When	startTime	Datetime	Required	ISO 8601 time	When access started.
When	endTime	Datetime	Optional Conditional	ISO 8601 time	When access stopped. At least endTime or duration needs to be present.
When	duration	Value/Unit	Optional Conditional	# of Seconds	Duration of access
Where	accessPoint	String	Required	Dialup number, Gateway IP Address, etc	Identifies access point to the internet. Equivalent to APN for GPRS, NAS
Who	subscriberID	Value/Type	Required	Phone Number, IP Address, Device ID, SIM Card ID, Custom ID	Identifies a unique subscriber in the system. Type can be IMSI, IMEI, IP, PN, CUST
Where	serviceElement	String	Optional	MSCID, CGSN or SGSN (GPRS)	Service element used to provide access. Probably part of IPDR base header.
Who	serviceProviderID	String	Required		Service provider providing internet access.

Category	Usage Attribute Name	Data Type	Presence	Possible Values	Remarks
Where	routingArea	String	Required for cellular only		Subset of location area.
Where	locationArea	String	Required for cellular only		The geographical area from which the connection is established.
Where	cellID	String	Required for cellular only		ID of the cell that is handling the connection to the wireless terminal.
What	serviceBearer	String	Required for cellular only	TDMA, CDMA	Service Bearer that was used for connection protocol.

3.8.6 Content/Service

3.8.6.1 Service Definition

The Content/Service IPDR provides a simple method for accounting of access to content on the web or for execution of simple transactions over the internet. The Content/Service IPDR can be used to bill for access to premium sites and/or for the use of transaction based services (for example directory assistance or stock quotes).

3.8.6.2 Service Requirements

- The Service/Content IPDR must provide all the information necessary so that a user can be billed for access to content and services that are available at a premium.
- The Service/Content IPDR must cover access to content and services from stationary as well as mobile devices.

3.8.6.3 Usage Attributes List

Category	Usage Attribute Name	Data Type	Presence	Possible Values	Remarks
What / Where	ipServiceID	Value/Unit	Required	URL or other service ID	Service address
What	ipService Class	String	Optional		Service classification, for example web site, quote service, directory service etc.
What	numberOfTransactions	Number	Optional		Number of transaction associated with the service.
What	amount	Value/Type	Optional	$5.50	Amount to be charged for service.

Category	Usage Attribute Name	Data Type	Presence	Possible Values	Remarks
Where	routingArea	String	Required for cellular only		Cellular specific information
Where	locationArea	String	Required for cellular only		Cellular specific information
Where	cellID	String	Required for cellular only		Cellular specific information
What	serviceBearer	String	Required for cellular only	TDMA, CDMA	Service Bearer that was used for connection protocol.

3.8.7 Push-Delivery

3.8.7.1 Service Definition

Push-delivery of content is used in the areas of SMS, WAP and broadcast email deliveries. The Push-Delivery IPDR can be used to account for push-delivery transactions.

3.8.7.2 Service Requirements

- The Push-Delivery IPDR must provide the information necessary so that a content provider using push delivery or a recipient of content delivered through push-delivery can be billed for content, volume or other method.
- The Push-Delivery IPDR must cover push delivery to stationary as well as mobile devices.

3.8.7.3 Usage Attributes List

Category	**Usage Attribute Name**	**Data Type**	**Presence**	**Possible Values**	**Remarks**
Who	pushInitiator	Value/Type	Required	IMSI, IMEI, IP, PN, Custom ID	Identity of push initiator
What	deliveryStatus	String	Required	Unconfirmed, succeeded, failed	Status of push delivery
What	deliveryFeatures	String	Optional	Deliver at, after, before	Mode of delivery
When	deliveryTime	DateTime	Required	ISO 8601 time	Time of push delivery or schedule
What	pushID	String	Optional		ID of push message as provided by push initiator.
What	contentType	String	Required	Text, Query, etc	Type of push content
What	contentSize	Value/Unit	Required	200 bytes	Size of push message
What	priority	Number	Optional	1-10	Priority of push message
What	applicationID	String	Optional		ID of receiving application
Who	pushRecipient	Value/Type	Required	Phone Number, IP Address, Device ID, SIM Card ID, Custom ID	Identifies a unique subscriber in the system to receive the push message. Type can be IMSI, IMEI, IP, PN, CUST
What	transportProtocol	String	Required	WAP, TCP, PPPoE	Transport protocol that was used for internet access.
What	connectionType	String	Optional	Fixed Wire Dialup, DSL, …	Connection type that was used for internet access.
Where	serviceElement	String	Optional	MSCID, CGSN or SGSN (GPRS)	Service element used to deliver push message.
Who	serviceProviderID	String	Required		Service provider delivering push message.

Table 17 – Wireless Push-Delivery (Extension to Push-Delivery)

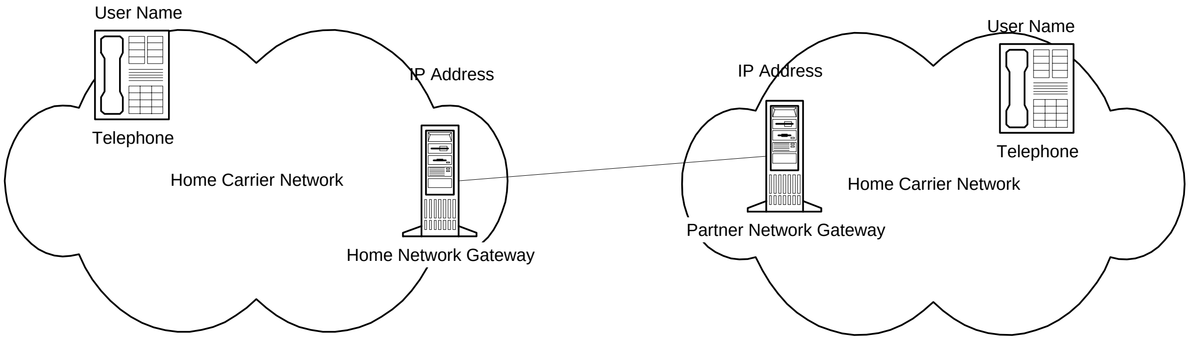
Category	**Usage Attribute Name**	**Data Type**	**Presence**	**Possible Values**	**Remarks**
Where	routingArea	String	Required for cellular only		Cellular specific information
Where	locationArea	String	Required for cellular only		Cellular specific information
Where	cellID	String	Required for cellular only		Cellular specific information
What	serviceBearer	String	Required for cellular only	TDMA, CDMA	Service Bearer that was used for message delivery.

3.8.8 Wholesale requirements across all products covered by IPDR

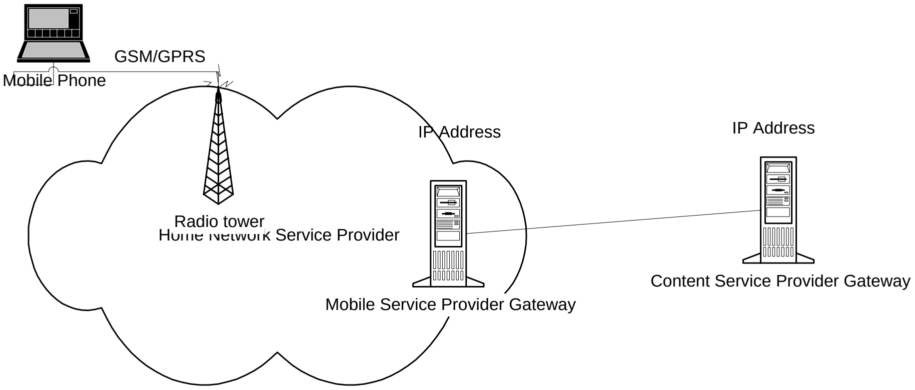
In all services of voice and data more than one service provider may be involved. In such cases the service provider who provides the service charges his partner (another service provider) for the service. The charges are calculated according to the data captured from the system's records during the period. The basic step in wholesale billing is the guiding of the record to the relevant partner.

The requirements here are the same across all products using the TCP/IP protocol, for example:

VoIP - Using VoIP will require both end service providers to communicate over IP. Thus there are origination and destination IP addresses.
Even when user names are used to initiate the calls the service providers must use IP address to communicate the other parties. These IP address can uniquely identified the service providers.



WAP - When this service is used the user connects to a WAP gateway using a cellular phone. The connection between the WAP Gateway (or other server of the service provider) to the content provider is done using IP.

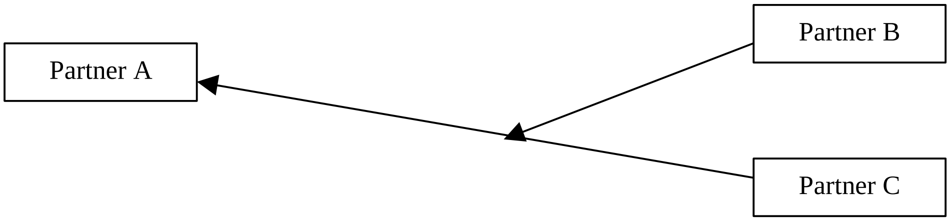

GSM/GPRS
Mobile Phone
Radio tower
Home Network Service Provider
IP Address
Mobile Service Provider Gateway
IP Address
Content Service Provider Gateway

Two approaches can be used for guiding record to the correct partner, the approaches are described below.

Basic flow

Approach 1

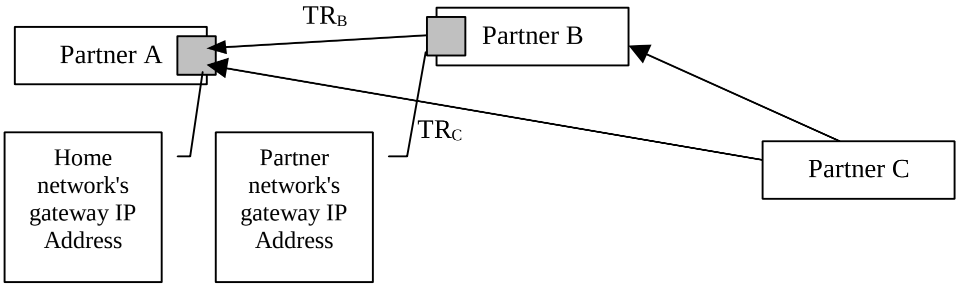
Using the Origination/Destination number to identify the service provider.



Partner A connects to both Partner B and Partner C.
Partner A receives data from Partner B.
Partner A receives data from Partner C.

Approach 2

Using gateway's identification of the closed network.


TR$_B$
Partner A
Partner B
TR$_C$
Home network's gateway IP Address
Partner network's gateway IP Address
Partner C

Partner A connects to both Partner B and Partner C.
Partner C sends data to Partner A directly or through Partner B.
Partner A settles with Partner C on the direct connection and with Partner B on all the data received on TR$_B$ (including data received from Partner C through Partner B).

Basic flow requirements

IPDR must provide unique identifier of the source and destination partner.
IPDR must provide identification of the network of the closed partner. This identifier can be the IP address of the router of the closed partner.

Category	**Usage Attribute Name**	**Data Type**	**Presence**	**Possible Values**	**Remarks**
Who	Origination IP Address	String	Required		
Who	Home Gateway IP Address	String	Required		
Who	Partner's gateway IP Address	String	Required		The partner's gateway that connected to the home service provider.
Where	Destination IP Address	String	Required		

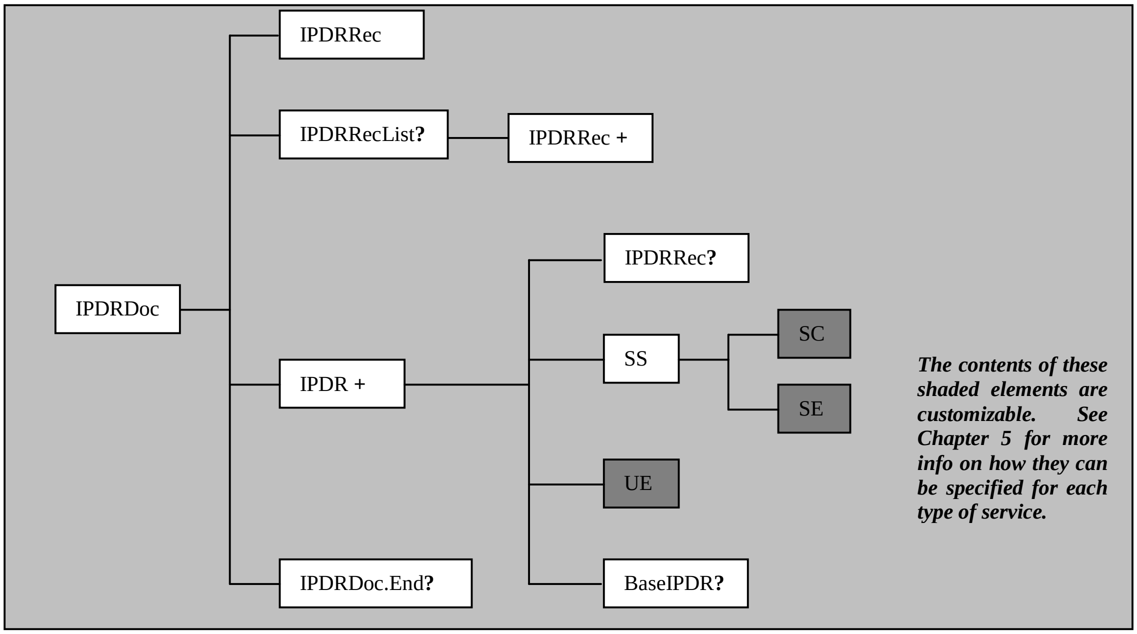
4. Protocol – IPDR Document Structure and Transport

This section contains a complete discussion of protocol; namely, the notation, encoding, message format, procedure and semantics needed to implement the requirements specified in sections 2 and 3.

4.1. IPDR Document Structure

The Diagrams in this section illustrate the IPDR document structure. A single Master IPDR Schema Document declares elements common to all IPDR Services. For each individual IPDR Service a schema document specific to that service is likely to exist. This service-specific schema specifies the data types required to define IPDR elements specific to each IPDR service. For example, an IPDR instance document that corresponds to VideoOnDemand services must account for the Movie Names, however an IPDR instance document corresponding to email services would not. For this reason VideoOnDemand and email services require separate service-specific schema documents.

The IPDR Document hierarchy allows an IPDRDoc to contain many usage records (IPDRs). A usage record is further divided into groupings defining a Service Session (SS) between a Service Consumer (SC) and a Service Element (SE) and a record of the Usage Entry (UE) itself. Details about the consumer, service elements, and usage events are contained in child elements of the SC, SE, and UE elements. The master schema does not include any child elements for the SC, SE, and UE elements, because these details are specific to a particular service and in fact may be different for each service. Section 5 contains a number of service-specific schemas which specify the contents of the SC, SE and UE elements for each of the services described in Section 3.



The contents of these shaded elements are customizable. See Chapter 5 for more info on how they can be specified for each type of service.

The figure above shows the elements under the top-level element, the IPDR document (IPDRDoc). The elements directly below consist of an identifier for the entity recording the usage (IPDRRec) followed by an optional list of additional recording entities (IPDRRecList).

The document's main body is made up of one or more IPDRs that represent single usage events.

The document has an optional ending block of information represented by IPDRDoc.End.

An IPDR then contains a pairing of Service Consumer (SC) and Usage Entry (UE) under the Service Session (SS).

The Service Session (SS) element is merely a structural convenience. It allows repeated pairs of the same SC and SE to be associated via reference. The usage represented may be measured as a discrete event, or part of an ongoing session.

The Usage Entry (UE) element contains data describing metrics or parameters of a specific Usage Entry. It is here that a link to service-specific schema occurs. This is the base type for the UE (Usage Entry) element. The service specific schema can extend this by deriving from it.

Note that, in addition to the elements shown, there are also “reference” elements for IPDRRec, SS, SC and SP elements. Because reference items are functionally equivalent to explicit elements, they have been omitted from the diagram for clarity.

4.1.1. IPDR Master Schema

The Master IPDR Schema Document formally describes how all IPDR documents are constructed. Additional details describing element use are included and considered part of the overall IPDR Document specification. Comments do not form part of the specification.

All service-specific schema documents include the Master IPDR Schema Document via the
<include schemaLocation=”…:”/>
construct. Each IPDR instance document refers to one of the service-specific schema documents via the
xsi:schemaLocation="..”
attribute assigned to the <IPDRDoc> element.

The most recent version of the Master IPDR Schema Document is available at:

http://www.ipdr.org/public/ipdr2.0.xsd

The entire Master Schema is presented below and is followed by an annotated section containing further description and restrictions on the various elements. The annotations are considered part of the specification. The service-specific schema and annotations for a Video On Demand example and the actual service specifications for the services defined in Chapter 3 follow in Chapter 5.

```
<?xml version="1.0"?>

<schema xmlns="http://www.w3.org/1999/XMLSchema"
        targetNamespace="http://www.ipdr.org/namespaces/ipdr"
        xmlns:ipdr="http://www.ipdr.org/namespaces/ipdr">

  <element name="IPDRDoc">
    <complexType content="elementOnly">
      <annotation>
        <documentation> The IPDRDoc element is the top-level container
            of a set of IPDRs. The document will also define the entity
            which recorded these IPDRs via the IPDRRec element.
        </documentation>
      </annotation>
      <element ref="ipdr:IPDRRec" minOccurs="1" maxOccurs="1"/>
      <element ref="ipdr:IPDRRecList" minOccurs="0" maxOccurs="1"/>
      <element ref="ipdr:IPDR" minOccurs="1" maxOccurs="unbounded"/>
      <element ref="ipdr:IPDRDoc.End" minOccurs="0" maxOccurs="1"/>
      <attribute name="seqNum" type="integer" use="optional"/>
      <attribute name="version" type="string" minOccurs="1" maxOccurs="1"/>
      <attribute name="startTime" type="timeInstant" use="optional"/>
      <attribute name="info" type="string" use="optional"/>
    </complexType>
  </element>

  <element name="IPDRDoc.End">
    <complexType content="empty">
      <annotation>
        <documentation> The IPDRDoc.End element optionally marks the
            end of the IPDR block. It may contain some check
            information like a count of IPDRs.
        </documentation>
      </annotation>
      <attribute name="count" type="integer" use="optional"/>
      <attribute name="endTime" type="timeInstant" use="optional"/>
    </complexType>
  </element>

  <element name="IPDRRec">
    <complexType content="empty">
      <annotation>
        <documentation> The IPDRRec element describes the entity that
            is responsible for creating (recording) the IPDRDoc.
        </documentation>
      </annotation>
      <attribute name="id" type="ID" use="optional"/>
      <attribute name="startTime" type="timeInstant" use="optional"/>
      <attribute name="info" type="string" use="optional"/>
    </complexType>
  </element>

  <element name="IPDRRecRef">
    <complexType content="empty">
      <annotation>
        <documentation> The IPDRRecRef element may be used to associate
            common references to the same IPDRRec element without
            repeating its other usage attributes.
        </documentation>
      </annotation>
      <attribute name="ref" type="IDREF" use="required"/>
    </complexType>
  </element>

  <element name="IPDRRecList">
    <complexType content="elementOnly">
      <sequence>
        <annotation>
          <documentation> The IPDRRecList identifies contributing IPDR
              recording entities which were used in the construction
              of the current IPDR Document. A typical example use
              would be for an aggregator of IPDR documents to
              identify the set of initial recorders presenting
              IPDRs.
          </documentation>
        </annotation>
        <element ref="ipdr:IPDRRec" minOccurs="1"
            maxOccurs="unbounded"/>
      </sequence>
    </complexType>
  </element>

  <element name="IPDR">
    <complexType content="elementOnly">
      <sequence>
        <annotation>
          <documentation> An IPDR describes an event between a Service
              Consumer (SC) and a Service Element (SE). The SC and SE
              elements are contained beneath an entity called the
              Service Session (SS). Details of the event is contained
              in the Usage Entry (UE) element. All IPDRs have a time
              indicating when the event occurred.
          </documentation>
        </annotation>
        <choice minOccurs="0" maxOccurs="1">
          <element ref="ipdr:IPDRRec"/>
          <element ref="ipdr:IPDRRecRef"/>
        </choice>
        <choice>
          <element ref="ipdr:SS" minOccurs="1" maxOccurs="1"/>
          <element ref="ipdr:SSRef" minOccurs="1" maxOccurs="1"/>
        </choice>
        <element ref="ipdr:UE" minOccurs="1" maxOccurs="1"/>
        <element ref="ipdr:BaseIPDR" minOccurs="0" maxOccurs="1"/>
      </sequence>
      <attribute name="id" type="ID" use="optional"/>
      <attribute name="time" type="timeInstant" use="required"/>
      <attribute name="seqNum" type="integer" use="optional"/>
    </complexType>
  </element>

  <element name="SS">
    <complexType content="elementOnly">
      <sequence>
        <annotation>
          <documentation> The Service Session (SS) element groups the
              Service Consumer and Service Element information. This
              grouping allows an SC/SE pair to be associated with other
              IPDRs via a single reference (the SSRef).
          </documentation>
        </annotation>
        <choice>
          <element ref="ipdr:SC" minOccurs="1" maxOccurs="1"/>
          <element ref="ipdr:SCRef" minOccurs="1" maxOccurs="1"/>
        </choice>
        <choice>
          <element ref="ipdr:SE" minOccurs="1" maxOccurs="1"/>
          <element ref="ipdr:SERef" minOccurs="1" maxOccurs="1"/>
        </choice>
      </sequence>
      <attribute name="id" type="ID" use="optional"/>
      <attribute name="service" type="string" use="optional"/>
    </complexType>
  </element>

  <complexType name="SCType" content="elementOnly" final="restriction">
    <annotation>
      <documentation> This is the base type for the Service Consumer
                      element. The service specific schema can extend
                      this by deriving from it.
      </documentation>
    </annotation>
    <attribute name="id" type="ID" use="optional"/>
  </complexType>
```

```
<element name="SC" type="ipdr:SCType">
  <annotation>
    <documentation> This element describes the Service Consumer.
    </documentation>
  </annotation>
</element>

<complexType name="SEType" content="elementOnly" final="restriction">
  <annotation>
    <documentation> This is the base type for the SE (Service
        Element)element. The service specific schema can extend
        this by deriving from it.
    </documentation>
  </annotation>
  <attribute name="id" type="ID" use="optional"/>
</complexType>

<element name="SE" type ="ipdr:SEType">
  <annotation>
    <documentation> This element describes the Service Element.
    </documentation>
  </annotation>
</element>

<complexType name="UEType" content="elementOnly" final="restriction">
  <annotation>
    <documentation> This is the base type for the UE (Usage Entry)
                    element. The service specific schema can extend
                    this by deriving from it.
    </documentation>
  </annotation>
  <attribute name="type" type="string" use="default" value="Start-Stop">
    <simpleType base="string">
      <enumeration value="Start"/>
      <enumeration value="Stop"/>
      <enumeration value="Start-Stop"/>
      <enumeration value="Interim"/>
    </simpleType>
  </attribute>
</complexType>

<element name="UE" type ="ipdr:UEType">
  <annotation>
    <documentation> This element describes the Usage Entry.
    </documentation>
  </annotation>
</element>

<element name="SSRef">
  <complexType content="empty">
    <annotation>
      <documentation> The SSRef element may be used to associate common
                      references to the same pairing of a Service Consumer
                      and a Service Element.
      </documentation>
```

```
   </annotation>
   <attribute name="ref" type="IDREF" use="required"/>
  </complexType>
 </element>

 <element name="SERef">
  <complexType content="empty">
   <annotation>
    <documentation> The SERef element may be used to associate
      common references to the Service Element.
    </documentation>
   </annotation>
   <attribute name="ref" type="IDREF" use="required"/>
  </complexType>
 </element>

 <element name="SCRef">
  <complexType content="empty">
   <annotation>
    <documentation> The SCRef element may be used to associate
      common references to the Service Consumer.
    </documentation>
   </annotation>
   <attribute name="ref" type="IDREF" use="required"/>
  </complexType>
 </element>

 <element name="BaseIPDR">
  <complexType content="empty">
   <annotation>
    <documentation> The BaseIPDR element allows reference to be
      made to IPDRs which contributed to the construction of the
      current IPDR element.
    </documentation>
   </annotation>
   <attribute name="refs" type="IDREFS" use="required"/>
  </complexType>
 </element>

</schema>
```

4.1.2. Annotated IPDR Master Schema

A description of each element in the IPDR Schema Document is presented below.

IPDRDoc

```
<element name="IPDRDoc">
    <complexType content="elementOnly">
      <annotation>
        <documentation> The IPDRDoc element is the top-level container
            of a set of IPDRs. The document will also define the entity
            which recorded these IPDRs via the IPDRRec element.
        </documentation>
      </annotation>
      <element ref="ipdr:IPDRRec" minOccurs="1" maxOccurs="1"/>
      <element ref="ipdr:IPDRRecList" minOccurs="0" maxOccurs="1"/>
      <element ref="ipdr:IPDR" minOccurs="1" maxOccurs="unbounded"/>
      <element ref="ipdr:IPDRDoc.End" minOccurs="0" maxOccurs="1"/>
      <attribute name="seqNum" type="integer" use="optional"/>
      <attribute     name="version"     type="string"     minOccurs="1"
                                               maxOccurs="1" />
      <attribute name="startTime" type="timeInstant" use="optional"/>
      <attribute name="info" type="string" use="optional"/>
    </complexType>
</element>
```

The attributes of the IPDRDoc element are described below.

- seqNum - an integer value for auditing sets of IPDRs.
- version - identifies the version of the Master IPDRDoc Schema being used. This version shall be ‘2.0’.
- startTime - indicates the time the recorder began producing this document. (See the “Additional Element Details” subsection for more information about timestamps”)
- info - optional describing string for this document.

IPDRDoc.End

```
<element name="IPDRDoc.End">
    <complexType content="empty">
      <annotation>
        <documentation> The IPDRDoc.End element optionally marks the
            end of the IPDR block. It may contain some check
            information like a count of IPDRs.
        </documentation>
      </annotation>
      <attribute name="count" type="integer" use="optional"/>
      <attribute name="endTime" type="timeInstant" use="optional"/>
    </complexType>
  </element>
```

The attributes of the IPDRDoc.End element are described below.

- count - the number of IPDRs contained in this document (used as a check). This includes IPDRs contained in tables.
- endTime - the time the IPDR recorder finished creating this document. (See the “Additional Element Details” subsection for more information about timestamps”)

IPDRRec

```
<element name="IPDRRec">
    <complexType content="empty">
      <annotation>
        <documentation> The IPDRRec element describes the entity that
            is responsible for creating (recording) the IPDRDoc.
        </documentation>
      </annotation>
      <attribute name="id" type="ID" use="optional"/>
      <attribute name="startTime" type="timeInstant" use="optional"/>
      <attribute name="info" type="string" use="optional"/>
    </complexType>
  </element>
```

The attributes of the IPDRRec element are described below.

- id - a unique identifier within this document. It allows subsequent IPDRs to repeat a common IPDR recorder by reference.
- startTime - indicates the time this IPDR recording entity began running. The same entity may have produced multiple documents during its run. (See the “Additional Element Details” subsection for more information about timestamps”)
- info - optional describing string for this document.

IPDRRecRef

```
<element name="IPDRRecRef">
    <complexType content="empty">
      <annotation>
        <documentation> The IPDRRecRef element may be used to associate
            common references to the same IPDRRec element without
            repeating its other usage attributes.
        </documentation>
      </annotation>
      <attribute name="ref" type="IDREF" use="required"/>
    </complexType>
 </element>
```

The attributes of the IPDRRecRef element are described below.

- ref - a reference to an IPDRRec element defined earlier in the document. Note that implicitly all IPDRs were recorded by the IPDR recorder described at the top of the IPDRDoc. Individual IPDRs or tables may optionally identify a different recorder.

IPDRRecList

```
<element name="IPDRRecList">
    <complexType content="elementOnly">
      <sequence>
        <annotation>
          <documentation> The IPDRRecList identifies contributing IPDR
              recording entities which were used in the construction
              of the current IPDR Document. A typical example use would
              be for an aggregator of IPDR documents to
              identify the set of initial recorders presenting
              IPDRs.
          </documentation>
        </annotation>
        <element ref="ipdr:IPDRRec"
            minOccurs="1"
            maxOccurs="unbounded"/>
      </sequence>
    </complexType>
 </element>
```

IPDR

```
<element name="IPDR">
    <complexType content="elementOnly">
      <sequence>
        <annotation>
          <documentation> An IPDR describes an event between a Service
              Consumer (SC) and a Service Element (SE). The SC and
              SE elements are contained beneath an entity called
              the Service Session (SS). Details of the event are
              contained in the Usage Entry (UE) element. All IPDRs
              have a time indicating when the event occurred.
          </documentation>
        </annotation>
        <choice minOccurs="0" maxOccurs="1">
          <element ref="ipdr:IPDRRec"/>
          <element ref="ipdr:IPDRRecRef"/>
        </choice>
        <choice>
          <element ref="ipdr:SS" minOccurs="1" maxOccurs="1"/>
          <element ref="ipdr:SSRef" minOccurs="1" maxOccurs="1"/>
        </choice>
        <element ref="ipdr:UE" minOccurs="1" maxOccurs="1"/>
        <element ref="ipdr:BaseIPDR" minOccurs="0" maxOccurs="1"/>
      </sequence>
      <attribute name="id" type="ID" use="optional"/>
      <attribute name="time" type="timeInstant" use="required"/>
      <attribute name="seqNum" type="integer" use="optional"/>
    </complexType>
 </element>
```

The attributes of the IPDR element are described below.

- id - a unique identifier for this IPDR in order to support the BaseIPDR element.
- time - the time the recorded usage event occurred. (See the “Additional Element Details” subsection for more information about timestamps”)
- seqNum - an optional integer value for auditing sets of IPDRs. *(ed note: additional work on numbering policy req'd)*

SSRef

```
<element name="SSRef">
    <complexType content="empty">
      <annotation>
        <documentation> The SSRef element may be used to associate
            common references to the same pairing of a Service Consumer
            and a Service Element.
        </documentation>
      </annotation>
      <attribute name="ref" type="IDREF" use="required"/>
    </complexType>
  </element>
```

The attributes of the SSRef element are described below:

- ref - a reference to an SS element that contains the same set of Consumer and Element identifiers.

SS

```
<element name="SS">
    <complexType content="elementOnly">
      <sequence>
        <annotation>
          <documentation> The Service Session (SS) element groups the
              Service Consumer and Service Element information. This
              grouping allows an SC/SE pair to be associated
              with other IPDRs via a single reference (the SSRef).
          </documentation>
        </annotation>
        <choice>
          <element ref="ipdr:SC" minOccurs="1" maxOccurs="1"/>
          <element ref="ipdr:SCRef" minOccurs="1" maxOccurs="1"/>
        </choice>
        <choice>
          <element ref="ipdr:SE" minOccurs="1" maxOccurs="1"/>
          <element ref="ipdr:SERef" minOccurs="1" maxOccurs="1"/>
        </choice>
      </sequence>
      <attribute name="id" type="ID" use="optional"/>
      <attribute name="service" type="string" use="optional"/>
    </complexType>
  </element>
```

The attributes of the SS element are described below.

- id - a unique identifier within this document. It allows subsequent IPDRs to repeat a common Service Session by reference.
- service – a name indicating the service being provided for this session. This may aid in classification of IPDR records in a document (e.g. HTTP vs. IMAP services).

SC

```
<element name="SC" type="ipdr:SCType">
    <annotation>
      <documentation> This element describes the Service Consumer.
      </documentation>
    </annotation>
</element>
```

SCType

```
<complexType name="SCType" content="elementOnly" final="restriction">
    <annotation>
      <documentation> This is the base type for the Service Consumer
          element. The service specific schema can extend this by
          deriving from it.
      </documentation>
    </annotation>
    <attribute name="id" type="ID" use="optional"/>
</complexType>
```

The attributes of the SCType element are described below.

- id - a unique identifier within this document. It allows subsequent IPDRs to repeat a common Service Consumer by reference.

SE

```
<element name="SE" type ="ipdr:SEType">
    <annotation>
      <documentation> This element describes the Service Element.
      </documentation>
    </annotation>
 </element>
```

SEType

```
<complexType name "SEType" content="elementOnly" final="restriction">
    <annotation>
        <documentation> This is the base type for the SE (Service
            Element) element. The service specific schema can extend
            this by deriving from it.
      </documentation>
    </annotation>
    <attribute name="id" type="ID" use="optional"/>
  </complexType>
```

The attributes of the SEType element are described below.

- id - a reference to an SE element that contains a Service Element Identifier.

UE

```
<element name="UE" type ="ipdr:UEType">
    <annotation>
        <documentation> This element describes the Usage Entry.
        </documentation>
    </annotation>
</element>
```

UEType

```
<complexType name="UEType" content="elementOnly" final="restriction">
    <annotation>
        <documentation> This is the base type for the UE (Usage
            Entry) element. The service specific schema can extend
            this by deriving from it.
      </documentation>
    </annotation>
    <attribute name="type" type="string" use="default"
        value="Start-Stop">
        <simpleType base="string">
            <enumeration value="Start"/>
            <enumeration value="Stop"/>
            <enumeration value="Start-Stop"/>
            <enumeration value="Interim"/>
        </simpleType>
    </attribute>
  </complexType>
```

The attributes of the UEType element are described below.

- id – a unique identifier within this document. It allows subsequent IPDRs to repeat a common Service Element by reference.

The UE element has the following attributes:

- type – indicates the type of event recorded. These may be Start, Stop or Interim events, or a Start-Stop. Start and Stop indicate the beginning or end of a service being delivered. Interim events provide updated metrics on an ongoing activity. Start-Stop events describe an entire service delivery.
- name – provides additional qualification of the type field.

SCRef, SERef

```
<element name="SCRef">
    <complexType content="empty">
      <annotation>
        <documentation> The SCRef element may be used to associate
            common references to the Service Consumer.
        </documentation>
      </annotation>
      <attribute name="ref" type="IDREF" use="required"/>
    </complexType>
  </element>
```

The attributes of the SCRef and SERef element are described below.

- ref - a reference to an SC or SE element that contains the same set of identifiers.

BaseIPDR

```
<element name="BaseIPDR">
    <complexType content="empty">
      <annotation>
        <documentation> The BaseIPDR element allows reference to be
            made to IPDRs which contributed to the construction of the
            current IPDR element.
        </documentation>
      </annotation>
      <attribute name="refs" type="IDREFS" use="required"/>
    </complexType>
  </element>
```

The attributes of the BaseIPDR element are described below.

- refs - a list of references to IPDRs that contributed to the formation of this IPDR. [*Ed note: additional work on how references are constructed is required*

4.2. Transport Protocol State Machine Model

4.2.1 Introduction

The following state diagrams depict the states and transitions used by the IPDR Transfer Protocol.

The IPDR Transfer Protocol address two models of delivery:

- IT Push, where the IPDR Transmitter is responsible for delivering IPDR documents to a known BSS system.
- BSS Pull, where a BSS system requests specific documents from an IT.

For each transfer scenario, there is a state diagram associated with the IT and another associated with the BSS.

The state diagram attempts to provide a protocol neutral way to depict the behavior of the IT and BSS during exchanges of IPDR documents.

The subsequent section "IPDR and SOAP Mapping Specifics" provides a concrete protocol implementing the behavior depicted in these diagrams.

Reading the State Diagrams

Each diagram depicts a set of states and the transition arcs between states are labeled with the Stimulus ("S") which causes a state transition and the Response ("R") to the stimulus.

All of the state machines begin in the Start state. Most state machines do not define a terminal state, as normal operation of these entities is to continue indefinitely.

State Definitions

Start

This is the starting point for each state machine. When initialization of the system is complete, represented by the stimulus "startup", this state is left. Systems only return to the Start state by some externally defined restart mechanism.

Finish

This represents a terminal state. It represents a BSS system completing an "ad hoc query" against an IT system. In this scenario, the pulling system only consumes some predefined set of IPDR documents.

Await Synch

A system is in this state, while it is attempting to agree with its peer on the appropriate sequence number for the IPDR document.

Idle

A system is in this state when it is synchronized with its peer, but no IPDR documents are currently in transit between the systems.

Await Response

A system is in this state when an IPDR document has been transmitted or requested, but acknowledgement from the peer has not yet been received.

Await Wakeup

An IPDR Transmitter will enter this state if it loses communication with its peer or is informed by the peer that documents cannot be received. This state can be exited by either peer successfully exchanging resynchronizing information.

Constrained

This state is entered by a BSS system that is not capable of processing an inbound IPDR document. Typically, this is due to some resource limitation on the part of the BSS.

Data Gap

This state is entered by a BSS system which detects that the IT system no longer has an expected IPDR document. This can happen when a BSS is down for a prolonged period of time and aging policies on the IT side result in an IPDR document being removed from the IT system.

Some operator intervention is required in order to allow the two peer systems to proceed.

Stimulus and Responses

The majority of the stimulus and responses in the system are messages: push or pull requests or responses or synchronization request or response.

In addition to the protocol messages, the systems maintain timers in states containing the word "Await". Timer expiration is the second most common stimulus.

The duration of the timers is an implementation matter and is not defined by this protocol.

The state machine also has some decisions keyed off the IPDR Document sequence number specified. The sequence numbers allow the IT and BSS to maintain synchronization of the documents transferred.

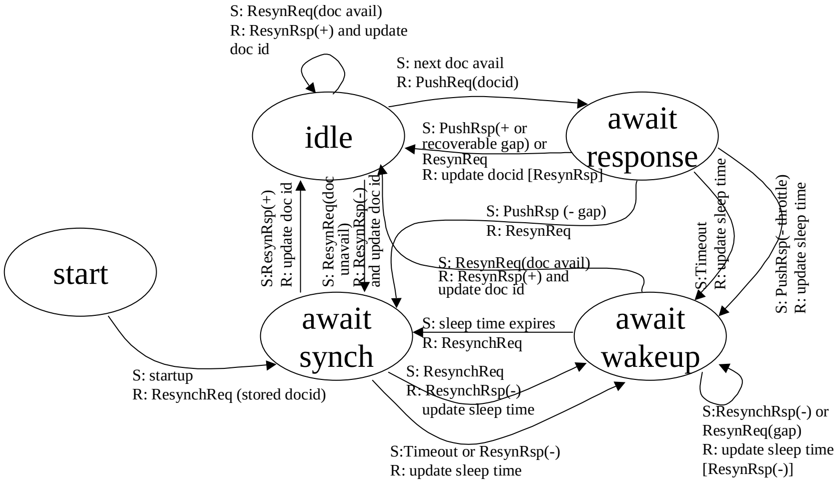
The following assumptions apply to the operation of an IPDR Transmitter (IT) forwarding IPDRDocs to a BSS (for the purposes of this discussion, the term BSS is used to refer to any target system with some established business need to receive IPDRDocs):

- There exists a universe of IPDRDocs (either available from an IPDR Recorder (IR) at the B interface or from an IPDR Store (IS) at the C interface) various subsets of which are required to be transmitted (based on bilaterally-agreed business rules) to one or more BSSs between a given IT and each BSS with which a business relationship exists. This implies that the IPDRDoc IDs, which are ungapped when coming from the IT or IS, may contain gaps (but not duplicates) in the subsets delivered to various BSSs, depending on the business rules.

- The protocol messages exchanged between an IT and each BSS shall be sequentially numbered in a monotonically increasing sequence local to that specific relationship. Gaps and duplicates will be considered error conditions and the protocol must address them. The message sequence number shall be a 64 bit unsigned positive integer, starting at 0 and rolling back to zero when the number space of the field is exhausted. The size of the field is such that no practical chances of duplicate numbers, due to a numerical "wrap around" effect, being detected by the recipient is likely.
- An apparent gap condition may be detected upon initial establishment of a "session" between an IT and a given BSS, or in the case of an interruption of communications between the two entities. The protocol will allow the two entities to "agree" on the next Message Number to be expected. The IT shall maintain "memory" of the last Message Number in the sequence associated with each BSS to which it is delivering IPDRDocs.

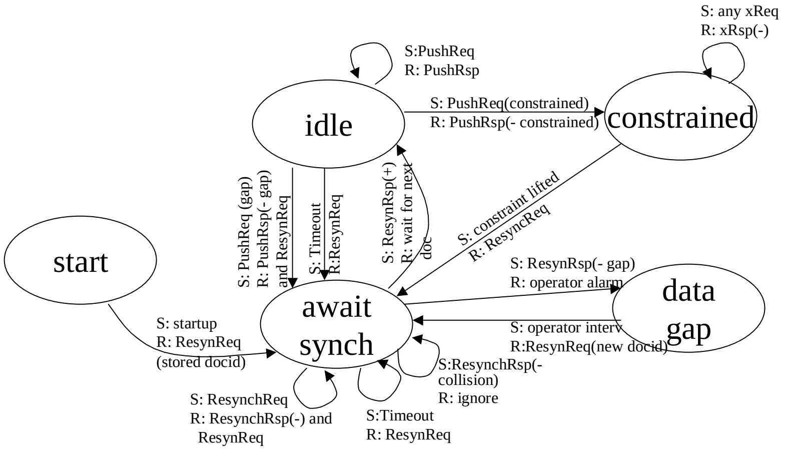
4.2.1. Push Scenario – IT View

IT states (IT Push Scenario)

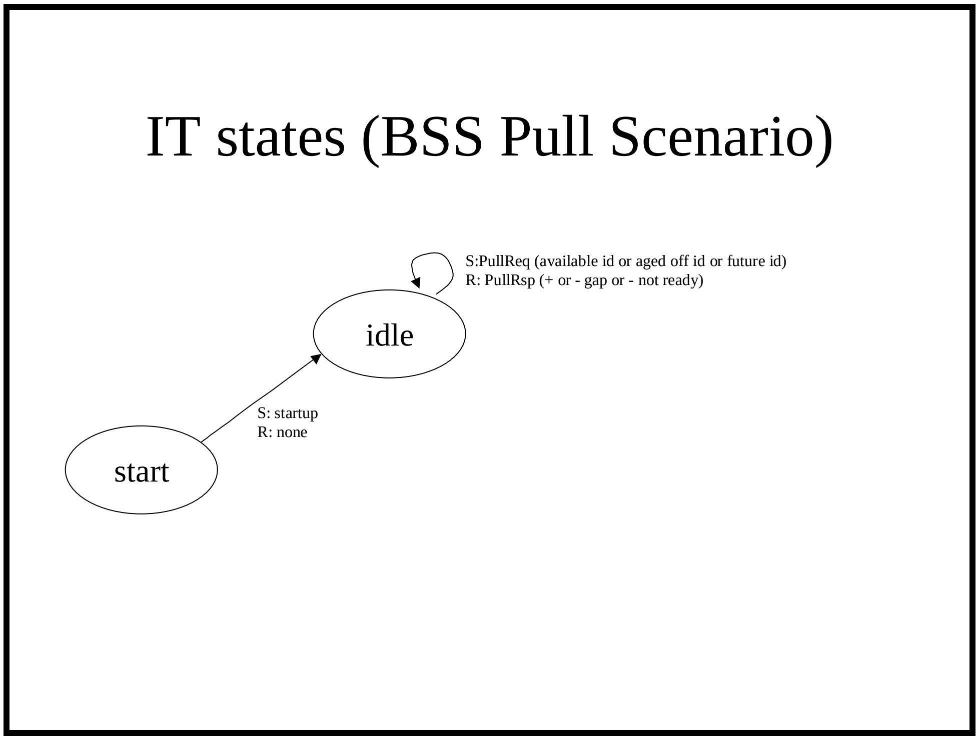


4.2.2. Push Scenario – BSS View

BSS states (IT Push Scenario)



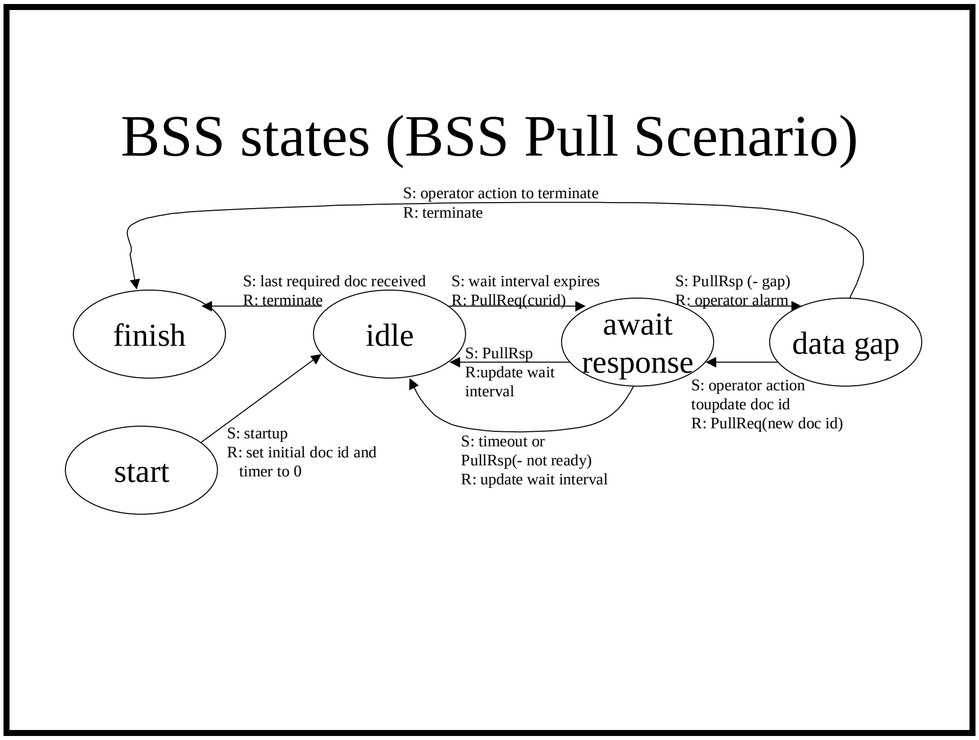
4.2.3. Pull Scenario – IT View



IT states (BSS Pull Scenario)

S:PullReq (available id or aged off id or future id)
R: PullRsp (+ or - gap or - not ready)
idle
S: startup
R: none
start

4.2.4. Pull Scenario – BSS View



BSS states (BSS Pull Scenario)

4.3. IPDR and SOAP Mapping Specifics

This section specifies an IPDR usage document transfer protocol corresponding to the "D-interface" outlined in the basic IPDR network model.

The Simple Object Access Protocol (SOAP) 1.1 is specified as the underlying transport mechanism for performing these transfers. The URL for the SOAP version used in this document is:

http://www.w3.org/TR/2000/NOTE-SOAP-20000508

4.3.1. Rationale

SOAP was chosen in the interest of utilizing emerging standards rather than recreating existing work.

In particular, SOAP specifies a means to map an XML based transfer protocol on top of HTTP. HTTP is attractive because of its ubiquity, affinity to XML (perhaps more emotional than technical) and its ability to easily traverse firewalls.

The SOAP specification is more than a mapping of XML documents to HTTP. SOAP is intended as an alternative means for remote procedure invocations similar to those offered by COM, CORBA or RMI, but with XML based serializations.

The use of SOAP by IPDR utilizes the concept of remote procedures, but one of the parameters in an IPDR exchange is the IPDR document itself.

Unfortunately, the base SOAP specification focuses on the transmission of basic data types such as integers, floats and strings, but not intact XML documents. Therefore, this convention is currently defined by IPDR.

The SOAP 1.1 specification does not preclude the usage described here, nor does it require the exclusive use of its defined RPC mechanism. Specifically the typing mechanism described in Section 5 "SOAP Encoding", is declared as optional in section 4.1.1, "SOAP encodingStyle Attribute".

4.3.2. Mapping Details

Section 4 of the SOAP specification describes the structure of SOAP messages. The IPDR specific aspects related to the various subsections are described here:

- The Envelope element will use the encoding Style attribute (see 4.1.1) defined by the URI:
  "http://www.ipdr.org/soap/encoding/"

- No Header elements as defined by 4.2 are required for IPDR.
- Body elements will be namespace qualified with the URL:

  "http://www.ipdr.org/namespaces/soap"

4.3.3. IPDR Transfer Operations

The IPDR Transport mechanism defines the following immediate child elements of the SOAP Body element:

- PushReq
- PushRsp
- ResynchReq
- ResynchRsp
- PullReq
- PullRsp

These correspond to three basic request response operations.

All names are from the http://www.ipdr.org/namespaces/soap namespace.

Negative responses are carried on SOAP Faults (section 4.4).

4.3.4. Data Types

The following data types are used by the IPDR transport operations and are defined by "XML Schema Part 2: Datatypes:

- int
- string
- uriReference

In addition, IPDR documents are a valid data type. An IPDR document is introduced with the IPDRDoc element name from the http://www.ipdr.org/namespaces/ipdr namespace.

4.3.5. IPDR Transfer Operation Parameters

Six transfer operations can be passed between the IPDR Transmitter (IT) and the BSS that is consuming the IPDRs: PushReq, PushRsp, PullReq, PullRsp, ResyncReq, ResyncRsp. Each of these transfer operations has a set of well defined parameters. Each parameter is carried either in an IPDR specified body element, or in the case of a negative response, a SOAP specified Fault element (see section 4.4.1). The following sub-sections list the parameters required for each transfer operation, as well as the element names and their corresponding types.

4.3.5.1. PushReq

- Description: This operation is used by the IT to deliver an IPDRDoc to the BSS.

- Parameters:

  - **seqNum** - a sequence number indicating the order of the document being sent by the IT. These numbers begin at 1 for the first document transferred to the BSS and increase by one for each subsequent document. There is a one to one correspondence between sequence numbers and IPDRDoc sequence numbers. Sequence numbers span the association between an IT and a BSS. This means that if either entity restarts, the sequence numbers do not reset to 1, but rather pick up where they were previously. The Resynch operations are used to reestablish the appropriate sequence number between an associated pair of IT and BSS.

  - **requestorId** - uniquely identifies the IT sending the IPDRDoc. These should be URL’s that can be used to access the individual entity. These are of the form:

    http://<host>[:<port>]/<entity_name> E.g. http://bss1.xyz.com:6000/voip_bss.

  - **targetId** - uniquely identifies the entity that is targeted with this message. This identifier should match the <entity_name> portion of the target's URL. E.g. voip_it.

  - **IPDRDoc** - the IPDRDocument itself is carried in this parameter.

4.3.5.1.1. PushReq Example

```
POST /IPDRDocs HTTP/1.1
       Host: bss1.xyz.com
       Content-Type: text/xml; charset="utf-8"
       Content-Length: nnnn
       SOAPAction: "http://www.ipdr.org/soap"

       <SOAP-ENV:Envelope
        xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
        SOAP-ENV:encodingStyle="http://www.ipdr.org/soap/encoding">
         <SOAP-ENV:Body>
          <m:PushReq
            xmlns:m="http://www.ipdr.org/namespaces/ipdr">
             <seqNum>2921</seqNum>
             <requestorId>http://ipdr_it1.xyz.com:6000/voip_it</requestorId>
             <targetId>voip_bss</targetId>

<!-- Indentation level reset to fit on page -->
<d:IPDRDoc xmlns:d="http://www.ipdr.org/namespaces/ipdr"
    xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
    xsi:schemaLocation="http://www.ipdr.org/namespaces/ipdr VideoOnDemand1.xsd
    seqNum="2921"
    version="2.0">

 <IPDRRec info="aggregator.ipdr.org"/>
 <IPDR seqNum="1" time="2000-02-01T07:37:00Z">
  <SS id="ses10" service="RTSP">
   <SC xsi:type="SC-VOD-Type">
    <subscriberId>Joe Blow</subscriberId>
    <ipAddress>192.168.1.10</ipAddress>
   </SC>
   <SE xsi:type="SE-VOD-Type">
    <hostName>rtsp.ipdr.org</hostName>
   </SE>
  </SS>
  <UE xsi:type="UE-VOD-Type">
   <movieName>Rocky CIX</movieName>
   <startTime>1999-12-31T23:59:00Z</startTime>
   <endTime>2000-01-01T04:15:00Z</endTime>
   <numAudioStreams>1</numAudioStreams>
   <numVideoStreams>1</numVideoStreams>
   <terminationStatus>normal</terminationStatus>
  </UE>
 </IPDR>
</d:IPDRDoc>

<!-- Indentation level restored -->
          </m:PushReq>
         </SOAP-ENV:Body>
       </SOAP-ENV:Envelope>
```

4.3.5.2. PushRsp

- Description: This operation is sent to acknowledge the receipt of a PushReq. It may indicate a positive or negative status.

- Parameters:

  - For positive acknowledgements there are no parameters needed. (Note that this presumes SOAP as the transport, which distinguishes between positive and negative responses via the "FAULT" mechanism.)

  - For negative acknowledgements the following parameters are used:

    - **reason** - a numeric identifier indicating the reason the request was rejected. The current reason codes are:

      1. A gap was detected. Meaning the proposed next sequence number is either unavailable or is not expected. This usually arises when the pushing entity has no more documents to send at the time.
      2. throttle - this message is sent to indicate that the BSS is not ready to receive additional messages at this time.

    - **delayHint** - specifies the number of milliseconds the throttling party would like the recipient to wait before attempting another synch. (Optional) This is a hint and the recipient may chose to ignore this value.

seqNumHint - specifies the sequence number that the BSS expected in the Push. This informational only. Subsequent Resynch operations will establish the agreed sequence number.

4.3.5.2.1. Positive PushRsp Example

```
HTTP/1.1 200 OK
        Content-Type: text/xml; charset="utf-8"
        Content-Length: nnnn

        <SOAP-ENV:Envelope
         xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Body>
            <m:PushRsp xmlns:m="http://www.ipdr.org/public/namespaces/ipdr">
          </SOAP-ENV:Body>
        </SOAP-ENV:Envelope>
```

4.3.5.2.2. Negative PushRsp (Throttle) Example

```
HTTP/1.1 500 Internal Server Error
        Content-Type: text/xml; charset="utf-8"
        Content-Length: nnnn

        <SOAP-ENV:Envelope
         xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
          <SOAP-ENV:Body>
            <SOAP-ENV:Fault>
              <faultcode>SOAP-ENV:Server</faultcode>
              <faultstring>Server Error</faultstring>
```

```
                <detail>
                  <m:PushRsp xmlns:m="http://www.ipdr.org/ipdr">
                   <reason>2</reason>
                   <delayHint>20000</delayHint>
                  </m:PushRsp>
                </detail>
              </SOAP-ENV:Fault>
            </SOAP-ENV:Body>
          </SOAP-ENV:Envelope>
```

4.3.5.3. ResynchReq

- Description: This operation is sent by either the IT or BSS to reestablish the next document to be pushed.

- Parameters:

  - **seqNum** - indicates the sequence number the sender believes corresponds to the next document to transfer.

  - **requestorId** - uniquely identifies the entity requesting the resynchronization. These should be URL's that can be used to access the individual entity. These are of the form:

    http://<host>[:<port>]/<entity_name> E.g. http://bss1.xyz.com:6000/voip_bss.

  - **targetId** - uniquely identifies the entity that is targeted with this message. This identifier should match the <entity_name> portion of the target's URL. E.g. voip_it.

4.3.5.3.1. ResynchReq Example

```
POST /IPDRDocs HTTP/1.1
        Host: ipdr_it1.xyz.com
        Content-Type: text/xml; charset="utf-8"
        Content-Length: nnnn
        SOAPAction: "http://www.ipdr.org/soap"

        <SOAP-ENV:Envelope
         xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
         SOAP-ENV:encodingStyle="http://www.ipdr.org/soap/encoding">
          <SOAP-ENV:Body>
            <m:ResynchReq
              xmlns:m="http://www.ipdr.org/namespaces/ipdr">
               <seqNum>2916</seqNum>
               <requestorId>http://bss1.xyz.com:6000/voip_bss</requestorId>
               <targetId>voip_it</targetId>
            </m:ResynchReq>
          </SOAP-ENV:Body>
        </SOAP-ENV:Envelope>
```

4.3.5.4. ResynchRsp

- Description: This operation is sent to acknowledge the receipt of a ResynchReq. It may indicate a positive or negative status.

- Parameters:

For positive acknowledgements there are no parameters needed. (Note that this presumes SOAP as the transport, which distinguishes between positive and negative responses via the "FAULT" mechanism.)

For negative acknowledgements the following parameters are used:

- **reason** - a numeric identifier indicating the reason the request was rejected. The current reason codes are:

  1. A gap was detected. Meaning the proposed next sequence number is either unavailable or is not expected (IT's send this if unavailable, BSS's send this if they don't expect that seq).
  2. throttle - this message is sent to indicate that the BSS is not ready to receive additional messages at this time.
  3. A resynch collision was detected. Because both parties can send Resynchs, there is the possibility that these messages will be sent simultaneously. In this event, both parties back off with a collision response. The BSS is expected to immediately retransmit, while the IT awaits the new message.

- **delayHint** - specifies the number of milliseconds the throttling party would like the recipient to wait before attempting another synch. (Optional) This is a hint and the recipient may chose to ignore this value.

- **seqNumHint** - specifies the sequence number which the rejecting party expected in the Resynch. This informational only. Subsequent Resynch operations will establish the agreed sequence number.

4.3.5.4.1. ResynchRsp (Positive) Example

```
HTTP/1.1 200 OK
        Content-Type: text/xml; charset="utf-8"
        Content-Length: nnnn

        <SOAP-ENV:Envelope
          xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
           <SOAP-ENV:Body>
              <m:ResynchRsp xmlns:m="http://www.ipdr.org/public/namespaces/ipdr">
           </SOAP-ENV:Body>
        </SOAP-ENV:Envelope>
```

4.3.5.4.2. Negative ResynchRsp (Collision) Example

```
HTTP/1.1 500 Internal Server Error
        Content-Type: text/xml; charset="utf-8"
        Content-Length: nnnn

        <SOAP-ENV:Envelope
          xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
           <SOAP-ENV:Body>
              <SOAP-ENV:Fault>
                 <faultcode>SOAP-ENV:Server</faultcode>
                 <faultstring>Server Error</faultstring>
                 <detail>
                    <m:ResynchRsp xmlns:m="http://www.ipdr.org/ipdr">
                      <reason>3</reason>
                      <delayHint>0</delayHint>
```

```
                </m:ResynchRsp>
              </detail>
            </SOAP-ENV:Fault>
          </SOAP-ENV:Body>
        </SOAP-ENV:Envelope>
```

4.3.5.5. PullReq

- Description: This operation is used by the BSS to request an IPDRDoc from the IT.

- Parameters:

  - **seqNum** - a sequence number indicating the next document desired for this BSS.

  - **requestorId** - uniquely identifies the BSS requesting the IPDRDoc. These should be URL's which can be used to access the individual entity. These are of the form:

    http://<host>[:<port>]/<entity_name> E.g. http://bss1.xyz.com:6000/voip_bss.

  - **targetId** - uniquely identifies the entity which is targeted with this message. This identifier should match the <entity_name> portion of the target's URL. E.g. voip_it.

4.3.5.5.1. PullReq Example

```
POST /IPDRDocs HTTP/1.1
        Host: ipdr_it1.xyz.com
        Content-Type: text/xml; charset="utf-8"
        Content-Length: nnnn
        SOAPAction: "http://www.ipdr.org/soap"

        <SOAP-ENV:Envelope
          xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
          SOAP-ENV:encodingStyle="http://www.ipdr.org/soap/encoding">
           <SOAP-ENV:Body>
             <m:PullReq
               xmlns:m="http://www.ipdr.org/namespaces/ipdr">
                <seqNum>2916</seqNum>
                <requestorId>http://bss1.xyz.com:6000/voip_bss</requestorI
                <targetId>voip_it</targetId>
             </m:PullReq>
           </SOAP-ENV:Body>
        </SOAP-ENV:Envelope>
```

4.3.5.6. PullRsp

- Description: This operation is sent to acknowledge the receipt of a PullReq. It may indicate a positive or negative status.

- Parameters:

  - For positive acknowledgements the following parameters are used:
    - **ipdrDoc** - the IPDRDocument itself is carried in this parameter.
  - For negative acknowledgements the following parameters are used:

- **reason** - a numeric identifier indicating the reason the request was rejected. The current reason codes are:

  1. A gap was detected. Meaning the proposed next sequence number is not present on the IT, because aging policies have caused it to be discarded.

  2. unavailable - this message is sent to indicate that the IT has not yet produced this document.

- **delayHint** - specifies the number of milliseconds the throttling party would like the recipient to wait before attempting another request. (Optional) This is a hint and the recipient may chose to ignore this value.

- **seqNumHint** - specifies the lowest sequence number which the IT has available. This informational only. Subsequent Resynch operations will establish the agreed sequence number.

4.3.5.6.1. Positive PullRsp Example

```
HTTP/1.1 200 OK
        Content-Type: text/xml; charset="utf-8"
        Content-Length: nnnn

        <SOAP-ENV:Envelope
          xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
           <SOAP-ENV:Body>
              <m:PullRsp xmlns:m="http://www.ipdr.org/ipdr">

<!-- Indentation level reset to fit on page -->

<d:IPDRDoc xmlns:d="http://www.ipdr.org/namespaces/ipdr"
      xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
      xsi:schemaLocation="http://www.ipdr.org/namespaces/ipdr/ VideoOnDemand1.xsd”
      seqNum="2916"
      version="2.0">

  <IPDRRec info="aggregator.ipdr.org"/>
  <IPDR seqNum="1" time="2000-02-01T07:00:00Z">
<IPDR seqNum="1" time="2000-02-01T07:00:00Z">
   <SS id="ses10" service="RTSP">
     <SC xsi:type="SC-VOD-Type">
       <subscriberId>Joe Blow</subscriberId>
       <ipAddress>192.168.1.10</ipAddress>
     </SC>
     <SE xsi:type="SE-VOD-Type">
       <hostName>rtsp.ipdr.org</hostName>
     </SE>
   </SS>
   <UE xsi:type="UE-VOD-Type">
     <movieName>Rocky CIX</movieName>
     <startTime>1999-12-31T23:59:00Z</startTime>
     <endTime>2000-01-01T04:15:00Z</endTime>
     <numAudioStreams>1</numAudioStreams>
     <numVideoStreams>1</numVideoStreams>
     <terminationStatus>normal</terminationStatus>
   </UE>
```

```
 </IPDR>
</d:IPDRDoc>

<!-- Indentation level restored -->

          </SOAP-ENV:Body>
        </SOAP-ENV:Envelope>
```

4.3.5.6.2. Example 9. Negative PullRsp (Gap) Example

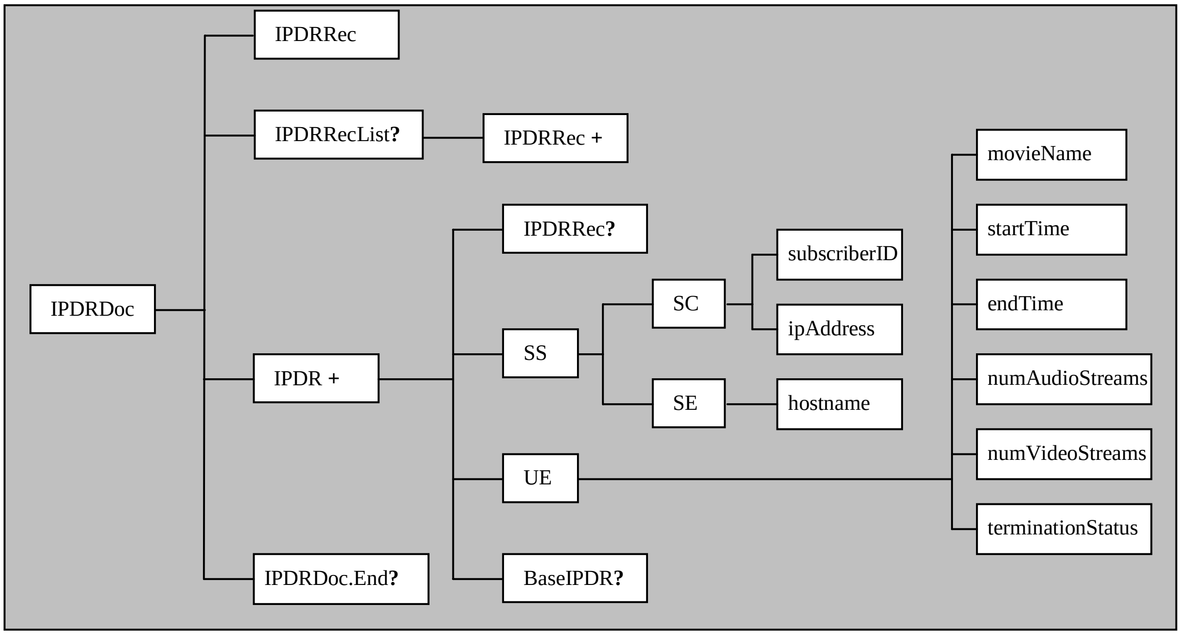
```
HTTP/1.1 500 Internal Server Error
        Content-Type: text/xml; charset="utf-8"
        Content-Length: nnnn

        <SOAP-ENV:Envelope
         xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
          <SOAP-ENV:Body>
            <SOAP-ENV:Fault>
              <faultcode>SOAP-ENV:Server</faultcode>
              <faultstring>Server Error</faultstring>
              <detail>
                <m:PullRsp xmlns:m="http://www.ipdr.org/ipdr">
                 <reason>1</reason>
                 <seqNumHint>2918</seqNumHint>
                </m:PullRsp>
              </detail>
            </SOAP-ENV:Fault>
          </SOAP-ENV:Body>
        </SOAP-ENV:Envelope>
```

5. IPDR Service Definitions

This section details the IPDR Service-Specific Schema Documents and provides example instance documents for each service. These Schema Documents are the formal specification of each service. Any inconsistency with the tabular attribute listings from Chapter 3 shall be resolved by giving the Chapter 5 specifications “normative” precedence.

The entire Service-Specific Schema diagram for Video on Demand is presented



Service Specification Schema Example – Video On Demand (VoD)

5.1.1. VOD Service-Specific Schema

```
<?xml version="1.0"?>
<schema xmlns="http://www.w3.org/1999/XMLSchema"
     targetNamespace="http://www.ipdr.org/namespaces/ipdr"
     xmlns:ipdr="http://www.ipdr.org/namespaces/ipdr">
     <include schemaLocation="http://www.ipdr.org/public/ipdr2.0.xsd"/>
  <complexType name="SC-VOD-Type" base="ipdr:SCType" derivedBy="extension">
     <element name="subscriberId" type="string" minOccurs="1" maxOccurs="1">
        <annotation>
           <documentation>  Subscriber ID.
           </documentation>
        </annotation>
     </element>
     <element name="ipAddress" type="string" minOccurs="1" maxOccurs="1">
        <annotation>
           <documentation>   Subscribers IP address.
           </documentation>
        </annotation>
     </element>
  </complexType>
  <complexType name="SE-VOD-Type" base="ipdr:SEType" derivedBy="extension">
```

```
   <element name="hostName" type="string" minOccurs="1" maxOccurs="1">
     <annotation>
       <documentation>  Name of the host that is providing the service.
       </documentation>
     </annotation>
   </element>
 </complexType>
 <complexType name="UE-VOD-Type" base="ipdr:UEType" derivedBy="extension">
   <element name="movieName" type="string" minOccurs="1" maxOccurs="1">
     <annotation>
       <documentation> Name of film.
      </documentation>
    </annotation>
  </element>
  <element name="startTime" type="timeInstant" minOccurs="1" maxOccurs="1">
    <annotation>
      <documentation> Time instant the film starts playing.
      </documentation>
   </annotation>
  </element>
  <element name="endTime" type="timeInstant" minOccurs="1" maxOccurs="1">
    <annotation>
      <documentation> Time instant the film stops playing.
      </documentation>
    </annotation>
  </element>
  <element name="numAudioStreams" type="nonNegativeInteger" minOccurs="1" maxOccurs="1">
    <annotation>
      <documentation> Count of audio streams.
      </documentation>
    </annotation>
  </element>
  <element name="numVideoStreams" type="nonNegativeInteger" minOccurs="1" maxOccurs="1">
    <annotation>
      <documentation> Count of video streams.
      </documentation>
    </annotation>
  </element>
  <element name="terminationStatus" type="ipdr:terminationStatusType" minOccurs="1"
maxOccurs="1">
    <annotation>
      <documentation> Describes how the session ended.
      </documentation>
    </annotation>
  </element>
 </complexType>
 <simpleType name="terminationStatusType" base="string">
   <enumeration value="normal"/>
   <enumeration value="clientFailure"/>
   <enumeration value="serverFailure"/>
 </simpleType>
</schema>
```

5.1.2. VOD Sample Instance Document

This example shows a minimal IPDR document containing a record describing a summary of a Video on Demand session.

```
<?xml version="1.0"?>

<IPDRDoc xmlns="http://www.ipdr.org/namespaces/ipdr"
     xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
     xsi:schemaLocation=
         “http://www.ipdr.org/namespaces/ipdr/VideoOnDemand1.xsd"
     seqNum="124"
     version="2.0">

  <IPDRRec info="aggregator.ipdr.org"/>
  <IPDR seqNum="1" time="2000-02-01T07:00:00Z">
    <SS id="ses10" service="RTSP">
      <SC xsi:type="SC-VOD-Type">
        <subscriberId>Joe Blow</subscriberId>
        <ipAddress>192.168.1.10</ipAddress>
      </SC>
      <SE xsi:type="SE-VOD-Type">
        <hostName>rtsp.ipdr.org</hostName>
      </SE>
    </SS>
    <UE xsi:type="UE-VOD-Type">
      <movieName>Rocky CIX</movieName>
      <startTime>1999-12-31T23:59:00Z</startTime>
      <endTime>2000-01-01T04:15:00Z</endTime>
      <numAudioStreams>1</numAudioStreams>
      <numVideoStreams>1</numVideoStreams>
      <terminationStatus>normal</terminationStatus>
    </UE>
  </IPDR>
</IPDRDoc>
```

5.2. Service Specifications from Chapter 3 Use Cases

5.2.1 Application Service Provider (ASP)

5.2.1.1 ASP Service-Specific Schema

```
<?xml version="1.0"?>
<schema xmlns="http://www.w3.org/1999/XMLSchema"
   targetNamespace="http://www.ipdr.org/namespaces/ipdr"
   xmlns:ipdr="http://www.ipdr.org/namespaces/ipdr">

      <include schemaLocation="http://www.ipdr.org/public/ipdr2.0.xsd"/>
 <annotation>
   <documentation>The name of the master IPDR schema file can either be:

        http://www.ipdr.org/public/ipdr2.0.xsd

     Alternatively, it can be a local copy of this file.  Please modify the preceding <include
schemaLocation>
     appropriately.  Referring to a local copy will normally yield significantly faster performance.
   </documentation>
 </annotation>

 <complexType name="SC-ASP-Type" base="ipdr:SCType" derivedBy="extension">
   <element name="userLoginName" type="string" minOccurs="1" maxOccurs="1">
     <annotation>
       <documentation>Identifies a unique user in the system.  Real time mapping of dynamically
                      allocated IP addresses might be necessary.
       </documentation>
     </annotation>
   </element>
 </complexType>

 <complexType name="SE-ASP-Type" base="ipdr:SEType" derivedBy="extension">
   <element name="providerLocation" type=”string” minOccurs="1" maxOccurs="1">
     <annotation>
       <documentation>
         Will support providers that host applications at different locations.
       </documentation>
     </annotation>
   </element>
   <element name="providerName" type=”string” minOccurs="1" maxOccurs="1">
     <annotation>
       <documentation>  Actual provider of the service.
       </documentation>
     </annotation>
   </element>
 </complexType>

 <complexType name="UE-ASP-Type" base="ipdr:UEType" derivedBy="extension">
   <element name="feature" type=”string” minOccurs="0" maxOccurs="1">
     <annotation>
       <documentation> Specific feature of the type of service provided.
```

```
           </documentation>
        </annotation>
     </element>
     <element name="type" type=”string” minOccurs="0" maxOccurs="1">
        <annotation>
           <documentation>  Type of application that is invoked.
           </documentation>
        </annotation>
     </element>
     <element name="appRequestTime" type= “datetime” minOccurs="1" maxOccurs="1">
        <annotation>
           <documentation> Time application was requested.  May be different from appStartTime.  This
will allow measuring response time.
           </documentation>
        </annotation>
     </element>
     <element name="appStartTime" type=”datetime” minOccurs="1" maxOccurs="1">
        <annotation>
           <documentation>Time when application starts.
           </documentation>
        </annotation>
     </element>
     <element name="loginLocation" type=”string” minOccurs="1" maxOccurs="1">
        <annotation>
           <documentation>
           </documentation>
        </annotation>
     </element>
     <choice>
       <element ref="ipdr:frontOffice" minOccurs="1" maxOccurs="1"/>
       <element ref="ipdr:backOffice" minOccurs="1" maxOccurs="1"/>
       <element ref="ipdr:onlineTrading" minOccurs="1" maxOccurs="1"/>
     </choice>
  </complexType>

  <group name="frontOffice">
     <element name="appActiveTime" type=”nonNegativeInteger” minOccurs="0" maxOccurs="1">
        <annotation>
           <documentation> Total elapsed active time for each process in the session.  Active time is a
measure of time when the CPU usage exceeds a certain percentage.  At least one of the conditional usage
attributes must be present.  Time is measured in milliseconds
           </documentation>
        </annotation>
     </element>
     <element name="appLoadedTime" type=”nonNegativeInteger” minOccurs="0" maxOccurs="1">
        <annotation>
           <documentation> Total elapsed loaded time for each process in the session.  Time is measured in
milliseconds.
           </documentation>
        </annotation>
     </element>
     <element name="numberOfApps" type=”nonNegativeInteger” minOccurs="0" maxOccurs="1">
        <annotation>
           <documentation> Number of apps invoked during the session time.
           </documentation>
```

```
          </annotation>
      </element>
      <element name="sessionDuration" type=”nonNegativeInteger” minOccurs="0" maxOccurs="1">
          <annotation>
              <documentation> Duration of the session, measured in seconds.
              </documentation>
          </annotation>
      </element>
   </group>

   <group name="backOffice">
      <element name="bytesTransferred " type=”nonNegativeInteger” minOccurs="0" maxOccurs="1">
          <annotation>
              <documentation> Number of bytes transferred on request basis.  Should not include any inline
images or ad view.
              </documentation>
          </annotation>
      </element>
      <element name="numberOfTransactionsRequested" type=”nonNegativeInteger” minOccurs="0"
maxOccurs="1">
          <annotation>
              <documentation> Number of transactions requested by the user during the session..
              </documentation>
          </annotation>
      </element>
      <element name="numberOfTansactionsCompleted " type=”nonNegativeInteger” minOccurs="0"
maxOccurs="1">
          <annotation>
              <documentation> Number of transactions completed.
              </documentation>
          </annotation>
      </element>
      <element name="requestDuration " type=”nonNegativeInteger” minOccurs="0" maxOccurs="1">
          <annotation>
              <documentation> Time between two consecutive requests.
              </documentation>
          </annotation>
      </element>
      <element name="visitTime" type=”nonNegativeInteger” minOccurs="1" maxOccurs="1">
          <annotation>
              <documentation> The duration that covers a series of consecutive requests to the ASP site,
bounded by the first and last requests made by user.
              </documentation>
          </annotation>
      </element>
   </group>

  <group name="onlineTrading ">
      <element name="userAccountNumber" type=”string” minOccurs="1" maxOccurs="1">
          <annotation>
              <documentation> User account number.  May be derived from the user login.
              </documentation>
          </annotation>
      </element>
```

```
    <element name="numberOfEmailAssistanceRequests" type=”nonNegativeInteger” minOccurs="0"
maxOccurs="1">
      <annotation>
        <documentation> Total number of e-mail assistance requests per session.
        </documentation>
      </annotation>
    </element>
    <element name="numberOfNewsRequests" type=”nonNegativeInteger” minOccurs="0"
maxOccurs="1">
      <annotation>
        <documentation> Total number of company news requests per session.
        </documentation>
      </annotation>
    </element>
    <element name="numberOfQuotes" type=”nonNegativeInteger” minOccurs="0" maxOccurs="1">
      <annotation>
        <documentation> Total number of stock quotes obtained during the session.
        </documentation>
      </annotation>
    </element>
    <element name="sessionDuration" type=”nonNegativeInteger” minOccurs="0" maxOccurs="1">
      <annotation>
        <documentation> Duration of the session, measured in seconds, including both browsing and
trading.
        </documentation>
      </annotation>
    </element>
    <element name="tradeRequests" type=”string” minOccurs="0" maxOccurs="1">
      <annotation>
        <documentation> List of trade requests.  For example, “MSFT, buy, 100”.
        </documentation>
      </annotation>
    </element>
  </group>
</schema>
```

5.2.2 Voice over IP (VOIP)

VoIP Service-Specific Schema

```
<?xml version="1.0"?>

<schema xmlns="http://www.w3.org/1999/XMLSchema"
     targetNamespace="http://www.ipdr.org/namespaces/ipdr"
     xmlns:ipdr="http://www.ipdr.org/namespaces/ipdr">

     <include schemaLocation="http://www.ipdr.org/public/ipdr2.0.xsd"/>
  <annotation>
     <documentation>The name of the master IPDR schema file can either be:

              http://www.ipdr.org/public/ipdr2.0.xsd

         Alternatively, it can be a local copy of this file.  Please modify the preceding <include
schemaLocation>
         appropriately.  Referring to a local copy will normally yield significantly faster performance.
     </documentation>
  </annotation>

  <complexType name="SC-VoIP-Type" base="ipdr:SCType" derivedBy="extension">
     <element ref="ipdr:subscriberId" minOccurs="1" maxOccurs="1"/>
     <element ref="ipdr:ipAddress" minOccurs="1" maxOccurs="1"/>
  </complexType>

  <element name="subscriberId" type="string"/>

  <complexType name="UE-VOD-Type" base="ipdr:UEType" derivedBy="extension">
     <element name="imsiIngress" type="integer" minOccurs="0" maxOccurs="1">
         <annotation>
             <documentation> Unique within a service provider network.  Tied to a Service Consumer or a
Service Element requesting a service.
            </documentation>
        </annotation>
    </element>
    <element name="esnIngrss" type="integer" minOccurs="0" maxOccurs="1">
        <annotation>
            <documentation> Optional Electronic Serial Number which uniquely identifies each cellular phone.
Required if calling party is using a cellular phone.
            </documentation>
       </annotation>
    </element>
    <element name="serviceConsumerType" type="string" minOccurs="0" maxOccurs="1">
        <annotation>
            <documentation> Time instant the film stops playing.
            </documentation>
        </annotation>
    </element>
    <element name="pin" type="string" minOccurs="0" maxOccurs="1">
        <annotation>
            <documentation>Unique within a a service provider network.  Tied to a Service Consumer or a
Service Element requesting a service.
```

```
      </documentation>
    </annotation>
  </element>
  <element name="startAccessTime" type="timeInstant" minOccurs="0" maxOccurs="1">
    <annotation>
      <documentation> Time when a Service Consumer starts using a Network Element
      </documentation>
    </annotation>
  </element>
  <element ref="startTime" type="timeInstant" minOccurs="1" maxOccurs="1">
    <annotation>
      <documentation>Time instant when a Service Consumer starts using a Service Element
      </documentation>
    </annotation>
  </element>
  <element name="callDuration" type="non-negative-integer" minOccurs="1" maxOccurs="1">
    <annotation>
      <documentation>Exclusive of all set-up procedures.
      </documentation>
    </annotation>
  </element>
  <element name="totalTime" type="non-negative-integer" minOccurs="1" maxOccurs="1">
    <annotation>
      <documentation>
      </documentation>
    </annotation>
  </element>
  <element name="type" type="string" minOccurs="0" maxOccurs="1">
    <annotation>
      <documentation>A is for administrative (e.g. authentication and authorization), I is for IVR, N is
for no answer, V for voice, F for fax, D for data, VF for voice and fax combination, VD for voice and data
combination.
      </documentation>
    </annotation>
  </element>
  <element name="feature" type="string" minOccurs="0" maxOccurs="1">
    <annotation>
      <documentation>R for roaming, H for home
      </documentation>
    </annotation>
  </element>
  <element name="codec" type="string" minOccurs="0" maxOccurs="1">
    <annotation>
      <documentation>G711Alaw, G711Mulaw, G723Low, G723High, G726, G727, G728, G729A, P
for proprietary.
      </documentation>
    </annotation>
  </element>
  <element name="Modem" type="string" minOccurs="0" maxOccurs="1">
    <annotation>



      <documentation>Required if a modem is involved.  [ed – WTF?]
      </documentation>
    </annotation>
  </element>
  <element name="supplementaryService" type="string" minOccurs="0" maxOccurs="1">
    <annotation>
      <documentation>This field needs to be extensible to accommodate any number and any type of new
service that could be used in conjunction with point-to-point telephony (e.g. call waiting, three-way calling,
call forwarding, etc).
      </documentation>
    </annotation>
  </element>
  <element name="disconnectReason" type="string" minOccurs="1" maxOccurs="1">
    <annotation>
      <documentation>
      </documentation>
    </annotation>
  </element>
  <element name="extendedReasonCode" type="string" minOccurs="0" maxOccurs="1">
    <annotation>
      <documentation>
      </documentation>
    </annotation>
  </element>
  <element name="proprietaryErrorCode" type="integer" minOccurs="0" maxOccurs="1">
    <annotation>
      <documentation>Vendor-specific error code
      </documentation>
    </annotation>
  </element>
  <element name="unitsConsumed" type="integer" minOccurs="0" maxOccurs="1">
    <annotation>
      <documentation>
      </documentation>
    </annotation>
  </element>
  <element name="averageLatency" type="integer" minOccurs="1" maxOccurs="1">
    <annotation>
      <documentation>Incoming from the IP network.  Measured from the preceding node in the call
path.
      </documentation>
    </annotation>
  </element>
  <element name="inboundByteCount" type="integer" minOccurs="0" maxOccurs="1">
    <annotation>
      <documentation>
      </documentation>
    </annotation>
  </element>
  <element name="outboundByteCount" type="integer" minOccurs="0" maxOccurs="1">
    </element>
  <element name="faxPageCount" type="integer" minOccurs="0" maxOccurs="1">
    <annotation>
      <documentation>[ed - This fax stuff should not be in this service definition]
      </documentation>
```

```
        </annotation>
    </element>
    <element name="packetLossPercentage" type="integer" minOccurs="0" maxOccurs="1">
        <annotation>
            <documentation>
            </documentation>
        </annotation>
    </element>
    <element name="outOfSequencePackets" type="integer" minOccurs="0" maxOccurs="1">
        <annotation>
            <documentation>
            </documentation>
        </annotation>
    </element>
    <element name="correctSequencePackets" type="integer" minOccurs="0" maxOccurs="1">
        <annotation>
            <documentation>
            </documentation>
        </annotation>
    </element>
    <element name="ani" type="string" minOccurs="0" maxOccurs="1">
        <annotation>
            <documentation>
            </documentation>
        </annotation>
    </element>
    <element name="iiDigits" type="string" minOccurs="0" maxOccurs="1">
        <annotation>
            <documentation>
            </documentation>
        </annotation>
    </element>
    <element name="dnis" type="string" minOccurs="0" maxOccurs="1">
        <annotation>
            <documentation>
            </documentation>
        </annotation>
    </element>
    <element name="destinationPhoneNumber" type="string" minOccurs="0" maxOccurs="1">
        <annotation>
            <documentation>Digit string entered by the calling party.   Different from DNIS if two-stage
dialing.
            </documentation>
        </annotation>
    </element>
    <element name="outpulsedDigits" type="integer" minOccurs="0" maxOccurs="1">
        <annotation>
            <documentation>Digit string given to the switch on the egress side [Ed – an integer?  WTF?]
            </documentation>
        </annotation>
    </element>
    <element name="ipAddressIngressDevice" type="string" minOccurs="0" maxOccurs="1">
        <annotation>
            <documentation>Required if using a gateway.  Empty if using DHCP [Ed – like that’s going to
work]
```

```
          </documentation>
        </annotation>
    </element>
    <element name="portNumber" type="string" minOccurs="0" maxOccurs="1">
        </element>
    <element name="imsiEgress" type="string" minOccurs="0" maxOccurs="1">
        <annotation>
          <documentation>International Mobile Subscriber Identity.  Required if called party is using a cellular phone.
          </documentation>
        </annotation>
    </element>
    <element name="esnEgress" type="string" minOccurs="0" maxOccurs="1">
        <annotation>
          <documentation>Electronic Serial Number which uniquely identifies each cellular phone.  Required if calling party is using a cellular phone.
          </documentation>
        </annotation>
    </element>
    <element name="homeLocationIdIngress" type="string" minOccurs="0" maxOccurs="1">
        <annotation>
          <documentation> An MSCID or IP address of an HLR.  Required if calling party is using a cellular phone.
          </documentation>
        </annotation>
    </element>
    <element name="homeLocationIdEgress" type="string" minOccurs="0" maxOccurs="1">
        <annotation>
          <documentation>An MSCID or IP address of an HLR.  Required if called party is using a cellular phone.
          </documentation>
        </annotation>
    </element>
  </complexType>
</schema>
```

5.2.3 EMail

EMail Service-Specific Schema

```
<?xml version="1.0"?>

<schema xmlns="http://www.w3.org/2000/XMLSchema"
      targetNamespace="http://www.ipdr.org/namespaces/ipdr"
      xmlns:ipdr="http://www.ipdr.org/namespaces/ipdr">

            <include schemaLocation="http://www.ipdr.org/public/ipdr2.0.xsd"/>
  <annotation>
      <documentation>The name of the master IPDR schema file can either be:

              http://www.ipdr.org/public/ipdr2.0.xsd

          Alternatively, it can be a local copy of this file.  Please modify the preceding <include
schemaLocation>
          appropriately.  Referring to a local copy will normally yield significantly faster performance.
      </documentation>
  </annotation>  <complexType
name="SC-EMAIL-Type" base="ipdr:SCType" derivedBy="extension">
      <element name="ipdr:UserLoginName" type=”string” minOccurs="1" maxOccurs="1"/>
      <annotation>
              <documentation> Unique user in the system </documentation>
      </annotation>
        </element>
      <element
ref="ipdr:UserLoginLocation" minOccurs="1" maxOccurs="1"/>
      <annotation>
              <documentation> Service consumer location information
</documentation>
      </annotation>
        </element>
  </complexType>

  <element name="UserLoginName" type="string"/>
  <element name="UserLoginLocation" type="string"/>

  <complexType
name="SE-EMAIL-Type" base="ipdr:SEType" derivedBy="extension">
       <element name="ipdr:ProviderName" type=”string” minOccurs="1" maxOccurs="1"/>
      <annotation>
              <documentation> Service provider name</documentation>
      </annotation>
        </element>
      <element
ref="ipdr:ProviderLocation" minOccurs="1" maxOccurs="1"/>
      <annotation>
```

```
            <documentation>  Service provider location  </documentation>
      </annotation>
        </element>
  </complexType>

  <element name="ProviderName" type="string"/>
  <element name="ProviderLocation" type="string"/>


  <complexType
name="UE-EMAIL-Type" base="ipdr:UEType" derivedBy="extension">
      <element name="ipdr:Service" type=”string” minOccurs="1" maxOccurs="1"/>
      <annotation>
            <documentation> Type field to distinguish multiple services by
a provider</documentation>
      </annotation>
       </element>
      <element name="ipdr:Feature" type=”string” minOccurs="1" maxOccurs="1"/>
      <annotation>
            <documentation> Distinguish multiple features for a given
service type, e.g., monitoring based on messages versus one based on
storage </documentation>
      </annotation>
       </element>
      <element name="ipdr:ProcessingTime" type=”integer” minOccurs="0" maxOccurs="1"/>
      <annotation>
            <documentation> Total time used by server to process an
email</documentation>
      </annotation>
       </element>
       <element name="ipdr:Storage" type=”integer” minOccurs="0" maxOccurs="1"/>
      <annotation>
            <documentation> Total storage at the time of generating this
usage record </documentation>
      </annotation>
       </element>
       <element
ref="ipdr:StorageDuration" minOccurs="0" maxOccurs="1"/>
      <annotation>
            <documentation>  Duration of time that messages were stored on
the provider's servers       </documentation>
      </annotation>
       </element>
       <element
name="ipdr:BytesTransferred" type=”integer” minOccurs="0" maxOccurs="1"/>
      <annotation>
            <documentation> Total bytes transferred either during email
arrival or email download </documentation>
      </annotation>
       </element>
      <element name="ipdr:EventTime" type=”datetime”minOccurs="0" maxOccurs="1"/>
     <annotation>
             <documentation> Timestamp: events are generated when a message
arrives and when a message is read </documentation>
     </annotation>
      </element>
  </complexType>

<element name="Service" type="string" />
<element name="Feature" type="string" />
<element name="ProcessingTime" type="nonNegativeInteger" />
<element name="Storage" type="nonNegativeInteger" />
<element name="StorageDuration" type="nonNegativeInteger"/>
<element name="BytesTransferred" type="nonNegativeInteger" />
<element name="EventTime" type="timeInstant"/>
</schema>
```

Email Sample Instance Document

This example shows a minimal IPDR document containing a record describing a summary of an e-mail session.

```
<?xml version="1.0"?>

<IPDRDoc xmlns="http://www.ipdr.org/namespaces/ipdr"
     xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
     xsi:schemaLocation=
         "http://www.ipdr.org/namespaces/ipdr/EmailService1.xsd"
     seqNum="124"
     version="2.0">

  <IPDRRec info="SNMPServer.myisp.com"/>
  <IPDR seqNum="1" time="2000-02-01T07:00:00Z">
    <SS id="ses100" service="EMAIL">
      <SC xsi:type="SC-EMAIL-Type">
        <UserLoginName>Joe Verbose</UserLoginName>
        <UserLoginLocation>152.168.1.10</UserLoginLocation>
      </SC>
      <SE xsi:type="SE-EMAIL-Type">
        <ProviderName>Acme ISP</ProviderName>
  <ProviderLocation>208.99.88.99</ProviderLocation>
      </SE>
    </SS>
    <UE xsi:type="UE-EMAIL-Type">
        <Service> Gold Email Service</Service>
        <Feature> Blue Feature Package</Feature>
        <ProcessingTime> 500</ProcessingTime>
        <Storage> 100</Storage>
        <StorageDuration> 3000</StorageDuration>
        <BytesTransferred> 110</BytesTransferred>
        <EventTime> 2000-02-01T07:00:00Z</EventTime>

    </UE>
  </IPDR>
</IPDRDoc>
```

5.2.4 Authentication and Authorization (A&A)

A&A Service-Specific Schema

```
<?xml version="1.0"?>

<schema xmlns="http://www.w3.org/1999/XMLSchema"
      targetNamespace="http://www.ipdr.org/namespaces/ipdr"
      xmlns:ipdr="http://www.ipdr.org/namespaces/ipdr">

      <include schemaLocation="http://www.ipdr.org/public/ipdr2.0.xsd"/>
  <annotation>
      <documentation>The name of the master IPDR schema file can either be:

               http://www.ipdr.org/public/ipdr2.0.xsd

          Alternatively, it can be a local copy of this file.  Please modify the preceding <include
schemaLocation>
          appropriately.  Referring to a local copy will normally yield significantly faster performance.
      </documentation>
  </annotation>
  <complexType name="SC-AA-Type" base="ipdr:SCType" derivedBy="extension">
      <element name="ipdr:subscriberId" type=”string” minOccurs="1" maxOccurs="1"/>
      <element name="ipdr:ipAddress" type=”string” minOccurs="1" maxOccurs="1"/>
  </complexType>

  <element name="subscriberId" type="string"/>
  <element name="ipAddress" type="string"/>

  <complexType name="SE-AA-Type" base="ipdr:SEType" derivedBy="extension">
      <element name="ipdr:hostName" type=”string minOccurs="1" maxOccurs="1"/>
  </complexType>

  <element name="hostName" type="string"/>

  <complexType name="UE-AA-Type" base="ipdr:UEType" derivedBy="extension">
      <element name="typeOfAAService" type="string" minOccurs="1" maxOccurs="1">
          <annotation>
              <documentation>Type of authentication service that is invoked
             </documentation>
        </annotation>
    </element>
      <element name="aArequestTime" type="timeInstant" minOccurs="1" maxOccurs="1">
          <annotation>
              <documentation>May be different from aAAcknowledgeTime.  This will allow measuring
response time.
             </documentation>
        </annotation>
    </element>
      <element name="aAAcknowledgeTime" type="timeInstant" minOccurs="1" maxOccurs="1">
          <annotation>
              <documentation>Time when authentication and/or authorization granted at NAS.
             </documentation>
        </annotation>
    </element>
      <element name="nasId" type="string" minOccurs="1" maxOccurs="1">
          <annotation>
              <documentation>Identification or Location of Network Access Server (NAS)
             </documentation>
        </annotation>
    </element>
      <element name="aAAServerId" type="string" minOccurs="1" maxOccurs="1">
          <annotation>
              <documentation>Identification or Location of Shared (Authentication/Authorization and
Accounting)  Server
             </documentation>
        </annotation>
    </element>
      <element name="providerName" type="string" minOccurs="1" maxOccurs="1">
          <annotation>
              <documentation>Actual provider of the AA service
             </documentation>
        </annotation>
    </element>
      <element name="userLoginName" type="string" minOccurs="1" maxOccurs="1">
          <annotation>
              <documentation>Identifies a unique user in the system.  Real time mapping of dynamically
allocated IP addresses might be necessary.
             </documentation>
        </annotation>
    </element>
  </complexType>

</schema>
```

5.2.5 Internet Access

Internet Access Service-Specific Schema

```
<?xml version="1.0"?>
<schema xmlns="http://www.w3.org/1999/XMLSchema"
    targetNamespace="http://www.ipdr.org/namespaces/ipdr"
    xmlns:ipdr="http://www.ipdr.org/namespaces/ipdr">

    <include schemaLocation="ipdr2.0.xsd"/>

  <annotation>
    <documentation>The name of the master IPDR schema file can either be:

          http://www.ipdr.org/public/ipdr2.0.xsd

      Alternatively, it can be a local copy of this file.  Please modify the preceding <include
schemaLocation>
      appropriately.  Referring to a local copy will normally yield significantly faster performance.
    </documentation>
  </annotation>

  <complexType name="SC-IA-Type" base="ipdr:SCType" derivedBy="extension">
    <element name="subscriberID" minOccurs="1" maxOccurs="1”>
      <complexType base="string" derivedBy="extension">
        <attribute name="type" type="string"/>
      </complexType>
      <annotation>
        <documentation>
          Identifies a unique subscriber in the system.  Type can be IMSI, IMEI, IP, PN, and CUST
        </documentation>
      </annotation>
    </element>
  </complexType>

  <complexType name="SE-IA-Type" base="ipdr:SEType" derivedBy="extension">
    <element name="serviceElement" type="string" minOccurs="0" maxOccurs="1">
      <annotation>
        <documentation> Service element used to provide access.
        </documentation>
      </annotation>
    </element>
    <element name="serviceProviderID" type="string" minOccurs="1" maxOccurs="1">
      <annotation>
        <documentation> Service provider that is providing internet access.
        </documentation>
      </annotation>
    </element>
  </complexType>

  <complexType name="UE-IA-Type" base="ipdr:UEType" derivedBy="extension">
    <element name="transportProtocol" type="string" minOccurs="1" maxOccurs="1">
      <annotation>
        <documentation> Transport protocol that was used for internet access.
        </documentation>
      </annotation>
    </element>
    <element name="connectionType" type="string" minOccurs="0" maxOccurs="1">
      <annotation>
        <documentation> Connection type that was used for internet access.
        </documentation>
      </annotation>
    </element>
    <element name="upBandwidth" minOccurs="0" maxOccurs="1">
      <complexType base="nonNegativeInteger" derivedBy="extension">
        <attribute name="unit" type="ipdr:throughputUnit"/>
      </complexType>
      <annotation>
        <documentation> Upstream bandwidth provided.
        </documentation>
      </annotation>
    </element>
    <element name="downBandwidth" minOccurs="0" maxOccurs="1">
      <complexType base="nonNegativeInteger" derivedBy="extension">
        <attribute name="unit" type="ipdr:throughputUnit"/>
      </complexType>
      <annotation>
       <documentation>  Downstream bandwidth provided.
       </documentation>
      </annotation>
    </element>
    <element name="upVolume" minOccurs="0" maxOccurs="1">
      <complexType base="nonNegativeInteger" derivedBy="extension">
        <attribute name="unit" type="ipdr:volumeUnit"/>
      </complexType>
      <annotation>
        <documentation> Volume that was uploaded.
        </documentation>
      </annotation>
    </element>
    <element name="downVolume" minOccurs="0" maxOccurs="1">
      <complexType base="nonNegativeInteger" derivedBy="extension">
        <attribute name="unit" type="ipdr:volumeUnit"/>
      </complexType>
      <annotation>
        <documentation> Volume that was downloaded.
        </documentation>
      </annotation>
    </element>
    <element name="qosRequested" type="byte" minOccurs="0" maxOccurs="1">
      <annotation>
        <documentation>  Requested QoS, corresponding to the SLA or dynamic QoS request.
        </documentation>
      </annotation>
    </element>
    <element name="qosDelivered" type="byte " minOccurs="0" maxOccurs="1">
      <annotation>
        <documentation>
```

```
                   Pre-calculated indicator representing the delivered/negotiated QoS.  Physical attributed such as
latency or error rates are weighted and combined into one value.
              </documentation>
          </annotation>
      </element>
      <element name="startTime" type="timeInstant" minOccurs="1" maxOccurs="1">
          <annotation>
              <documentation> When access started.
              </documentation>
          </annotation>
      </element>
      <choice>
         <element ref="ipdr:endTime" minOccurs="1" maxOccurs="1"/>
         <element ref="ipdr:duration" minOccurs="1" maxOccurs="1"/>
         <element ref="ipdr:endTimeAndDuration" minOccurs="1" maxOccurs="1"/>
      </choice>
      <element name="accessPoint" type="string" minOccurs="1" maxOccurs="1">
          <annotation>
              <documentation>
                   Identifies access point to the internet.  Equivalent to APN for GPRS, NAS.
              </documentation>
          </annotation>
      </element>
  </complexType>

  <group name="endTimeAndDuration">
     <element ref="ipdr:endTime" minOccurs="1" maxOccurs="1" />
     <element ref="ipdr:duration"  minOccurs="1" maxOccurs="1"/>
  </group>

  <element name="endTime" type="timeInstant" />
      <annotation>
          <documentation> When access stopped.  At least endTime or duration needs to be present.
          </documentation>
      </annotation>
  </element>

b  <element name="duration">
      <complexType base="nonNegativeInteger" derivedBy="extension">
          <attribute name="unit" type="ipdr:durationUnit" />
      </complexType>
      <annotation>
          <documentation> Duration of access.
          </documentation>
      </annotation>
  </element>

  <simpleType name="volumeUnit" base="string">
      <enumeration value="bytes"/>
      <enumeration value="KB"/>
      <enumeration value="MB"/>
      <enumeration value="GB"/>
      <enumeration value="TB"/>
</simpleType>
```

```
 <simpleType name="throughputUnit" base="string">
     <annotation>
          <documentation>  bps stands for bits per second.
          </documentation>
     </annotation>
     <enumeration value="bps"/>
     <enumeration value="Kbps"/>
     <enumeration value="Mbps"/>
     <enumeration value="Gbps"/>
     <enumeration value="Tbps"/>
</simpleType>

 <simpleType name="durationUnit" base="string">
     <enumeration value="ms"/>
     <enumeration value="s"/>
     <enumeration value="min"/>
     <enumeration value="h"/>
 </simpleType>

</schema>
```

Wireless Internet Access Service-Specific Schema

```
<?xml version="1.0"?>
<schema xmlns="http://www.w3.org/1999/XMLSchema"
      targetNamespace="http://www.ipdr.org/namespaces/ipdr"
      xmlns:ipdr="http://www.ipdr.org/namespaces/ipdr">

      <include schemaLocation="InternetAccess2.0.xsd"/>

  <annotation>
      <documentation>The name of the master IPDR schema file can either be:

              http://www.ipdr.org/public/ipdr2.0.xsd

          Alternatively, it can be a local copy of this file.  Please modify the preceding <include
schemaLocation>
          appropriately.  Referring to a local copy will normally yield significantly faster performance.
      </documentation>
  </annotation>

  <complexType name="SC-WIA-Type" base="ipdr:SC-IA-Type" derivedBy="extension">
      <element name="cellID" type="string" minOccurs="1" maxOccurs="1">
          <annotation>
              <documentation>  ID of the cell that that is handling the connection to the wireless terminal.
              </documentation>
          </annotation>
      </element>
  </complexType>

  <complexType name="SE-WIA-Type" base="ipdr:SE-IA-Type" derivedBy="extension">
      <element name="serviceBearer" type="string" minOccurs="1" maxOccurs="1">
          <annotation>
              <documentation>
                  Service bearer that was used for connection protocol.
              </documentation>
          </annotation>
      </element>
  </complexType>

  <complexType name="UE-WIA-Type" base="ipdr:UE-IA-Type" derivedBy="extension">
      <element name="routingArea" type="string" minOccurs="1" maxOccurs="1">
          <annotation>
              <documentation>  Subset of location area.
              </documentation>
          </annotation>
      </element>
      <element name="locationArea" type="string" minOccurs="1" maxOccurs="1">
          <annotation>
              <documentation>  The geographical area from which the connection is established.
              </documentation>
          </annotation>
      </element>
  </complexType>
</schema>
```

5.2.6 Content/Service

Content/Service Service-Specific Schema

```
<?xml version="1.0"?>
<schema xmlns="http://www.w3.org/1999/XMLSchema"
     targetNamespace="http://www.ipdr.org/namespaces/ipdr"
     xmlns:ipdr="http://www.ipdr.org/namespaces/ipdr">

     <include schemaLocation="InternetAccess2.0.xsd"/>

  <annotation>
     <documentation>The name of the master IPDR schema file can either be:

             http://www.ipdr.org/public/ipdr2.0.xsd

         Alternatively, it can be a local copy of this file.  Please modify the preceding <include
schemaLocation>
         appropriately.  Referring to a local copy will normally yield significantly faster performance.
     </documentation>
  </annotation>

  <complexType name="SC-CS-Type" base="ipdr:SC-IA-Type" derivedBy="extension">
     <annotation>
         <documentation> There are no new elements, this exists so that the SC-CS-Type is a valid type.
         </documentation>
  </complexType>

  <complexType name="SE-CS-Type" base="ipdr:SE-IA-Type" derivedBy="extension">
     <annotation>
         <documentation> There are no new elements, this exists so that the SE-CS-Type is a valid type.
         </documentation>
  </complexType>

  <complexType name="UE-CS-Type" base="ipdr:UE-IA-Type" derivedBy="extension">
    <element name="ipServiceID" minOccurs="1" maxOccurs="1">
         <complexType base="string" derivedBy="extension">
             <attribute name="type" type="string"/>
         </complexType>
         <annotation>
             <documentation>  Service address.
             </documentation>
         </annotation>
     </element>
    <element name="ipServiceClass" type="string" minOccurs="0" maxOccurs="1">
         <annotation>
             <documentation>
                  Service classification, for example website, quote service, directory service, etc.
             </documentation>
         </annotation>
     </element>
    <element name="numberOfTransactions" type="integer" minOccurs="0" maxOccurs="1">
         <annotation>
             <documentation>
         Number of transactions associated with the service.
       </documentation>
     </annotation>
   </element>
   <element name="amount" minOccurs="0" maxOccurs="1">
     <complexType base="decimal" derivedBy="extension">
       <attribute name="unit" type="string"/>
       <annotation>
         <documentation>
           Use ISO 4217 codes for monetary unit (USD, EUR, etc)
         </documentation>
       </annotation>
     </complexType>
     <annotation>
       <documentation>  Monetary amount to be charged for service.
       </documentation>
     </annotation>
   </element>
 </complexType>
</schema>
```

Wireless Content/Service Service-Specific Schema

```
<?xml version="1.0"?>
<schema xmlns="http://www.w3.org/1999/XMLSchema"
     targetNamespace="http://www.ipdr.org/namespaces/ipdr"
     xmlns:ipdr="http://www.ipdr.org/namespaces/ipdr">

     <include schemaLocation="ContentService2.0.xsd"/>

  <annotation>
     <documentation>The name of the master IPDR schema file can either be:

              http://www.ipdr.org/public/ipdr2.0.xsd

         Alternatively, it can be a local copy of this file.  Please modify the preceding <include
schemaLocation>
         appropriately.  Referring to a local copy will normally yield significantly faster performance.
     </documentation>
  </annotation>

  <complexType name="SC-WCS-Type" base="ipdr:SC-CS-Type" derivedBy="extension">
     <element name="cellID" type="string" minOccurs="1" maxOccurs="1">
         <annotation>
             <documentation> ID of the cell that that is handling the connection to the wireless terminal.
             </documentation>
         </annotation>
     </element>
  </complexType>

  <complexType name="SE-WCS-Type" base="ipdr:SE-CS-Type" derivedBy="extension">
     <element name="serviceBearer" type="string" minOccurs="1" maxOccurs="1">
         <annotation>
             <documentation>  Service bearer that was used for connection protocol.
             </documentation>
         </annotation>
     </element>
  </complexType>

  <complexType name="UE-WCS-Type" base="ipdr:UE-CS-Type" derivedBy="extension">
    <element name="routingArea" type="string" minOccurs="1" maxOccurs="1">
         <annotation>
             <documentation> Subset of location area.
             </documentation>
         </annotation>
     </element>
    <element name="locationArea" type="string" minOccurs="1" maxOccurs="1">
         <annotation>
             <documentation>  The geographical area from which the connection is established.
             </documentation>
         </annotation>
     </element>
  </complexType>
</schema>
```

5.2.7 Push-Delivery

Push-Delivery Service-Specific Schema

```
<?xml version="1.0"?>

<schema xmlns="http://www.w3.org/1999/XMLSchema"
    targetNamespace="http://www.ipdr.org/namespaces/ipdr"
    xmlns:ipdr="http://www.ipdr.org/namespaces/ipdr">

    <include schemaLocation="ipdr2.0.xsd"/>

  <annotation>
    <documentation>The name of the master IPDR schema file can either be:

          http://www.ipdr.org/public/ipdr2.0.xsd

      Alternatively, it can be a local copy of this file.  Please modify the preceding <include
schemaLocation>
      appropriately.  Referring to a local copy will normally yield significantly faster performance.
    </documentation>
  </annotation>

  <complexType name="SC-PD-Type" base="ipdr:SCType" derivedBy="extension">
    <element name="pushRecipient" minOccurs="1" maxOccurs="1">
      <complexType base="string" derivedBy="extension">
         <attribute name="type" type="string"/>
      </complexType>
      <annotation>
        <documentation>
          Identifies a unique subscriber in the system to receive the push message.
        </documentation>
      </annotation>
    </element>
  </complexType>

  <complexType name="SE-PD-Type" base="ipdr:SEType" derivedBy="extension">
    <element name="serviceElement" type="string" minOccurs="0" maxOccurs="1">
      <annotation>
        <documentation>  Service element used to deliver push message.
        </documentation>
      </annotation>
    </element>
    <element name="serviceProviderID" type="string" minOccurs="1" maxOccurs="1">
      <annotation>
        <documentation>  Service provider delivering push message.
        </documentation>
      </annotation>
    </element>
  </complexType>

  <complexType name="UE-PD-Type" base="ipdr:UEType" derivedBy="extension">
   <element name="pushInitiator" minOccurs="1" maxOccurs="1">
      <complexType base="string" derivedBy="extension">
      <attribute name="type" type="string"/>
   </complexType>
   <annotation>
      <documentation>   Identity of push initiator.
      </documentation>
   </annotation>
</element>
<element name="deliveryStatus" type="string" minOccurs="1" maxOccurs="1">
   <annotation>
      <documentation>
         Status of push delivery.  Can be unconfirmed, succeeded, failed.
      </documentation>
   </annotation>
</element>
<element name="deliveryFeatures" type="string" minOccurs="0" maxOccurs="1">
   <annotation>
      <documentation>  Mode of delivery .
      </documentation>
   </annotation>
</element>
<element name="deliveryTime" type="timeInstant" minOccurs="1" maxOccurs="1">
   <annotation>
      <documentation> Time of push delivery.
      </documentation>
   </annotation>
</element>
<element name="pushID" type="string" minOccurs="0" maxOccurs="1">
   <annotation>
      <documentation> ID of push message as provided by push initiator.
      </documentation>
   </annotation>
</element>
<element name="contentType" type="string" minOccurs="1" maxOccurs="1">
   <annotation>
      <documentation> Type of push content.
      </documentation>
   </annotation>
</element>
<element name="contentSize" minOccurs="1" maxOccurs="1">
   <complexType base="nonNegativeInteger" derivedBy="extension">
      <attribute name="unit" type="ipdr:volumeUnit"/>
   </complexType>
   <annotation>
      <documentation> Size of push message.
      </documentation>
   </annotation>
</element>
<element name="priority" type="nonNegativeInteger" minOccurs="0" maxOccurs="1">
   <annotation>
      <documentation> Priority of push message.
      </documentation>
   </annotation>
</element>
<element name="applicationID" type="string" minOccurs="0" maxOccurs="1">
   <annotation>
```

```
            <documentation>
                 ID of receiving application.
            </documentation>
        </annotation>
    </element>
    <element name="transportProtocol" type="string" minOccurs="1" maxOccurs="1">
        <annotation>
            <documentation>
                Transport protocol that was used for internet access.
            </documentation>
        </annotation>
    </element>
    <element name="connectionType" type="string" minOccurs="0" maxOccurs="1">
        <annotation>
            <documentation>
                Connection type that was used for internet access.
            </documentation>
        </annotation>
    </element>
  </complexType>
  <simpleType name="volumeUnit" base="string">
      <enumeration value="bytes"/>
      <enumeration value="KB"/>
      <enumeration value="MB"/>
      <enumeration value="GB"/>
      <enumeration value="TB"/>
 </simpleType>

</schema>
```

Wireless Push-Delivery Service-Specific Schema

```
<?xml version="1.0"?>

<schema xmlns="http://www.w3.org/1999/XMLSchema"
    targetNamespace="http://www.ipdr.org/namespaces/ipdr"
    xmlns:ipdr="http://www.ipdr.org/namespaces/ipdr">

    <include schemaLocation="PushDelivery2.0.xsd"/>

  <annotation>
    <documentation>The name of the master IPDR schema file can either be:

         http://www.ipdr.org/public/ipdr2.0.xsd

      Alternatively, it can be a local copy of this file.  Please modify the preceding <include
schemaLocation>
      appropriately.  Referring to a local copy will normally yield significantly faster performance.
    </documentation>
  </annotation>

  <complexType name="SC-WPD-Type" base="ipdr:SC-PD-Type" derivedBy="extension">
    <element name="cellID" type="string" minOccurs="1" maxOccurs="1">
      <annotation>
        <documentation> ID of the cell that that is handling the connection to the wireless terminal.
        </documentation>
      </annotation>
    </element>
  </complexType>

  <complexType name="SE-WPD-Type" base="ipdr:SE-PD-Type" derivedBy="extension">
    <element name="serviceBearer" type="string" minOccurs="1" maxOccurs="1">
      <annotation>
        <documentation>  Service bearer that was used for connection protocol.
        </documentation>
      </annotation>
    </element>
  </complexType>

  <complexType name="UE-WPD-Type" base="ipdr:UE-PD-Type" derivedBy="extension">
   <element name="routingArea" type="string" minOccurs="1" maxOccurs="1">
     <annotation>
       <documentation> Subset of location area.
       </documentation>
     </annotation>
   </element>
   <element name="locationArea" type="string" minOccurs="1" maxOccurs="1">
     <annotation>
       <documentation>The geographical area from which the connection is established.
       </documentation>
     </annotation>
   </element>
  </complexType>
<schema/>
```

5.2.8 Wholesale

```
<?xml version="1.0"?>
<schema xmlns="http://www.w3.org/1999/XMLSchema"
      targetNamespace="http://www.ipdr.org/namespaces/ipdr"
      xmlns:ipdr="http://www.ipdr.org/namespaces/ipdr">
            <include schemaLocation="http://www.ipdr.org/public/ipdr2.0.xsd"/>
  <annotation>
      <documentation>The name of the master IPDR schema file can either be:

               http://www.ipdr.org/public/ipdr2.0.xsd

          Alternatively, it can be a local copy of this file.  Please modify the preceding <include
schemaLocation>
          appropriately.  Referring to a local copy will normally yield significantly faster performance.
      </documentation>
  </annotation>
  <complexType name="wholesaleType" base="ipdr:SCType" derivedBy="extension">
      <element ref="ipdr:subscriberId" minOccurs="1" maxOccurs="1"/>
      <element ref="ipdr:ipAddress" minOccurs="1" maxOccurs="1"/>
  </complexType>
    <element name="originationIPAddress" type="string"/>
    <element name="homeGatewayIPAddress" type="string"/>
    <element name="partnerGatewayIPAddress" type="string"/>
    <element name="destinationIPAddress" type="string"/>
    </element>
</schema>
```

5.2.9 Additional Element Details

Element Naming

Some element names have been shortened. Human readability is a nice feature of ASCII XML, but for the most part these documents will be processed by applications. Transfer, storage and parsing costs should not be entirely ignored when developing service-specific schemas. The most frequently used elements gain the most from shortened names.

Timestamps

Human readability is a desirable property of an XML document. When humans are intended or even somewhat likely consumers of an IPDR document, the format are described (along with many more) in ISO 8601.

Date representations which are human readable are not a particularly efficient mechanism for some applications. Using the direct integer value of UTC time can be a simpler format and parse exercise . Both the 32-bit (UNIX, NT) and 64-bit (Java) representations should be allowed. To aid parsing distinction from the human format, the UTC time representations will be written beginning the letter 'U'. Whether the value is a 32-bit or 64-bit quantity is determined by the second character position. For 64-bits the second position is the letter 'L', for 32-bit it will be a digit. Note that 32-bit values record seconds since Epoch, while the 64-bit records milliseconds since Epoch. Epoch is defined as Jan 1, 1970 00:00 GMT.

Examples

1999-09-22T11:20:01-08:00
1999-09-22T19:20:01Z
U938292388
UL93829238801

6. Appendix – Use Case Template

There is no need for an introduction to this section since it is part of a bigger document where introduction is included.

Service Definition

Include in this section a brief description of the service covered in this use case. Also, list service features where it is important to know the feature for a true identification of the service. This is a one paragraph high-level description of the service.

Service Requirements

Include in this section the general requirements that apply to all the basic and the alternative flows of the use case. Itemize the requirements and include a few sentences description of each one of them. Those requirements should be derived from the use cases described below.

Service Usage Attribute List

Include in this section a listing of the usage attributes that are common to all the basic and alternative flows of the use cases. A table format is preferable. Include information such as usage attribute category, usage attribute name, data type, possible values, and remarks. Reference the use case step and the requirement number that lead to this usage attribute.

Use Case

Basic Flow

Describe the basic flow steps of the use case. Itemize the steps and include a few sentences description of each step. Also, include a high-level state diagram that shows the general flow. Concentrate on the interfaces between the different sub-systems. Try not to go into great detail about each sub-system internals. This approach will be closer to the business level use case than that of the technical flow of information.

Basic Flow Requirements

Include in this section the requirements that are specific to this basic flow. You do not need to repeat the general requirements mentioned earlier. However, to indicate the continuity of those requirements numbering could be consecutive from the previous section.

Basic Flow Usage Attribute List

Use the same table format as for the general usage attribute list.

Alternative/Specific Flow (repeat as necessary)

Describe the alternative flow steps using the same format as the basic flow. Include any other interfaces that are presented in this alternative. Repeat the state diagram highlighting the changes of this alternative flow.

Alternative/Specific Flow Requirements

Same as for Basic Flow

Alternative/Specific Flow Usage Attribute List

Same as basic flow